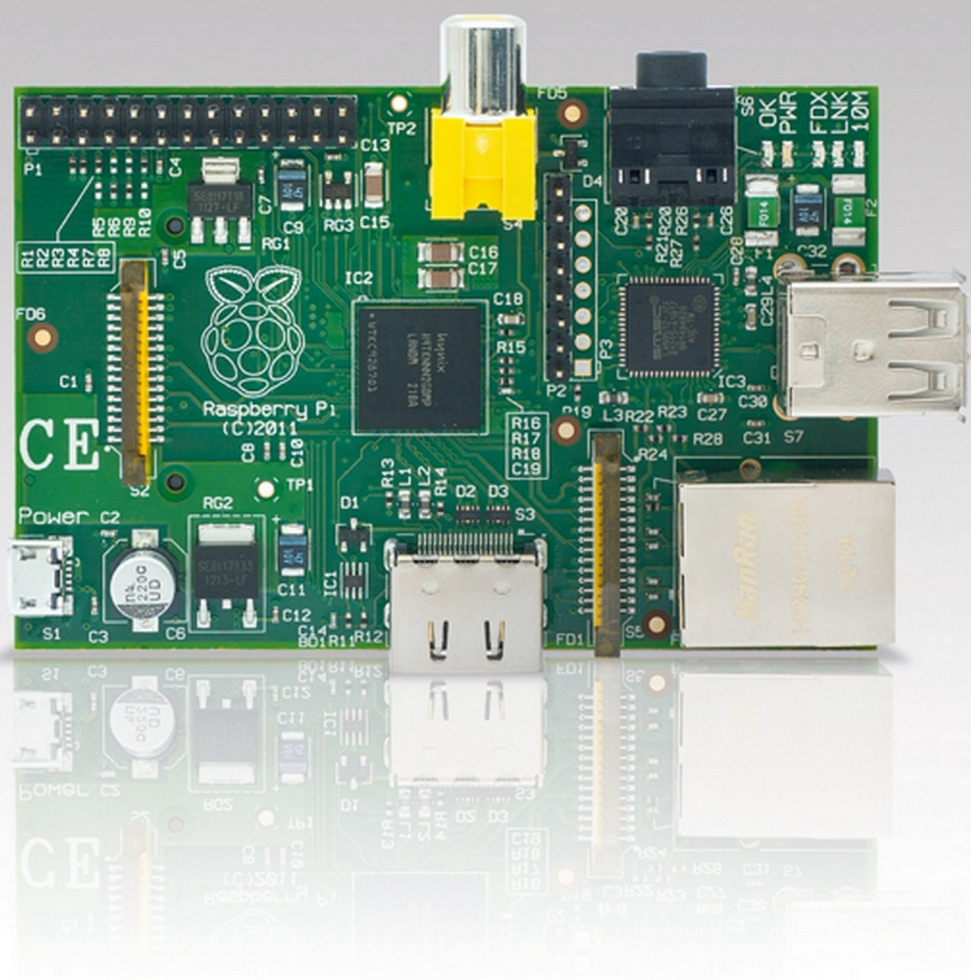




Raspberry Pi®

Uživatelská příručka



computer
press

Eben Upton

Spoluautor počítače Raspberry Pi

Gareth Halfacree

Eben Upton, Gareth Halfacree

Raspberry Pi

Uživatelská příručka

**Computer Press
Brno
2013**

Raspberry Pi

Uživatelská příručka

Eben Upton, Gareth Halfacree

Překlad: Jakub Goner

Odpovědný redaktor: Martin Herodek

Technický redaktor: Jiří Matoušek

Přeloženo z anglického originálu: Raspberry Pi® User Guide

© 2012 Eben Upton and Gareth Halfacree

All Rights Reserved. Authorised translation from the English language edition published by John Wiley & Sons Limited. Responsibility for the accuracy of the translation rests solely with Albatros Media a.s. and is not the responsibility of John Wiley & Sons Limited. No part of this book may be reproduced in any form without the written permission of the original copyright holder, John Wiley & Sons Limited.

Translation © Jakub Goner, 2013

Objednávky knih:

<http://knihy.cpress.cz>

www.albatrosmedia.cz

eshop@albatrosmedia.cz

bezplatná linka 800 555 513

ISBN 978-80-251-4116-8

Vydalo nakladatelství Computer Press v Brně roku 2013 ve společnosti Albatros Media a. s. se sídlem Na Pankráci 30, Praha 4. Číslo publikace 18075.

© Albatros Media a. s. Všechna práva vyhrazena. Žádná část této publikace nesmí být kopírována a rozmnožována za účelem rozšiřování v jakékoli formě či jakýmkoli způsobem bez písemného souhlasu vydavatele.

1. vydání

 **ALBATROS** MEDIA a.s.

Obsah

O autorech	11
Úvod	13
Programování je zábavné!	13
Trocha historie	15
K čemu je tedy počítač Raspberry Pi dobrý?	19
Zpětná vazba od čtenářů	20
Zdrojové kódy ke knize	21
Errata	21

ČÁST I

PŘIPOJENÍ ZÁKLADNÍ DESKY

KAPITOLA 1

Seznámení s počítačem Raspberry Pi	25
ARM versus x86	25
Windows versus Linux	27
Začínáme s počítačem Raspberry Pi	28
Připojení displeje	28
Zvukové připojení	30
Připojení klávesnice a myši	31
Nahrání systému na kartu SD	32
Připojení externího úložiště	36
Připojení k síti	36
Připojení napájení	39

KAPITOLA 2

Správa systému Linux	41
Linux: přehled	41
Základy systému Linux	43
Představení distribuce Debian	44
Použití externích úložných zařízení	47
Vytvoření nového uživatelského účtu	48

Struktura systémů souborů	50
Logická struktura	50
Fyzická struktura	51
Instalování a odinstalování softwaru	52
Hledání softwaru	53
Instalace softwaru	53
Odinstalování softwaru	54
Upgrade softwaru	55
KAPITOLA 3	
Řešení potíží	57
Diagnostika klávesnice a myši	57
Diagnostika napájení	58
Diagnostika zobrazení	60
Diagnostika spouštění	60
Diagnostika sítě	61
Nouzové jádro	63
KAPITOLA 4	
Konfigurace sítě	65
Kabelové připojení k síti	65
Bezdrátové připojení k síti	68
KAPITOLA 5	
Správa oddílů	77
Vytvoření nového oddílu	77
Změna velikosti stávajících oddílů	81
Automatická změna velikosti	81
Ruční změna velikosti	82
Přechod na větší kartu SD	86
Vytvoření bitové kopie v Linuxu	86
Vytvoření bitové kopie v systému OS X	87
Vytvoření bitové kopie v systému Windows	88

KAPITOLA 6

Konfigurace počítače Raspberry Pi	91
Nastavení hardwaru – soubor config.txt	91
Úpravy zobrazení	92
Možnosti spouštění	95
Přetaktování počítače Raspberry Pi	95
Vypnutí mezipaměti L2	99
Zapnutí testovacího režimu	99
Rozdělení paměti – soubor start.elf	100
Nastavení softwaru – soubor cmdline.txt	101

ČÁST II

POUŽITÍ POČÍTAČE PI JAKO MEDIÁLNÍHO CENTRA,
KANCELÁŘSKÉHO POČÍTAČE A WEBOVÉHO SERVERU

KAPITOLA 7

Počítač Pi jako domácí kino	107
Přehrávání hudby v konzole	107
Specializovaný počítač typu HTPC díky distribuci Raspbmc	109
Streamování internetových médií	110
Streamování médií v místní síti	112
Konfigurace distribuce Raspbmc	113

KAPITOLA 8

Počítač Pi jako kancelářský počítač	115
Použití cloudových aplikací	115
Použití sady OpenOffice.org	118
Úpravy obrázků programem Gimp	120

KAPITOLA 9

Počítač Pi jako webový server	123
Instalace sady LAMP	123
Instalace platformy WordPress	127

ČÁST III

PROGRAMOVÁNÍ A HACKOVÁNÍ

KAPITOLA 10

Úvod do jazyka Scratch 133**Představení jazyka Scratch 133****Příklad 1: Ahoj Raspberry 134****Příklad 2: Animace a zvuk 137****Příklad 3: Jednoduchá hra 140****Robotika a senzory 146**

Detekce pomocí sady PicoBoard 146

Robotika se sadou LEGO 146

Další zdroje informací 147

KAPITOLA 11

Úvod do jazyka Python 149**Představení jazyka Python 149****Příklad 1: Ahoj Raspberry 149****Příklad 2: Komentáře, vstupy, proměnné a cykly 154****Příklad 3: Tvorba her pomocí knihovny pygame 158****Příklad 4: Jazyk Python a síťová konektivita 166****Další zdroje informací 171**

KAPITOLA 12

Hackování hardwaru 173**Elektronické příslušenství 173****Čtení barevných kódů rezistorů 175****Získávání komponent 177**

Zdroje online 177

Zdroje offline 177

Firmy specializované na nadšence 178

Port GPIO 179

Sériová sběrnice UART 181

Sběrnice I²C 181

Sběrnice SPI 182

Použití portu GPIO v jazyce Python 182

Instalace knihovny jazyka Python pro port GPIO 183

Výstup portu GPIO: Blikání diody LED	184
Vstup portu GPIO: Detekce tlačítka	188
Postup od zkušebního modelu obvodu	191
Stručný průvodce pájením	194
KAPITOLA 13	
Rozšiřující desky	199
Slice of Pi společnosti Ciseco	199
Prototyping Pi Plate společnosti Adafruit	203
Gertboard společnosti Fen Logic	206
ČÁST IV	
PŘÍLOHY	
PŘÍLOHA A	
Recepty jazyka Python	213
Had s malinou (kapitola 11, příklad 3)	213
Seznam uživatelů IRC (kapitola 11, příklad 4)	215
Vstup a výstup portu GPIO (kapitola 12)	217
PŘÍLOHA B	
Režimy zobrazení HDMI	219
Rejstřík	225

Věnuji Liz, která to všechno umožnila.

Eben

*Mému otci, který představuje nadšení minulosti, a mé dceři,
která má před sebou slibnou budoucnost.*

Gareth

O autorech

Eben Upton je zakladatel a člen dozorčí rady nadace Raspberry Pi Foundation a zastává funkci jejího výkonného ředitele. Odpovídá za celkovou softwarovou a hardwarovou architekturu počítače Raspberry Pi a za vztahy nadace s jejími klíčovými dodavateli a zákazníky. Ve svém předchozím životě založil dvě úspěšné firmy zaměřené na mobilní hry a middleware, Idea-works 3d Ltd. a Podfun Ltd., a pracoval na pozici studijního ředitele pro informatiku na St John's College v Cambridgi. Je držitelem titulů BA, PhD a MBA z Cambridgeské univerzity.

Pracuje ve společnosti Broadcom jako architekt integrovaných obvodů ASIC a univerzální potížišťa.

Gareth Halfacree je nezávislý technologický novinář, který s jedním ze zakladatelů projektu Raspberry Pi Ebenem Uptonem napsal tuto knihu *Raspberry Pi Uživatelská příručka*. Gareth se původně živil jako správce systému ve vzdělávacím sektoru, ale nadšení pro projekty open source jej neopouštělo ani v dalších profesích. Často pracuje na revizích a dokumentaci projektů, jako je GNU/Linux, LibreOffice, Fritzing a Arduino, nebo k nim přímo přispívá. Je také autorem otevřených hardwarových projektů Sleepduino a Burnduino, které rozšiřují možnosti systému prototypování elektroniky Arduino. Přehled jeho aktuálních aktivit naleznete na adrese <http://freelance.halfacree.co.uk>.

Úvod

„Dnešní děti berou počítače jako něco samozřejmého,“ prohlásil minulého roku chlapík, se kterým jsem mluvil na veřejné slavnosti. „Nechápu, proč tu věc vyrábíte. Moje děti vědí o nastavení PC více než já.“

Zeptal jsem se jej, zda dokáže programovat, na což mi odpověděl: „Proč by se něčím takovým měly zabývat? Počítače už umí všechno, co od nich potřebujeme. O to přeci jde, ne?“

Pro spoustu dětí kupodivu nejsou počítače druhou přirozeností. Ještě jsme nepotkali žádného z těch očekávaných digitálních divochů, kteří se houpou na lanech z kroucené dvojlinky a pokřikují bojové písně v elegantně formátovaném jazyku Python. V rámci aktivních vzdělávacích programů naší nadace Raspberry Pi Foundation se setkáváme s mnoha dětmi, jejichž interakce s technologiemi se omezuje jen na uzavřené platformy s grafickými uživatelskými rozhraními, které používají k přehrávání filmů, občasnému psaní domácích úloh v textovém procesoru a hraní her. Dokáží procházet web, odesílat obrázky a video, a někdy dokonce navrhovat webové stránky. (Také si zpravidla lépe než jejich rodiče umí poradit s nastavením satelitní televize.) Tato sada dovedností je sice užitečná, ale překvapivě neúplná. V Británii, kde stále není vybaveno počítačem 20 % domácností (v České republice třetina), přitom není všem dětem dostupná ani tato omezená sada.

Navzdory pevné víře mého známého, se kterým jsem mluvil na slavnosti, se počítače nedokáží programovat samy. Potřebujeme zástupy zkušených techniků, kteří budou technologie dále rozvíjet, a potřebujeme, aby mladí lidé obsazovali pracovní místa uvolněná staršími odborníky, kteří z oboru odcházejí na odpočinek. Výuka programátorského myšlení však není užitečná jen k pěstování nové generace kodérů a hardwarových kutilů. Schopnosti uspořádat vlastní kreativní myšlenky a úkoly komplexním a nelineárním způsobem se lze naučit a tato schopnost poskytuje značné výhody každému, kdo ji získá, od historiků po návrháře, právníky a chemiky.

Programování je zábavné!

Je to mimořádná, vděčná a kreativní zábava. Můžeme přitom vytvářet úžasná spleťtá díla a také (což je dle názoru autora ještě úžasnější) nacházet chytré, neuvěřitelně rychlé a zdánlivě jednoduché zkratky, které umožní překonat libovolné překážky. Dokážeme tvořit díla, která budou naši kolegové závistivě sledovat a díky kterým se můžeme celé odpoledne dmout pýchou. Ve svém běžném zaměstnání, kde navrhuji křemíkové čipy procesorů toho typu, které se používají v počítači Raspberry Pi, a pracuji na nízkourovňovém softwaru, který na těchto procesorech běží, prakticky dostávám peníze za to, že si celý den hraji. Co může být lepší než připravit lidi, aby mohli trávit celou svou kariéru takovým způsobem?

Nejsme přitom v situaci, že by se děti nechtěly v oboru počítačů realizovat. Pořádný impulz jsme dostali před několika lety, kdy projekt Raspberry Pi postupoval docela pomalu. Členové a dobrovolníci nadace vykonávali veškeré vývojové práce na počítači Raspberry Pi během volných večerů a víkendů. Jsme nezisková organizace, takže členové za svou práci pro nadaci nedostávají žádné peníze, a abychom mohli platit své účty, všichni máme řádná zaměstnání. Občas tedy bylo těžké najít dostatečnou motivaci, když jsem večer netoužil po ničem jiném než koukat na sitcom se sklenicí vína v ruce. Když jsem se jednoho večera zrovna nerozvaloval na gauči, dal jsem se do řeči se synovcem svého souseda o předmětech, které si zvolí ke zkouškám GCSE (General Certificate of Secondary Education, britský systém veřejných zkoušek z různých předmětů, které studenti absolvují přibližně od svých 16 let), a zeptal jsem se jej, čím se chce později žít.

„Chci psát počítačové hry,“ prohlásil.

„To je skvělé. Jaký počítač máš doma? Půjčím ti pár knížek o programování, které by tě mohly zajímat.“

Odpověď zněla: „Wii a Xbox.“

Když jsem se ho ještě chvíli vypytał, ukázalo se, že tento chytrý kluk nikdy doopravdy neprogramoval, doma nemá žádný počítač, na kterém by mohl programovat, a jeho školní hodiny informačních a komunikačních technologií – kde měl k dispozici sdílený počítač a dozvěděl se o návrhu webových stránek a práci s tabulkovými a textovými procesory – jej na používání počítače nepřipravily ani na té nejzákladnější úrovni. Byl ovšem nadšenec do počítačových her (a není nic divného, že člověk chce pracovat v oblasti, která ho baví). Doufal tedy, že vybraná témata zkoušky GCSE mu přitom pomohou. Rozhodně měl umělecké sklony, které vyvojáři her potřebují, a ani jeho známky z matematiky a přírodních věd nebyly špatné. Při svém studiu se však nikdy nedostal k programování. V jeho studijním programu žádná taková volba nebyla, pouze podobné kurzy informačních a komunikačních technologií, které se místo programování zaměřují jen na ovládání. A vzhledem k tomu, s jakými počítači si hrál doma, měl jen nepatrnou šanci osvojit si dovednosti, které potřeboval k tomu, aby se uplatnil ve svém vysněném oboru.

Právě takovým situacím, kdy se zbytečně promarní lidský potenciál a nadšení, se snažíme předcházet. Samozřejmě nejsem natolik naivní, abych se domníval, že když nabídneme počítač Raspberry Pi, všechno se změní k lepšímu. Věřím však, že může fungovat jako katalyzátor. Již pozorujeme zásadní změny v osnovách britských škol, kde se do studijního plánu dostává výuka počítačů a dochází k přepracování kurzů informačních a komunikačních technologií. V krátkém čase od uvedení počítače Raspberry Pi jsme si také mohli všimnout toho, že značně vzrostlo povědomí o tom, jaké má naše vzdělávací a kulturní péče o děti mezery.

Příliš mnoho počítačových zařízení, s nimiž děti každodenně interagují, je natolik uzavřených, že s nimi nelze pracovat jako s kreativním nástrojem – ačkoli informatika je kreativní předmět. Zkuste použít svůj iPhone jako mozek robota nebo si na své konzole PS3 zahrát hru, kterou jste sami naprogramovali. Samozřejmě můžete programovat na domácím počítači, ale musíte přitom překonat značné překážky, které mohou být pro mnoho dětí zásadní: děti musí stáh-

nout a nainstalovat speciální software a jejich rodiče se nesmí bát, že děti něco pokazí a oni to nebudou schopni napravit. Hodně dětí navíc vůbec netuší, že něco jako programování domácího počítače je vůbec možné. Domácí počítač berou jako zařízení s hezkými barevnými ikonkami, které jim poskytují snadný přístup k potřebným funkcím, o nichž nemusí příliš přemýšlet. Počítač je uzavřen v zapečetěném krytu, máma s tátou jej používají k elektronickému bankovníctví, a když se rozbije, bude jeho oprava stát spoustu peněz!

Počítač Raspberry Pi je dostatečně levný, aby si jej děti mohly koupit za několikátý denní kapesné, a všechno příslušenství, které ke své činnosti vyžaduje, již v domácnosti pravděpodobně najdete: televizor, kartu SD, kterou lze vytáhnout ze starého fotoaparátu, nabíječku mobilních telefonů, klávesnici a myš. Nepůjdou si jej všichni členové rodiny, ale patří jen dítěti a je dostatečně malý, aby jej dítě mohlo vzít do kapsy a přenést ke kamarádovi. Když se něco pokazí, nic se neděje – stačí vyměnit kartu SD za novou a Raspberry Pi funguje stejně jako po zakoupení. Chcete-li se pustit na dlouhou cestu, na jejímž konci dokážete svůj počítač Raspberry Pi programovat, máte všechny potřebné nástroje, funkce a výukové materiály k dispozici hned poté, co počítač zapnete.

Trocha historie

Na miniaturním, levném a jednoduchém počítači jsem začal pracovat asi před šesti lety, když jsem byl studijním ředitelem pro informatiku na Cambridgeské univerzitě. Studoval jsem v univerzitní počítačové laboratoři a poté jsem tam během práce na svém doktorátu vyučoval. Během tohoto období jsem si všiml patrného poklesu dovedností mladých lidí, kteří se hlásili ke studiu informatiky v naší laboratoři. Od poloviny 90. let, kdy sedmnáctiletí uchazeči o studium informatiky přicházeli se znalostí základů několika počítačových jazyků, měli obecný přehled o zapojení hardwaru, a často dokonce uměli programovat v assembleru, jsme se postupně dostali do situace, kdy v roce 2005 zájemci o studium u nás měli trochu zkušeností s jazykem HTML – a když jsme měli štěstí, občas jim něco říkaly i zkratky PHP a CSS. Pořád to byly mimořádně chytré děti s velkým potenciálem, ale jejich zkušenosti v oblasti počítačů byly naprosto odlišné od toho, co jsme znali z dřívějších let.

Studium informatiky na Cambridgeské univerzitě zabírá asi 60 týdnů přednášek a seminářů v průběhu tří let. Pokud bychom celý první rok věnovali na to, abychom studenty naučili základy oboru, stěží by mohli za další dva roky začít pracovat na titulu PhD. nebo nastoupit do praxe. Nejlepší z našich studentů – ti, kteří měli nejlepší výsledky na konci tříletého kurzu – se vyznačovali tím, že neprogramovali jen tehdy, když to dostali úkolem v rámci svého školního projektu. Byli to ti studenti, kteří programovali ve svém volném čase. Projekt Raspberry Pi byl tedy původně inspirován lokálními potřebami a měl jen omezený a nepřilíš ambiciózní cíl: chtěl jsem vytvořit nástroj, který by malému počtu uchazečů o jeden z mnoha univerzitních kurzů poskytl počáteční impulz. Spolu se svými kolegy jsem si představoval, že tato zařízení při dnech otevřených dveří rozdáme středoškolákům, a když k nám do Cambridge přijdou o několik měsíců později na pohovor, zeptáme se jich, co s tím počítačem, který od nás do-

stali zdarma, udělali. Do našeho programu bychom pak pozvali ty z nich, kteří by s počítačem provedli něco zajímavého. Mysleli jsme, že možná vyrobíme několik set těchto zařízení nebo se v nejlepším případě jejich celkový vyrobený počet bude počítat v tisících.

Jakmile se však práce na projektu rozeběhla naplno, začalo nám být jasné, že takový malý a levný počítač by toho mohl dokázat mnohem více. Na začátku naše představy příliš nepřipomínaly počítač Raspberry Pi, jak vypadá dnes. Začal jsem na svém kuchyňském stole pájet nejdelší zkušební model obvodu, který se prodává v obchodech Maplin, s čipem Atmel. První hrubé prototypy přímo ovládaly televizor se standardním rozlišením pomocí levných jednocipových počítačů. Tyto prototypy s pouhými 512 kB paměti RAM a několika MIPS výpočetního výkonu se svými možnostmi velmi podobaly původním osmibitovým mikropočítačům. Stěží jsme mohli očekávat, že tato zařízení zaujmou děti, které jsou zvyklé na moderní herní konzoly a iPady.

V univerzitní počítačové laboratoři probíhaly obecné diskuze o stavu informatického vzdělávání, a když jsem z laboratoře odešel pracovat do komerční sféry, všiml jsem si, že mladí uchazeči o práci mají stejné potíže, s jakými jsme se setkávali na univerzitě. Sešel jsem se tedy s několika kolegy: dr. Robem Mullinsem a profesorem Alanem Mycroftem (dvěma kolegy s počítačové laboratoře), Jackem Langem (který na univerzitě vyučuje podnikání), Petem Lomasem (hardwarovým guru) a Davidem Brabenem (cambridgeskou autoritou v oblasti počítačových her s neocenitelným adresářem). U piva (a v Jackově případě sýra a vína) jsme pak založili malou nadaci s velkými cíli a nazvali ji Raspberry Pi Foundation.

Proč „Raspberry Pi“?

Často dostáváme otázku, jak vznikl název „Raspberry Pi“. Části toho názvu pocházejí od různých členů dozorčí rady. Jedná se o jeden z mála případů, které znám, kdy úspěšný návrh vytvořila komise, a upřímně řečeno se mi ten název na začátku vůbec nelíbil. (Později jsem si jej zamiloval, protože je opravdu praktický.) Abych si na něj však zvykl, potřeboval jsem nějaký čas, protože jsem projekt mnoho let sám pro sebe označoval jako „ABC Micro“. Slovo „Raspberry“ (malina) jsme zvolili proto, že názvy ovoce mají v počítačových firmách dlouhou tradici. Každý si samozřejmě vzpomene na nakousnuté jablko, ale kdysi se prodávaly také počítače značky Tangerine (mandarinka) a Apricot (meruňka). Mezi ovoce bychom mohli zařadit také Acorn (žalud), což byl britský výrobce počítačů založený roku 1978. „Pi“ je zkomolenina slova „Python“, protože v počátečních fázích vývoje jsme se domnívali, že to bude jediný programovací jazyk dostupný na mnohem méně výkonné platformě, než jakou jsme nakonec v Raspberry Pi získali. Python mimochodem stále doporučujeme jako svůj oblíbený jazyk pro výuku i vývoj, ale platforma Raspberry Pi umožňuje zkoumat i celou řadu dalších jazyků.

Ve své nové roli architekta čipů ve společnosti Broadcom, což je velký výrobce polovodičů, jsem získal přístup k levnému, ale vysoce výkonnému hardwaru, který tato společnost vyrábí pro nasazení ve špičkových mobilních telefonech (takových, které například umožňují nahrávat video HD a pořizovat fotografie s rozlišením 14 megapixelů). Překvapilo mě, jaké zásadní rozdíly existují mezi čipy, které může za 10 dolarů koupit malý vývojář, a čipy, které za přibližně stejnou cenu dokáže pořídit výrobce mobilních telefonů: ty druhé poskytují funkce

univerzálního výpočetního zpracování, 3D grafiky, videa a paměti integrované do jediného pouzdra BGA velikosti nehtu. Tyto mikročipy mají velmi nízkou spotřebu, ale nabízejí značné možnosti. Vynikají zejména v multimédiích a mezi jejich první odběratele patřili výrobci set-top boxů, kteří chtěli umožnit přehrávání videa HD. Vzhledem ke směru, který nabral vývoj počítače Raspberry Pi, byly čipy této třídy jasnou volbou. Snažil jsem se tedy vyvinout levnou variantu s mikroprocesorem ARM na základní desce, která by dokázala zvládnout požadovanou výpočetní zátěž.

Shodli jsme se na tom, že je důležité, aby děti používání počítače Raspberry Pi lákalo, i když se zatím příliš nezajímaly o programování. Když jste si v 80. letech chtěli zahrát počítačovou hru, museli jste nabootovat krabici, která zablikala a zobrazila příkazový řádek. Kvůli spuštění hry bylo potřeba zadat pár příkazů. Většina uživatelů se sice nikdy nedostala dále, někteří však ano. Tato jednoduchá interakce s počítačem je lákala k tomu, aby se začali učit programovat. Uvědomili jsme si, že počítač Raspberry Pi by mohl fungovat jako velmi výkonné, miniaturní a dostatečně levné moderní mediální centrum. Začali jsme proto zdůrazňovat tyto možnosti, abychom přitáhli neopatrné uživatele s nadějí, že pak třeba přičichnou i k programování.

Asi po pěti letech tvrdé dřiny jsme vytvořili docela atraktivní prototyp základní desky velký asi jako flash disk. Na horní část desky jsme napevno přidali modul fotoaparátu, abychom ukázali, jaký druh periférií by bylo možné snadno doplňovat. Prototyp jsme pak brávali na schůzky s pracovníky oddělení výzkumu a vývoje BBC. Ti z nás, kteří vyrůstali v Británii v 80. letech, se hodně naučili díky počítačům BBC Microcomputer (pro BBC je vyráběla společnost Acorn Microcomputers) a díky celému ekosystému, který kolem nich vznikl – z knih, časopisů a televizních pořadů produkovaných společností BBC. Doufal jsem tedy, že lidé v BBC budou mít zájem platformu Raspberry Pi dále rozvíjet. Jak se však ukázalo, od té doby, kdy jsme byli dětmi, se něco změnilo: kvůli různým britským a evropským předpisům na ochranu hospodářské soutěže se BBC nemohla zapojit takovým způsobem, v jaký jsme doufali. V rámci posledního zoufalého pokusu *něco* s BBC podniknout jsme opustili představu o spolupráci s oddělením výzkumu a vývoje a David (ten s tlustým adresářem) v květnu 2011 domluvil schůzku s vedoucím technickým publicistou Rory Cellan-Jonesem. Ten nám na partnerství s BBC nedával příliš nadějí, ale zeptal se nás, zda by mohl svým mobilním telefonem pořídit video našeho malého prototypu, které chtěl umístit na svůj blog.

Druhého dne se jeho video virálním způsobem rozšířilo. Uvědomil jsem si, že jsme před celým světem mimochodem slíbili, že každému poskytneme počítač za 25 dolarů.

Rory se pustil do psaní dalšího příspěvku, kde vysvětloval, co videu poskytuje virální kvality, a my jsme dali hlavy dohromady. Ten původní prototyp velikosti flash disku neodpovídal požadavkům: kvůli standardně integrovanému fotoaparátu byl příliš drahý, než aby se vešel do původního cenového odhadu. (Částka 25 dolarů pocházela z mého výroku pro BBC, že počítač Raspberry Pi by měl stát asi tolik jako učebnice. Dokonale dokumentuje fakt, že nemám představu, jak jsou učebnice v současnosti drahé.) Maličký prototyp také neměl na svém okraji dost místa na všechny porty, které jsme potřebovali zahrnout, aby byl náš počítač natolik praktický, jak jsme si představovali. Strávili jsme tedy další rok návrhem základní desky, který by umožnil co nejvíce snížit náklady, a přitom jsme se snažili udržet všechny požadova-

né funkce (navrhovat s ohledem na nízkou cenu je těžší, než by se mohlo zdát). Zároveň bylo naším záměrem, aby byl počítač Raspberry Pi maximálně užitečný i lidem, kteří si nemohou dovolit mnoho periférií.

Rozhodně jsme chtěli, aby se počítač Raspberry Pi stejně jako počítač ZX Spectrum v 80. letech dal používat s domácími televizory, což by uživatelům ušetřilo výdaj za monitor. Všichni však doma nemají televizi HDMI, takže jsme přidali i kompozitní port, aby bylo možné Raspberry Pi připojit ke starému televizoru s katodovou trubicí. Jako paměťové médium jsme zvolili karty SD, protože jsou levné a snadno dostupné. Při výběru paměťového média jsme zavrhlí karty microSD, protože děti s těmito drobnými kartami velikosti nehtu těžko manipulují a mohou je příliš snadno ztratit. Prošli jsme také několika variantami dodávky napájení, až jsme skončili u kabelu mikro USB. Kabely mikro USB v současnosti slouží jako standardní nabíjecí kabely mobilních telefonů v rámci EU (a jako standard se prosazují i jinde). To znamená, že tyto kabely jsou stále rozšířenější a v mnoha případech je lidé již mají doma.

Koncem roku 2011 se blížilo plánované únorové datum uvedení na trh a začalo nám být zřejmé, že se vše vyvíjí rychleji a poptávka je mnohem vyšší, než jsme vůbec mohli zvládnout. Nejdříve jsme produkt chtěli představit vývojářům a do oblasti vzdělávání jsme chtěli vstoupit v pozdějších měsících roku 2012. Máme malou skupinu nadšených dobrovolníků, ale potřebovali jsme se obrátit na širší linuxovou komunitu, aby nám pomohla připravit sadu softwaru a odladit všechny drobné vady, než náš počítač poskytneme vzdělávacímu trhu. Ve své nadaci jsme měli dostatečný kapitál, abychom mohli nakoupit součástky a sestavit 10 000 počítačů Raspberry Pi asi během jednoho měsíce. Domnívali jsme se, že členů naší komunity, kteří by měli zájem o první verzi základní desky, bude přibližně stejný počet. Naštěstí i naneštěstí jsme byli mimořádně úspěšní při budování velké komunity online kolem našeho zařízení. Zájemci navíc nepocházeli jen z Británie a neomezovali se na oblast vzdělávání. Deset tisíc kusů vypadalo stále méně realisticky.

Na našem distribučním seznamu zájemců o Raspberry Pi bylo 100 000 lidí – a všichni zadali objednávku hned první den! Asi vás nepřekvapí, že nám to způsobilo pár problémů.

V prvé řadě se při balení 100 000 malých počítačů a jejich rozesílání nevyhnete občasnému pořezání – a bohužel jsme neměli peníze, abychom si na to někoho najali. Místo vlastního skladu jsme mohli používat jen Jackovu garáž. Neměli jsme žádnou možnost, jak získat peníze na výrobu všech 100 000 kusů. Představovali jsme si, že budeme počítač vyrábět po dávkách nějakých 2 000 kusů každých několik týdnů. Při aktuálním zájmu o náš produkt by to však trvalo tak dlouho, že by byl zastaralý dříve, než bychom všechny objednávky uspokojili. Bylo jasné, že výrobu a distribuci musíme vzdát a svěřit tyto úkoly někomu, kdo již má potřebnou infrastrukturu a kapitál. Kontaktovali jsme tedy britské dodavatele elektroniky s celosvětovou působností element14 a RS Components a uzavřeli jsme s nimi smlouvu týkající se aspektů výroby a distribuce, abychom se mohli soustředit na vlastní vývoj a charitativní cíle nadace Raspberry Pi Foundation.

Přesto byl počáteční zájem tak velký, že weby společností RS a element14 po většinu prvního dne nefungovaly – v jisté době přicházelo do firmy element14 sedm objednávek za sekundu

a pár hodin během 29. února byl podle vyhledávače Google výraz „Raspberry Pi“ celosvětově populárnější než „Lady Gaga“. Tento text píše začátkem července 2012. Objednávky za první tři měsíce, kdy Raspberry Pi nabízíme, dosáhly půl milionu kusů. Zatím bohužel u žádného dodavatele stále nelze zakoupit více než jeden kus (než nabídnou možnost objednání více kusů, snaží se distributoři vyřešit nahromaděné nevyřízené objednávky). Pokud bychom se drželi našich původních plánů, vyrobili bychom asi 100 těchto přístrojů pro univerzitní dny otevřených dveří – a tím by to skončilo.

Naše komunita

Komunita uživatelů Raspberry Pi patří mezi naše největší úspěchy. Hned po zveřejnění Roryho videa v květnu 2011 jsme začali velmi jednoduchým blogem na adrese <http://www.raspberrypi.org> a brzy poté jsme na stejném webu zpřístupnili i fórum. Toto fórum má nyní více než 20 000 členů, kteří o Raspberry Pi společně napsali více než 100 000 vtipných i informativních příspěvků. Pokud máte libovolně náročnou otázku týkající se počítače Raspberry Pi nebo programování obecně, někdo vám ji tam zodpoví (nenajdete-li odpověď v této knize, na fóru určitě).

V rámci své práce na počítači Raspberry Pi vystupuji na setkáních hackerů, na počítačových konferencích, před učiteli, týmy programátorů atp. Téměř pokaždé se v publiku najde někdo, kdo na webu Raspberry Pi diskutoval se mnou nebo mou ženou Liz (která komunitu spravuje). Někteří z těchto lidí se postupně stali našimi dobrými přáteli. Web Raspberry Pi každou sekundu obsluhuje přibližně jeden požadavek.

V současnosti také existují stovky fanouškovských webů. Fanoušci mohou číst časopis *The MagPi* (bezplatně ke stažení na adrese <http://www.themagpi.com>), který každý měsíc vydávají členové komunity. Obsahuje zdrojové kódy programů, mnoho článků, projektové příručky, kurzy a další informace. Díky zdrojovým kódům her v časopisech a knihách jsem se snadno naučil programovat – své první programátorské zkušenosti s počítačem BBC Micro jsem získal při úpravách kódu vrtulníkové hry, kam jsem přidával nepřátele a zastávky.

Na webu <http://www.raspberrypi.org> zveřejňujeme o našem zařízení něco zajímavého alespoň jednou denně. Navštivte jej a zapojte se do konverzace!

Z vlastní zkušenosti mohu říci, že nic vám nezvedne krevní tlak tolik, jako když máte najednou řídit velkou počítačovou firmu!

K čemu je tedy počítač Raspberry Pi dobrý?

V této knize ukážeme různé věci, které lze s počítačem Raspberry Pi dělat – od řízení hardwaru pomocí kódu jazyka Python přes použití počítače jako mediálního centra až po tvorbu her v jazyce Scratch. Působení zařízení Raspberry Pi spočívá v tom, že se jedná o univerzální počítač nevelkých rozměrů (v kancelářských aplikacích sice může být poněkud pomalejší než standardní PC, ale v jiných aplikacích je může značně předčit). Zvládne tedy cokoli, co dokáže běžný počítač. Kromě toho poskytuje počítač Raspberry Pi široké možnosti v oblasti multimédií a 3D grafiky, takže má potenciál nasazení jako herní platforma. Pevně doufáme, že uživatelé pro něj začnou psát hry.

Domníváme se, že fyzické aplikace počítačů – budování systémů se senzory, motory, světly a mikrokontroléry – jsou v mnoha případech přehlíženy ve prospěch čistě softwarových projektů, což je škoda, protože jsou *mimořádně zábavné*. Pokud v současnosti existuje nějaké silnější počítačové hnutí určené pro děti, jedná se o projekty fyzických aplikací počítačů. Želvy LOGO, které reprezentovaly fyzické aplikace počítačů v dobách našeho dětství, se nyní změnilly na válečné roboty, helikoptéry se čtyřmi rotory nebo rodičovské senzory na dveřích dětského pokoje. Děti takové hračky milují. Mnohé zájemce o robotické projekty však odrazují chybějící porty GPIO (General Purpose Input/Output) u domácích počítačů. Počítač Raspberry Pi poskytuje piny GPIO, které jsou připraveny k použití.

Členové naší komunity mě neustále překvapují svými ideami, které by mě nenapadly ani ve snu: australská škola svým projektem sledování meteorů, v Británii oddíl skautů se svým robotem, který je řízen pomocí elektroencefalografické náhlavní sady (první robot na světě s ovládáním pomocí mozkových vln), nebo rodina, která buduje robotický vysavač. Osobně mám slabost pro lety do vesmíru, takže mi naskakuje husí kůže, když se dozvídám o uživatelích, kteří své Raspberry Pi vystřelují raketami a vypouštějí v balónech.

Za úspěch bychom považovali, kdyby v Británii každým rokem nastoupilo ke studiu informatiky o 1 000 zájemců více. Pomohlo by to nejen celé zemi, softwarovému a hardwarovému průmyslu a ekonomice, ale ještě více by to prospělo každému z této tisícovky. Doufám, že tito noví informatici zjistí, že je čeká celý svět plný nových možností a spousta zábavy. Když jako dítě postavíte robota, můžete se dostat na místa, která jste si do té doby nedokázali představit. Vím to, protože mně samotnému se to podařilo!

Eben Upton

Zpětná vazba od čtenářů

Nakladatelství a vydavatelství Computer Press, které pro vás tuto knihu přeložilo, stojí o zpětnou vazbu a bude na vaše podněty a dotazy reagovat. Můžete se obrátit na následující adresy:

Computer Press
Albatros Media a.s., pobočka Brno
IBC
Příkop 4
602 00 Brno

nebo

sefredaktor.pc@albatrosmedia.cz

Computer Press neposkytuje rady ani jakýkoli servis pro aplikace třetích stran. Pokud budete mít dotaz k programu, obraťte se prosím na jeho tvůrce.

Zdrojové kódy ke knize

Z adresy <http://knihy.cpress.cz/K2123> si po klepnutí na odkaz Soubory ke stažení můžete přímo stáhnout archiv s ukázkovými kódy.

Errata

Přestože jsme udělali maximum pro to, abychom zajistili přesnost a správnost obsahu, chybám se úplně vyhnout nelze. Pokud v některé z našich knih najdete chybu, ať už chybu v textu nebo v kódu, budeme rádi, pokud nám ji oznámíte. Ostatní uživatelé tak můžete ušetřit frustrace a nám můžete pomoci zlepšit následující vydání této knihy.

Veškerá existující errata zobrazíte na adrese <http://knihy.cpress.cz/K2123> po klepnutí na odkaz Soubory ke stažení.



PŘIPOJENÍ ZÁKLADNÍ DESKY

V této části:

- **KAPITOLA 1** – Seznámení s počítačem Raspberry Pi
- **KAPITOLA 2** – Správa systému Linux
- **KAPITOLA 3** – Řešení potíží
- **KAPITOLA 4** – Konfigurace sítě
- **KAPITOLA 5** – Správa oddílů
- **KAPITOLA 6** – Konfigurace počítače Raspberry Pi

Seznámení s počítačem Raspberry Pi

Základní deska počítače Raspberry Pi je zázrak miniaturizace, který ukrývá značný výpočetní výkon v rozměrech nepřesahujících platební kartu. Dokáže úžasné věci, ale než se pustíte do experimentování, měli byste se o ní něco málo dozvědět.



Tip: Pokud se nemůžete dočkat, až s počítačem začnete pracovat, přeskočte o několik stránek dopředu, kde naleznete informace o připojení počítače Raspberry Pi k displeji, klávesnici a myši.

ARM versus x86

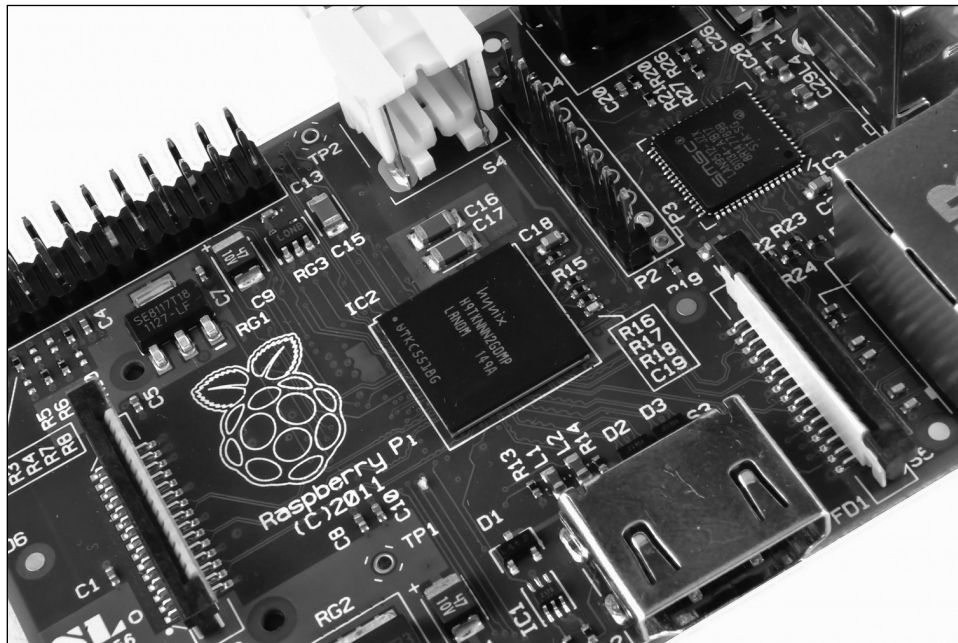
Jádrem systému Raspberry Pi je multimediální procesor typu SoC (system-on-chip) Broadcom BCM2835. To znamená, že většina systémových komponent, včetně jeho hlavního a grafického procesoru spolu se zvukovým a komunikačním hardwarem, je integrována do jediné součástky ukryté pod paměťovým čipem s kapacitou 256 MB uprostřed základní desky (viz obrázek 1.1).

Procesor BCM2835 se od procesorů, kterými jsou vybaveny stolní počítače nebo notebooky, však neliší jen svým návrhem typu SoC. Používá také jinou architekturu instrukční sady (ISA – instruction set architecture), která se označuje jako ARM.

Architektura ARM, kterou již v 80. letech vyvinula společnost Acorn Computers, se v počítačích třídy PC uplatňuje poměrně zřídka. Vyniká však v mobilních zařízeních: telefon, který nosíte v kapse, je téměř určitě vybaven alespoň jedním výpočetním jádrem typu ARM. Vzhledem ke kombinaci jednoduché architektury s redukovanou instrukční sadou (RISC – reduced instruction set) a nízké spotřeby energie představuje v mobilních zařízeních mnohem výhodnější volbu než procesory stolních počítačů, které se vyznačují vysokými nároky na napájení a architekturou s komplexní instrukční sadou (CISC – complex instruction set).

Procesor BCM2835 třídy ARM vysvětluje, jak je možné, že počítač Raspberry Pi dokáže fungovat se zdrojem napájení s napětím jen 5 V a proudem 1 A, který poskytuje integrovaný port mikro USB. Ze stejného důvodu také na zařízení nenajdete žádné tepelné jímky: díky nízké

spotřebě energie procesor produkuje velmi málo odpadního tepla, dokonce i během komplikovaných výpočetních operací.



Obrázek 1.1: Procesor BCM2835 typu SoC, který je umístěn pod paměťovým čipem Hynix

Zároveň to však znamená, že počítač Raspberry Pi není kompatibilní s tradičním softwarem pro PC. Většina programů pro stolní počítače a notebooky odpovídá architektuře s instrukční sadou x86, na které jsou založeny procesory značek AMD, Intel a VIA. Z tohoto důvodu tyto programy v počítači Raspberry Pi s procesorem typu ARM nemohou fungovat.

Procesor BCM2835 vychází z varianty návrhu procesoru ARM, která se označuje jako ARM11. Tato varianta je zase navržena podle verze architektury instrukční sady ARMv6. Je dobré si uvědomit, že architektura ARMv6 je jednoduchá a výkonná, ale konkuruje jí pokročilejší architektura ARMv7, která se používá v procesorech třídy ARM Cortex. Obdobně jako software vyvinutý pro procesory x86 ani software určený pro architekturu ARMv7 bohužel není kompatibilní s procesorem BCM2835 v počítači Raspberry Pi, ačkoli vývojáři mohou obvykle své programy konvertovat tak, aby se na tomto procesoru spustit daly.

To ovšem neznamená, že se při výběru programů budete muset omezovat. Jak zjistíte v další části této knihy, pro instrukční sadu ARMv6 je k dispozici spousta softwaru, a jak vzrůstá popularita počítače Raspberry Pi, nabídka se bude jen rozšiřovat. V této knize se také naučíte, jak tvořit vlastní software pro počítač Pi, i když zatím s programováním nemáte žádné zkušenosti.

Windows versus Linux

Další důležitý rozdíl mezi počítačem Raspberry Pi a stolním počítačem nebo notebookem spočívá kromě velikosti a ceny také v operačním systému – softwaru, který umožňuje řídit činnost počítače.

Většina stolních i přenosných počítačů, které se v současnosti prodávají, pracuje s jedním ze dvou operačních systémů: Microsoft Windows nebo Apple OS X. Obě platformy patří do kategorie *closed source* a vznikají v nepřístupném prostředí s použitím proprietárních metod.

Tyto operační systémy se označují jako *closed source* z povahy svého *zdrojového kódu* (source code), což je popis v počítačovém jazyce, který systému sděluje, co má dělat. V případě softwaru typu *closed source* tento popis zůstává pečlivě hlídaným tajemstvím. Uživatelé mohou získat hotový software, ale nemají možnost se podívat, jak je vytvořen.

Návrh počítače Raspberry Pi oproti tomu počítá s tím, že na něm bude fungovat operační systém s názvem GNU/Linux, který budeme dále označovat prostě jako Linux. Na rozdíl od systémů Windows či OS X je systém Linux systémem typu *open source*: můžete si stáhnout zdrojový kód celého operačního systému a změnit cokoli, co chcete. Nic není skryté a všechny změny kompletně probíhají na očích veřejnosti. Díky tomuto stylu vývoje *open source* bylo možné standardní systém Linux rychle přizpůsobit ke spuštění v počítači Raspberry Pi v rámci procesu, který se označuje jako *portování*. V době psaní této knihy bylo na čip BCM2835 počítače Raspberry Pi portováno několik verzí systému Linux, které se nazývají *distribuce*. Patří k nim Debian, Fedora Remix a Arch Linux.

Jednotlivé distribuce se zaměřují na odlišné uživatelské požadavky, ale všem je společné to, že se jedná o projekty *open source*. Vesměs jsou také vzájemně kompatibilní: software napsaný v systému Debian bude dokonale fungovat i v systému Arch Linux a naopak.

Systém Linux není omezen jen na počítače Raspberry Pi. K dispozici jsou stovky různých distribucí určených pro stolní počítače, notebooky, a dokonce mobilní zařízení. Z jádra Linux vychází i oblíbená platforma Android společnosti Google. Pokud budete se systémem Linux ve svém počítači Raspberry Pi spokojeni, můžete jej nainstalovat i do jiných počítačů, které používáte. Tento systém dokáže bez problémů koexistovat se stávajícím operačním systémem, a můžete tak využívat výhody obou a mít k dispozici známé prostředí i tehdy, když nemáte po ruce svůj počítač Pi.

Stejně jako v případě rozdílů mezi procesory ARM a x86 je potřeba zmínit důležitý fakt týkající se praktické odlišnosti systémů Windows, OS X a Linux: software určený pro systém Windows či OS X nebude v systému Linux fungovat. Pro naprostou většinu běžných softwarových produktů však naštěstí existuje mnoho odpovídajících alternativ. Většinu z nich lze navíc používat zdarma a patří do kategorie *open source* stejně jako samotný operační systém.

Začínáme s počítačem Raspberry Pi

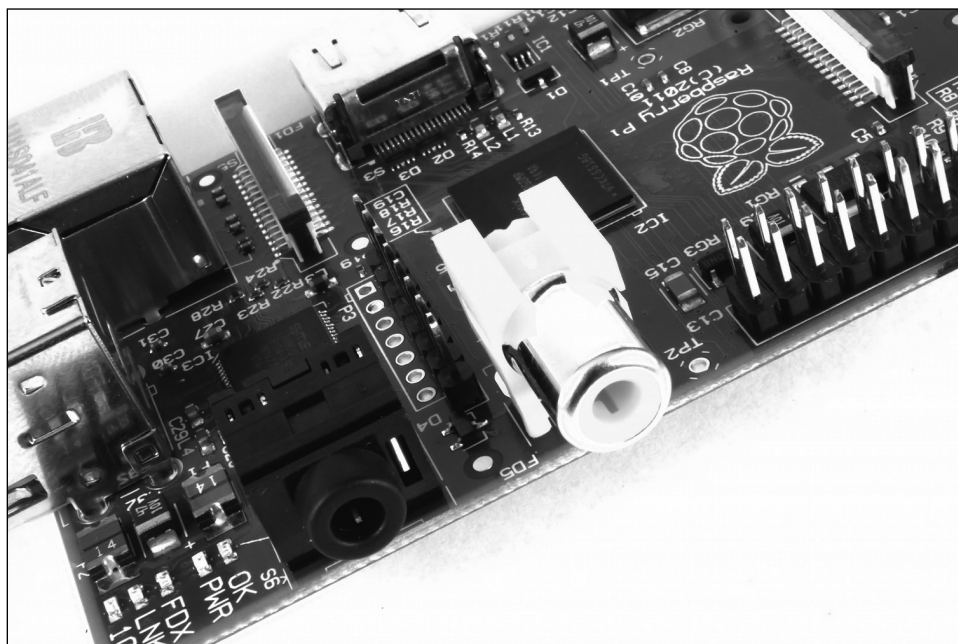
Když nyní rozumíte tomu, jak se počítač Pi liší od jiných výpočetních zařízení, můžete jej začít používat. Pokud jste svůj počítač Pi právě získali, vyjměte jej z ochranného antistatického obalu a položte jej na rovný a nevodivý povrch. Potom můžete pokračovat ve čtení této kapitoly.

Připojení displeje

Abyste mohli s počítačem Raspberry Pi pracovat, musíte jej připojit k zobrazovacímu zařízení. Pi nabízí tři různé výstupy videa: kompozitní video, video HDMI a video DSI. Jak ukážeme v této části, kompozitní video a video HDMI jsou pro koncového uživatele snadno přístupné, zatímco video DSI vyžaduje specializovanější hardware.

Kompozitní video

Kompozitní video je dostupné díky žluto-stříbrnému portu v horní části počítače Pi, který se označuje jako *konektor RCA* (RCA phono – viz obrázek 1.2). Tento port slouží k připojení počítače Raspberry Pi ke starším zobrazovacím zařízením. Jak je patrné z jeho názvu, konektor kombinuje barvy obsažené v obrazu – červenou, zelenou a modrou složku – a přenáší je po jediném vodiči do zobrazovacího zařízení, což je zpravidla starý televizor typu CRT (s katodovou trubicí).



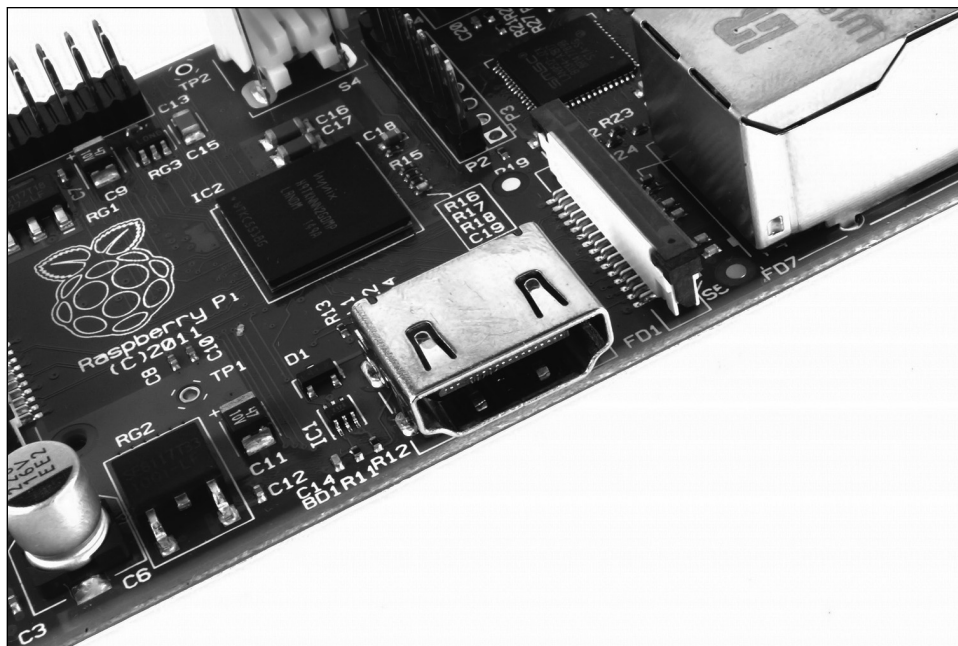
Obrázek 1.2: Žlutý konektor RCA pro výstup kompozitního videa

Nemáte-li k dispozici žádné jiné zobrazovací zařízení, můžete s počítačem Pi začít pracovat díky připojení kompozitního videa. Kvalita obrazu však není příliš vysoká. Připojení kompozitního videa je značně náchylnější k rušení, obraz bývá nezřetelný a vyznačuje se omezeným rozlišením, což znamená, že se na obrazovku vejde méně ikon a řádků textu.

Video HDMI

Kvalitnější obraz získáte pomocí konektoru *HDMI* (*High Definition Multimedia Interface*), což je jediný port v dolní části počítače Pi (viz obrázek 1.3). Na rozdíl od analogového kompozitního připojení poskytuje port HDMI vysokorychlostní digitální připojení, které umožňuje dokonale přenášet jednotlivé pixely jak do počítačových monitorů, tak do televizorů s vysokým rozlišením. Pomocí portu HDMI může Pi promítat obraz v plném rozlišení HD 1920x1080 většiny moderních přijímačů HDTV. Při tomto rozlišení je na obrazovce k dispozici mnohem více podrobností.

Jestliže jste doufali, že počítač Pi připojíte ke svému stávajícímu počítačovému monitoru, možná zjistíte, že nemá vstup HDMI. Nic se neděje: digitální signály přenášené kabelem HDMI lze mapovat na běžný standard pro připojení počítačových monitorů, který se označuje jako *DVI* (*Digital Video Interconnect*). Stačí koupit převodní kabel HDMI-DVI a dokážete port HDMI svého počítače Pi připojit k monitoru s konektorem DVI-D.



Obrázek 1.3: Stříbrný konektor HDMI pro výstup videa ve vysokém rozlišení

Pokud máte monitor se vstupem VGA (konektor ve tvaru D s 15 piny, obvykle v modré a stříbrné barvě), počítač Raspberry Pi k němu nelze připojit. Existují sice adaptéry, které přijímají

digitální signál DVI a převádějí jej na analogový signál VGA, jsou však rozměrné a drahé. V tomto případě je nejvhodnější pořídit modernější monitor se vstupem DVI nebo HDMI.

Video DSI

Poslední výstup videa počítače Pi najdete nad patičí karty SD v horní části tištěné desky s tištěnými spoji. Konektor vypadá jako malá stužka chráněná vrstvou plastu. Konektor odpovídá standardu videa s názvem *DSI (Display Serial Interface)*. Toto rozhraní slouží k připojení plochých displejů tabletů a smartphonů. Displeje s konektorem DSI se v maloobchodě těžko shánějí a zpravidla jsou určeny pro techniky, kteří chtějí vytvořit kompaktní a autonomní systém. Displej standardu DSI lze připojit vložením plochého kabelu do odpovídajícího konektoru počítače Pi, ale pro začátečníky je vhodnější použít konektor kompozitního videa nebo HDMI.

Zvukové připojení

Jestliže používáte HDMI port počítače Raspberry Pi, nemusíte se zvukovým výstupem příliš zabývat: při správné konfiguraci přenáší port HDMI signál videa i digitální zvukový signál. To znamená, že můžete své zobrazovací zařízení připojit jediným kabelem a těšit se zvukem i obrazem.

Za předpokladu, že počítač Pi připojujete ke standardnímu displeji HDMI, jste v této fázi prakticky hotovi. Stačí jen zasunout kabel do konektorů.

Pokud používáte počítač Pi s monitorem DVD-D připojeným pomocí adaptéru či kabelu, zvuk neuslyšíte. Z toho je zřejmé, v čem spočívá hlavní rozdíl mezi rozhraními HDMI a DVI: zatímco HDMI dokáže přenášet zvukové signály, DVI nikoli.

Máte-li monitor typu DVI-D nebo používáte-li výstup kompozitního videa, získáte analogový zvukový signál pomocí černého konektoru *audio jack s průměrem 3,5 mm*, který se nachází na horní hraně počítače Pi vedle žlutého konektoru RCA (viz obrázek 1.2). Stejný konektor slouží k připojení sluchátek a mikrofonů spotřební zvukové elektroniky a má přesně stejné zapojení. Chcete-li zvukový výstup rychle zprovoznit, stačí, když do tohoto konektoru připojíte sluchátka.



Tip: Sluchátka lze sice připojit přímo k počítači Raspberry Pi, ale možná zjistíte, že zvuk není příliš hlasitý. Máte-li možnost, zvolte raději napájená sluchátka. Jejich vestavěný zesilovač pomůže zlepšit slyšitelnost signálu.

Pokud chcete zajistit robustnější zvukový výstup, můžete sáhnout po standardních počítačových reproduktorech s konektorem průměru 3,5 mm nebo si můžete koupit kabely adaptéru. Uživatelům kompozitního videa se bude hodit kabel adaptéru z audio jacku 3,5 mm na konektor RCA. Tento kabel je vybaven dvěma konektory RCA (červený a bílý), vedle nichž je dostupné připojení videa. Každý z těchto konektorů přenáší do televizoru jeden kanál stereofonního zvukového signálu.

Uživatelé, kteří připojují počítač Pi k zesilovači nebo stereosystému, budou potřebovat buď kabel adaptéru z audio jacku 3,5 mm na konektor RCA, nebo adaptér mezi dvěma audio jac-

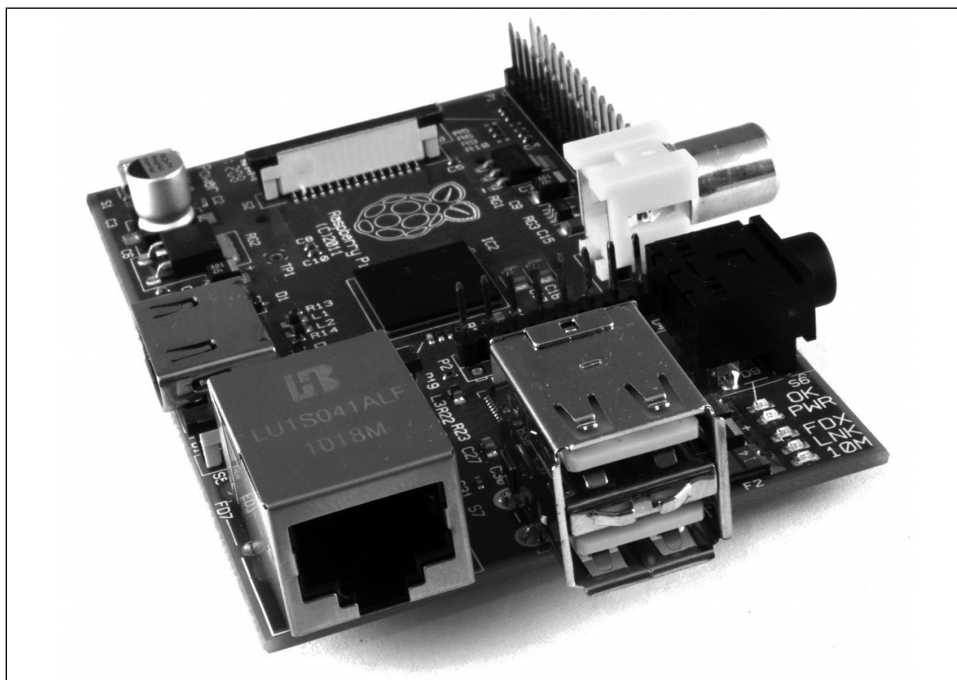
ky 3,5 mm – v závislosti na tom, které systémové konektory jsou volné. Oba typy kabelu lze za nevelký obnos snadno zakoupit v libovolném obchodě se spotřební elektronikou nebo je můžete ještě levněji objednat u dodavatelů online, jako je Amazon.

Připojení klávesnice a myši

Když jste nyní vyřešili připojení výstupních zařízení počítače Raspberry Pi, je na čase podívat se na vstup. Naprostou nezbytností je klávesnice a většina uživatelů se neobejde ani bez myši či trackballu.

Nejdříve vás čekají špatné zprávy: máte-li klávesnici a myš s konektorem PS/2 (kulatým konektorem se sadou pinů ve tvaru podkovy) musíte se vypravit do obchodu a investovat do nových vstupních zařízení. Připojení PS/2 je již překonané a počítač Pi očekává, že periferie budou připojeny pomocí *portu USB (Universal Serial Bus)*.

Podle toho, zda jste zakoupili model A či model B, máte na pravé straně počítače Pi dostupný jeden nebo dva porty USB (viz obrázek 1.4). Jestliže máte model B, lze klávesnici a myš připojit k těmto portům přímo. Uživatelé modelu A, kteří chtějí připojit dvě zařízení USB současně, si musí pořídit rozbočovač USB (USB hub).



Obrázek 1.4: Dva porty USB modelu B

Rozbočovač USB se vyplatí každému uživateli počítače Pi: i když máte model B, obsadíte oba dostupné porty jen připojením klávesnice a myši a nezůstane vám žádný volný port k připoje-

ní dalších zařízení, jako je externí optická jednotka, úložné zařízení nebo joystick. Při výběru rozbočovače USB dávejte pozor, aby byl napájený: pasivní modely jsou levnější a menší, ale nedokáží obsluhovat zařízení náročná na napájení, jako jsou jednotky CD a externí pevné disky.



Tip: Chcete-li omezit počet obsazených elektrických zásuvek, připojte přívod napájení USB počítače Raspberry Pi k napájenému rozbočovači USB. Počítač Pi tak bude napájen přímo z rozbočovače a nebude potřebovat vlastní elektrickou zásuvku a síťový adaptér. Tento postup funguje jen u rozbočovačů, které dokáží portu USB počítače Pi poskytnout proud 700 mA a zároveň vyhovět požadavkům na napájení jiných zařízení.

Připojení klávesnice a myši je velmi jednoduché. Stačí připojit jejich kabely k portům USB, ať už přímo v případě modelu B, nebo pomocí rozbočovače USB v případě modelu A.

Poznámka týkající se ukládání

Pravděpodobně jste si již všimli, že počítač Raspberry Pi nemá tradiční pevný disk. Místo toho pracuje s *paměťovými kartami SD (Secure Digital)*, což je polovodičový úložný systém, který se běžně uplatňuje v digitálních fotoaparátech. Do počítače Raspberry Pi lze vložit téměř libovolnou kartu SD, ale vzhledem k tomu, že karta uchovává celý operační systém, musí mít kvůli uložení všech potřebných souborů kapacitu alespoň 2 GB.

Karty SD s předinstalovaným operačním systémem jsou dostupné na oficiálním prodejním webu Raspberry Pi Store i v mnoha dalších internetových obchodech. Pokud jste si pořídili jednu z těchto karet nebo ji dostali přibalenu k počítači Pi, stačí, když ji zasunete do patice karet SD na dolním konci levého okraje. V opačném případě musíte kvůli zprovoznění počítače nejdříve nainstalovat operační systém, což se označuje jako *nahrání karty flash (flashing)*.

Některé karty SD fungují spolehlivěji než jiné a existují typy, které s počítačem Raspberry Pi úplně odmítají komunikovat. Aktuální seznam modelů karet SD, které byly s počítačem Pi úspěšně vyzkoušeny, naleznete na stránce eLinux wiki: http://www.elinux.org/RPi_VerifiedPeripherals#SD_cards.

Nahrání systému na kartu SD

Chcete-li připravit prázdnou kartu SD pro použití v počítači Raspberry Pi, musíte na kartu *nahrát* operační systém. Je to sice poněkud komplikovanější než pouhé přetažení souborů na kartu, ale nemělo by vám to zabrat více než několik minut.

V prvé řadě se musíte rozhodnout, kterou linuxovou distribuci chcete se svým počítačem Raspberry Pi používat. Každá má své výhody a nevýhody. Nemějte obavy, že se časem rozmyslíte a budete svého výběru distribuce litovat: kartu SD můžete kdykoli znovu přehrát novým operačním systémem.

Nejaktuálnější seznam vydání systému Linux, která jsou kompatibilní s počítačem Pi, naleznete na webu Raspberry Pi na adrese <http://www.raspberrypi.org/downloads>.

Naše nadace poskytuje odkazy *BitTorrent* na všechny distribuce. Jedná se o malé soubory, které umožňují pomocí softwaru BitTorrent stáhnout cílové soubory od jiných uživatelů. Tyto sou-

bory nabízejí efektivní a rychlý způsob distribuce velkých souborů a zajišťují, aby souborové servery nadace nebyly přetěžovány.

Chcete-li použít odkaz BitTorrent, potřebujete nejdříve nainstalovat kompatibilního *klienta*. Jestliže jste klientský software protokolu BitTorrent zatím neinstalovali, stáhněte jej a nainstalujte dříve, než se pokusíte stáhnout některou z distribucí Linuxu pro Raspberry Pi. Ke klientům pro operační systémy Windows, OS X a Linux patří *µTorrent*, který je k dispozici na adrese <http://www.utorrent.com/downloads>.

Výběr distribuce ke stažení je jen na vás. Pokyny ve zbývajících částech knihy budou vycházet z distribuce Debian Raspberry Pi (Raspbian), která představuje dobrou volbu pro začátečníky. Tam, kde to bude možné, budeme uvádět pokyny i pro jiné distribuce.

Distribuce Linuxu pro počítač Raspberry Pi jsou dostupné jako jednotlivé *soubory bitových kopií* (image file), které jsou kvůli rychlejšímu stahování komprimovány. Jakmile stáhnete archiv ZIP (komprimovaný soubor, jehož stahování trvá kratší dobu, než by trvalo stažení nekomprimovaného souboru) své zvolené distribuce, musíte jej na nějakém místě svého systému dekomprimovat. Ve většině operačních systémů lze soubor jednoduše otevřít poklepáním a poté rozbalit jeho obsah příkazem Extract (Extrahovat) nebo Unzip (Rozbalit).

Po dekomprimaci archivu dostanete dva samostatné soubory. Soubor s příponou *sha1* obsahuje *hodnotu hash*, která umožňuje ověřit, zda nebyl stažený soubor při přenosu poškozen. Soubor s příponou *img* obsahuje přesnou kopii karty SD, kterou tvůrci distribuce vytvořili tak, aby byla kompatibilní s počítačem Raspberry Pi. Tento soubor je nutné nahrát na kartu SD.



Upozornění: V následujícím postupu budete pracovat se softwarovým nástrojem, který se nazývá *dd*. Při nesprávném použití může nástroj *dd* bez varování zapsat bitovou kopii na hlavní pevný disk a tím vymazat operační systém i všechna uložená data. Pozorně si přečtěte pokyny v následující části a pečlivě si poznamenejte adresu zařízení své karty SD. Dvakrát měř a jednou řež!

Nahrání systému na kartu SD z Linuxu

Pokud již v počítači PC máte nějakou variantu systému Linux, můžete zapsat obsah souboru bitové kopie na kartu SD pomocí příkazu *dd*. Tento program s textovým rozhraním se spouští z příkazového řádku, který se v linuxové terminologii označuje jako *terminál*. Chcete-li nahrát systém na kartu SD, postupujte takto:

1. Z nabídky aplikací své distribuce spusťte terminál.
2. Vložte prázdnou kartu SD do čtečky karet připojené k PC.
3. Zobrazte seznam disků příkazem `sudo fdisk -l`. Vyhledejte kartu SD podle její velikosti a poznamenejte si adresu zařízení: `/dev/sdX`, kde *X* je písmeno identifikující úložné zařízení. Některé systémy s integrovanými čtečkami karet SD mohou používat alternativní formát `/dev/mmcblkX` – v takovém případě nezapomeňte odpovídajícím způsobem změnit cíl v následujících pokynech.

4. Pomocí příkazu `cd` přejděte do adresáře se souborem typu `img`, který jste extrahovali z archivu Zip.
5. Příkazem `sudo dd if=soubor_bitové_kopie.img of=/dev/sdX bs=2M` zapíšete soubor `soubor_bitové_kopie.img` na kartu SD připojenou na adrese zařízení z kroku 3. Řetězec `soubor_bitové_kopie.img` nahraďte skutečným názvem souboru, který jste rozbalili z archivu Zip. Tento krok může chvíli trvat. Buďte proto trpěliví. Během nahrávání program nic nevypisuje na obrazovku. Zprávu zobrazí teprve poté, co je proces plně dokončen (viz obrázek 1.5).

```

pi@Virtual-Ubuntu: ~
Disk /dev/sda: 21.5 GB, 21474836480 bytes
hlav: 255, sektorů na stopu: 63, cylindrů: 2 610, celkem 41 943 040 sektorů
Jednotky = sektory po 1 * 512 = 512 bajtech
Velikost sektoru (logického/fyzického): 512 bajtů / 512 bajtů
Velikost I/O (minimální/optimální): 512 bajtů / 512 bajtů
Identifikátor disku: 0x0000f8ce

Zařízení Zavádět   Začátek       Konec        Bloky      Id  Systém
/dev/sda1      *           2048         37748735    18873344   83   Linux
/dev/sda2      37750782    41940991    2095105     5   Rozšířený
/dev/sda5      37750784    41940991    2095104     82   Linux swap/Solaris

Disk /dev/sdb: 2 062 MB, 2 062 548 992 bajtů
hlav: 16, sektorů na stopu: 32, cylindrů: 7 868, celkem 4 028 416 sektorů
Jednotky = sektory po 1 * 512 = 512 bajtech
Velikost sektoru (logického/fyzického): 512 bajtů / 512 bajtů
Velikost I/O (minimální/optimální): 512 bajtů / 512 bajtů
Identifikátor disku: 0x00000000

Zařízení Zavádět   Začátek       Konec        Bloky      Id  Systém
/dev/sdb1      *              32         4028415    2014192     b   W95 FAT32
pi@Virtual-Ubuntu:~$ sudo dd if=raspbian.img of=/dev/sdb1 bs=2M

```

Obrázek 1.5: Nahrání systému na kartu SD pomocí příkazu `dd` v systému Linux

Nahrání systému na kartu SD ze systému OS X

Jestliže máte k dispozici počítač Mac se systémem Apple OS X, určitě vás potěší, že celý postup je stejně jednoduchý jako v Linuxu. Díky společnému původu systémů OS X i Linux je příkaz `dd` součástí obou z nich. Pomocí tohoto příkazu pak můžete nahrát bitovou kopii systému na prázdnou kartu SD následovně:

1. Z nabídky Applications (Aplikace) vyberte příkaz Utilities (Nástroje) a klepněte na aplikaci Terminal (Terminál).
2. Vložte prázdnou kartu SD do čtečky karet připojené k počítači Mac.
3. Zobrazte seznam disků příkazem `diskutil list`. Vyhledejte kartu SD podle její velikosti a poznamenejte si adresu zařízení (`/dev/diskX`, kde `X` je písmeno identifikující úložné zařízení).
4. Pokud byla karta SD automaticky připojena (příkazem `mount`) a zobrazena na ploše, nejdříve ji odpojte příkazem `diskutil unmountdisk /dev/diskX`.
5. Pomocí příkazu `cd` přejděte do adresáře se souborem typu `img`, který jste extrahovali z archivu Zip.

6. Příkazem `dd if=soubor_bitové_kopie.img of=/dev/diskX bs=2m` zapíšete soubor *soubor_bitové_kopie.img* na kartu SD připojenou na adrese zařízení z kroku 3. Řetězec *soubor_bitové_kopie.img* nahraďte skutečným názvem souboru, který jste rozbalili z archivu Zip. Tento krok může chvíli trvat. Buďte proto trpěliví.

Nahrání systému na kartu SD ze systému Windows

Pokud máte ve svém stolním počítači nainstalován systém Windows, je postup poněkud složitější než v systémech Linux nebo OS X. Systém Windows neobsahuje nástroj jako `dd`, takže je k nahrání souboru bitové kopie na kartu SD potřeba získat software jiného dodavatele. Lze sice nainstalovat verzi nástroje `dd` kompatibilní se systémem Windows, ale snadnější způsob nabízí program Win32 Disk Imager. Tento nástroj byl navržen zejména kvůli vytváření bitových kopií linuxových distribucí na flash discích nebo kartách SD. Poskytuje jednoduché uživatelské rozhraní, díky němuž lze kartu SD pro počítač Raspberry Pi připravit velmi jednoduše.

Nejnovější verzi programu Win32 Disk Imager naleznete na oficiálním webu: <http://sourceforge.net/projects/win32diskimager/>. Následujícím postupem lze stáhnout, nainstalovat a spustit program Win32 Disk Imager, který umožní vytvořit kartu SD pro počítač Pi:

1. Stáhněte si soubor Zip s *binární* verzí programu Win32 Disk Imager (nikoli jeho *zdrojovým kódem*) a extrahujte archiv do složky v počítači.
2. Vložte prázdnou kartu SD do čtečky karet připojené k PC.
3. Poklepáním na soubor `Win32DiskImager.exe` spusťte program a klepnutím na modrou ikonu složky otevřete dialogové okno, které umožňuje procházet soubory.
4. Přejděte na soubor *soubor_bitové_kopie.img*, který jste extrahovali z distribučního archivu. Řetězec *soubor_bitové_kopie.img* přitom nahraďte skutečným názvem souboru, který jste rozbalili z archivu Zip, a poté klepněte na tlačítko Open (Otevřít).
5. Z rozevíracího seznamu Device vyberte písmeno jednotky, které odpovídá připojené kartě SD. Pokud si nejste jisti, které písmeno zvolit, zkontrolujte přiřazení jednotek v okně Tento počítač nebo Průzkumníku Windows.
6. Soubor bitové kopie nahrajete na kartu SD klepnutím na tlačítko Write. Tento proces může chvíli trvat. Buďte proto trpěliví.



Upozornění: Bez ohledu na to, ve kterém systému kartu SD vytváříte, je důležité, abyste ji ponechali připojenou až do chvíle, kdy bude bitová kopie úplně zapsána. V opačném případě se vám může stát, že se počítač Pi po vložení karty SD *nespustí*. Pokud k tomu dojde, celý postup přípravy karty opakujte.

Když je bitová kopie nahrána na kartu SD, vyjměte kartu z PC a vložte ji do patice karet SD v počítači Raspberry Pi. Tato patice se nachází pod deskou s tištěnými spoji. Kartu SD je potřeba vložit popiskem na opačné straně než deska a zasunout ji až na doraz, aby měla dobrý kontakt.

Připojení externího úložiště

Počítač Raspberry Pi sice jako hlavní úložné zařízení – označované jako *spouštěcí zařízení* – používá kartu SD, ale poměrně rychle se můžete dostat do situace, kdy na kartě není dost místa. Velké karty SD sice mohou mít kapacitu 32 GB, 64 GB nebo více, ale tyto karty bývají příliš drahé.

Naštěstí existují zařízení, která po připojení kabelem USB poskytují libovolnému počítači dodatečný pevný disk. Mezi tato zařízení hromadného úložiště označovaná zkratkou *UMS* (*USB Mass Storage*) patří fyzické pevné disky, jednotky SSD (solid-state drive), nebo dokonce miniaturní přenosné jednotky flash (viz obrázek 1.6).



Obrázek 1.6: Dvě zařízení USB Mass Storage: flash disk a externí pevný disk

K počítači Pi lze připojit většinu zařízení USB Mass Storage bez ohledu na to, zda již obsahují nějaká data. Aby mohl počítač Pi k těmto zařízením přistupovat, je nutné jejich jednotky *připojit* procesem, s nímž se seznámíte v kapitole 2, „Správa systému Linux“. Prozatím stačí, když jednotky připojíte k počítači Pi a ponecháte je připravené.

Připojení k síti

Většina pokynů týkajících se nastavení se sice týká stejně tak modelu A jako modelu B počítače Raspberry Pi, ale připojení k síti představuje zvláštní případ. Abychom mohli co nejvíce omezit počet komponent – a tedy i cenu, není model A vybaven integrovanou síťovou kartou. To naštěstí neznamená, že byste jej nemohli připojit k síti, jen k tomu budete potřebovat určité dodatečné vybavení.

Kabelové připojení k síti

Chcete-li v počítači Raspberry Pi využívat síťovou konektivitu, musíte jej propojit s přepínačem, směrovačem nebo rozbočovačem pomocí *ethernetového propojovacího kabelu RJ45*. Po-

kud nemáte směrovač či rozbočovač, můžete počítač Pi pomocí propojovacího kabelu přímo propojit se svým stolním počítačem či notebookem.

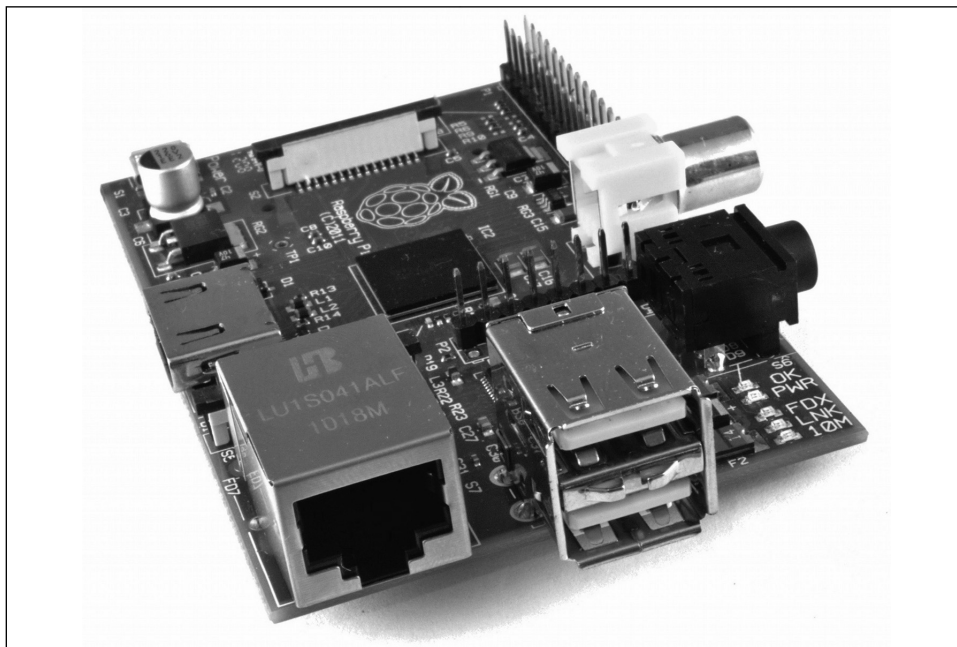
Připojení modelu A k síti

Chcete-li model A vybavit stejnými síťovými možnostmi, jako má jeho dražší varianta (model B), budete potřebovat adaptér sítě Ethernet připojený přes USB. Tento adaptér se připojí k volnému portu USB počítače Raspberry Pi nebo k připojenému rozbočovači a poskytne kabelové připojení sítě Ethernet pomocí konektoru RJ45, který je identický s konektorem modelu B.

V internetových obchodech si můžete velmi levně pořídit ethernetový adaptér pro USB typu 10/100. Tato čísla přitom označují jeho dva rychlostní režimy: 10 Mb/s a 100 Mb/s. Při nákupu ethernetového adaptéru nezapomeňte zkontrolovat, zda je mezi podporovanými operačními systémy uveden také systém Linux. Některé modely fungují pouze v systému Microsoft Windows a s počítačem Raspberry Pi nejsou kompatibilní.

Nenechte se zlákat k nákupu adaptéru gigabitové třídy, které bývají označeny hodnotami 10/100/1000. Standardní porty USB, kterými je počítač Raspberry Pi vybaven, nedokáží rychlosti gigabitového ethernetového připojení stačit a dražší adaptér vám proto neposkytne žádnou výhodu.

Tento způsob propojení dvou síťových klientů obvykle vyžaduje speciální kabel, který se označuje jako *překřížený kabel*. V překříženém kabelu jsou přehozeny dvojice vysílacích a přijímacích vodičů. Obě zařízení díky tomu nemohou vzájemně rušit svou komunikaci – obvykle se o to stará síťový přepínač nebo rozbočovač.



Obrázek 1.7: Ethernetový port modelu B počítače Raspberry Pi

Počítač Raspberry Pi je však ještě inteligentnější. Port RJ45 na boční straně počítače Pi (viz obrázek 1.7) poskytuje funkci označovanou jako *auto-MDI*, která mu umožňuje, aby automaticky měnil svou konfiguraci. Díky tomu lze počítač Pi připojit k síti pomocí libovolného kabelu RJ45 – překříženého či standardního – a počítač podle toho svou konfiguraci sám upraví.

Pokud počítač Pi připojíte přímo k PC či notebooku, ve výchozím nastavení se z něj nebude možné dostat na Internet. Za tímto účelem je potřeba nakonfigurovat PC tak, aby přemostil kabelový ethernetový port a jiné připojení (obvykle bezdrátové). Popis této operace překračuje rozsah této knihy, ale jestliže nemáte žádnou jinou možnost, jak svůj počítač Pi připojit k Internetu, můžete v nápovědě svého operačního systému vyhledat další pokyny pomocí klíčových slov „bridge network“ (přemostění sítě).

Je-li počítač Pi připojen kabelem, při spuštění svého operačního systému automaticky načte nastavení potřebná pro přístup k Internetu pomocí protokolu *DHCP* (*Dynamic Host Configuration Protocol*). Tato funkce přiřadí počítači Pi *IP* (*Internet Protocol*) adresu v síti a oznámí mu adresu brány, kterou potřebuje pro přístup k Internetu (obvykle se jedná o IP adresu směrovače či modemu).

V některých sítích neexistuje žádný server DHCP, který by počítači Pi poskytl IP adresu. V případě připojení k takové síti je nutné počítač Pi nakonfigurovat ručně. Další informace naleznete v kapitole 4, „Konfigurace sítě“.

Bezdrátové připojení k síti

Současné modely počítače Raspberry Pi nejsou standardně vybaveny žádným modulem bezdrátové sítě, ale stejně jako u přidání kabelového připojení sítě Ethernet u modelu A lze libovolný počítač Pi rozšířit o podporu sítě Wi-Fi pomocí bezdrátového adaptéru pro USB (viz obrázek 1.8).



Obrázek 1.8: Dva bezdrátové adaptéry pro USB, které jsou kompatibilní s počítačem Raspberry Pi

Pomocí takového zařízení se počítač Pi může připojit k různým typům bezdrátových sítí včetně těch, které využívají nejnovější vysokorychlostní standard 802.11n. Před nákupem bezdrátového adaptéru pro USB:

- Ověřte, zda je mezi podporovanými operačními systémy uveden i systém Linux. Některé bezdrátové adaptéry se dodávají pouze s ovladači pro systémy Windows a OS X, takže pomocí nich nelze počítač Raspberry Pi připojit. Seznam adaptérů Wi-Fi, u kterých byla ověřeno, že s počítačem Raspberry Pi fungují, naleznete na následujícím webu: http://linux.org/RPi_VerifiedPeripherals#USB_WiFi_Adapters.
- Zkontrolujte, zda je bezdrátový adaptér pro USB kompatibilní s použitým typem sítě Wi-Fi. Typ sítě je ve specifikaci označen číslem, po kterém následuje písmeno. Pokud například používáte síť typu 802.11a, nebude bezdrátový adaptér typu 802.11g fungovat.
- Zjistěte, které frekvence karta podporuje. Některé standardy bezdrátových sítí, jako je 802.11a, poskytují více než jednu frekvenci. Je-li bezdrátový adaptér pro USB určen pro nasazení v síti 2,4 GHz, nepřipojí se k síti 5 GHz.
- Ověřte, zda souhlasí typ šifrování, který je nastaven v bezdrátové síti. Většina moderních bezdrátových adaptérů pro USB je kompatibilní se všemi formami šifrování, ale pokud kupujete adaptér z druhé ruky nebo starší model adaptéru, můžete zjistit, že se k vaší síti nepřipojí. K běžným typům šifrování patří zastaralý standard WEP a modernější WPA a WPA2.

Konfigurace bezdrátového připojení probíhá v systému Linux, takže prozatím stačí adaptér připojit k počítači Pi (optimálně přes napájený rozbočovač USB). Postup konfigurace připojení naleznete v kapitole 4, „Konfigurace sítě“.

Připojení napájení

Počítač Raspberry Pi je napájen pomocí malého *konektoru mikro USB*, který se nachází v levém dolním rohu desky s tištěnými spoji. Jedná se o stejný konektor, kterým je vybavena většina smartphonů a některé tablety.

Mnoho nabíječek určených pro smartphony dovoluje nabíjet i počítač Raspberry Pi, ale neplatí to pro všechny. Počítač Pi má vyšší spotřebu než většina zařízení s konektorem mikro USB a ke svému fungování vyžaduje až 700 mA. Některé nabíječky umožňují dodávat proud nejvýše 500 mA, což při činnosti počítače Pi způsobuje občasné problémy (viz kapitola 3, „Řešení potíží“).

Počítač Pi lze sice připojit k portu USB stolního počítače nebo notebooku, ale nedoporučuje se to. Stejně jako u menších nabíječek platí, že porty USB v počítači nedokáží poskytovat takový výkon, který Pi ke svému správnému fungování potřebuje.

Kabel mikro USB s dodávkou napájení připojte teprve tehdy, když s počítačem Pi chcete začít pracovat. Vzhledem k tomu, že počítač nemá vypínač, spustí se ihned po připojení napájení. Chcete-li jej znovu vypnout, musíte odpojit napájecí kabel.

Správa systému Linux

Většina moderních distribucí systému Linux je uživatelsky přátelská a poskytuje *grafické uživatelské rozhraní (GUI)*, které umožňuje snadno provádět běžné úkoly. Značně se však liší od systémů Windows a OS X. Chcete-li tedy svůj počítač Raspberry Pi maximálně využít, musíte se seznámit se základy používání jeho operačního systému.

Linux: přehled

Jak jsme stručně vysvětlili v kapitole 1, „Seznámení s počítačem Raspberry Pi“, systém Linux se vyvíjí jako projekt open source. Jeho původním cílem bylo vytvořit *jádro* (kernel), se kterým by mohl kdokoli svobodně pracovat. Jádro je srdcem operačního systému a obsluhuje komunikaci mezi uživatelem a hardwarem.

Správně by se sice termínem Linux mělo označovat jen samotné jádro, ale často se tento pojem vztahuje na kolekci projektů open source od různých dodavatelů. Tyto kolekce společně tvoří různé *varianty* systému Linux, které jsou známy jako *distribuce*.

Původní verze Linuxu byla doplněna o kolekci nástrojů, které vytvořila skupina s názvem GNU. Výsledný systém označovaný jako *GNU/Linux* byl sice prostý, ale výkonný. Na rozdíl od jiných operačních systémů své éry poskytoval funkce typu více uživatelských účtů, kdy může několik uživatelů sdílet jediný počítač. Tuto vlastnost později získaly i konkurenční operační systémy typu closed source a v současnosti podporuje více uživatelských účtů v jednom systému jak systém Windows, tak OS X. Zmíněná funkce je v systému Linux samozřejmě dostupná i nadále a zajišťuje bezpečnost a ochranu operačního systému.

V systému Linux uživatel většinu času pracuje s *omezeným* uživatelským účtem. To ovšem neznamená, že by nemohl využít všech možností systému. Díky tomuto nastavení se však nemůže stát, že by uživatel náhodou provedl akci, která by poškodila software počítače Raspberry Pi. Omezený uživatelský účet také chrání systém před napadením *víry* a jiným *malwarem* (škodlivým softwarem), protože uzamyká přístup ke kritickým systémovým souborům a adresářům.

Než začnete se systémem pracovat, měli byste se seznámit s některými pojmy a koncepty, které se ve světě Linuxu používají. Jejich definice shrnuje tabulka 2.1. I když máte dostatek zkušeností s jinými operačními systémy, je vhodné, abyste se před prvním spuštěním počítače Pi do této tabulky podívali.

Terminál a grafické uživatelské rozhraní

Stejně jako v systémech OS X a Windows lze v Linuxu určitého cíle obvykle dosáhnout dvěma hlavními způsoby: přes grafické uživatelské rozhraní (GUI – graphical user interface) a pomocí příkazového řádku (který se v linuxové terminologii označuje jako *konzola* nebo *terminál*).

Různých linuxových distribucí se mohou značně lišit svým vzhledem, který závisí na použitém prostředí plochy. V této knize vycházíme z doporučené distribuce Debian, ale většina uváděných příkazů se zadává pomocí *terminálu* a obvykle bývají ve všech distribucích stejné.

Tam, kde se ovládání jiných distribucí liší, budeme poskytovat alternativní metody, jak dosáhnout stejných cílů.

Tabulka 2.1: Stručný slovníček pojmů systému Linux

Termín či princip	Definice
Adresář (directory)	Linuxový termín pro místo k ukládání souborů, které se v systému Windows označuje jako <i>složka</i> .
Balíček (package)	Kolekce souborů potřebných ke spuštění aplikace, kterou obvykle zpracovává správce balíčků.
Bash	Nejoblíbenější <i>příkazový interpret</i> (shell), který je součástí většiny distribucí systému Linux.
Distribuce (distribution)	Určitá verze systému Linux. K distribucím patří Fedora Remix, Arch a Debian.
EXT2/3/4	Rozšířený (EXTended) systém souborů. Tento formát systému souborů se v Linuxu používá nejčastěji.
GNOME	Jedno z nejrozšířenějších prostředí plochy v systému Linux.
GNU	Svobodný softwarový projekt, který poskytuje mnoho nástrojů zahrnutých do distribucí systému Linux.
GRUB	Zavaděč (GRand Unified Bootloader), který vznikl v nadaci GNU a slouží k načítání jádra systému Linux.
GUI	Grafické uživatelské rozhraní, kde uživatel ovládá počítač pomocí myši nebo dotykem.
KDE	Jiné mimořádně oblíbené prostředí plochy systému Linux.
Konzola (console)	Neustále dostupná verze terminálu, která se při spuštění počítače Pi zobrazí jako první.
Linux	Přesně řečeno se jedná o jádro, na kterém je založen operační systém GNU/Linux. Běžně se tímto názvem označuje celý operační systém typu open source.
Live CD	Linuxová distribuce poskytovaná na disku CD nebo DVD, která nevyžaduje instalaci.
Oddíl (partition)	Část pevného disku, která je připravena k vytvoření systému souborů, aby na ni bylo možné ukládat data.
Prostředí plochy (desktop environment)	Software, který zajišťuje atraktivní vzhled grafického uživatelského rozhraní. Oblíbená jsou prostředí plochy GNOME a KDE.
Příkazový interpret (shell)	Textový příkazový řádek, který se zobrazuje v okně terminálu.
Root	Hlavní uživatelský účet v systému Linux, který odpovídá účtu administrator v systému Windows.

Termín či princip	Definice
Správce balíčků (package manager)	Nástroj, který umožňuje sledovat nainstalovaný software a instalovat nový.
Spustitelný soubor (executable)	Soubor, který je možné spustit jako program. V systému Linux lze spustit pouze soubor, který je jako spustitelný označen.
sudo	Program, který umožňuje omezeným uživatelským účtům spouštět příkazy jako uživatel root .
Superuživatel (superuser)	Viz <i>root</i> .
Systém souborů (file system)	Způsob, kterým je formátován pevný disk nebo jiné úložné zařízení, aby na něj bylo možné ukládat soubory.
Terminál (terminal)	Textový příkazový řádek, ve kterém uživatel interaguje s programem <i>příkazového interpretu</i> .
X11	Systém X Window, což je balíček, který zpřístupňuje grafické uživatelské rozhraní (GUI).
Zavaděč (bootloader)	Software, který odpovídá za načítání jádra systému Linux. Nejčastěji se používá zavaděč GRUB.

Základy systému Linux

K dispozici jsou sice stovky různých linuxových distribucí, ale všechny sdílejí společnou sadu nástrojů. Tyto nástroje, které se ovládají pomocí terminálu, mají své obdobné protějšky v systémech Windows a OS X. Chcete-li začít pracovat se systémem Linux, měli byste znát následující příkazy:

- **ls** – zkratka anglického slova *listing* (výpis). Příkaz `ls` poskytuje přehled o obsahu aktuálního adresáře. Případně jej lze zavolat s argumentem adresáře, který má být vypsán. Příklad: Zadáte-li příkaz `ls /home`, dostanete seznam obsahu adresáře `/home` bez ohledu na aktuální adresář. Ekvivalent v systému Windows se nazývá `dir`.
- **cd** – iniciálová zkratka slov *change directory* (změnit adresář). Příkaz `cd` umožňuje procházet systémem souborů. Zadáte-li příkaz `cd` bez parametrů, přejdete zpět do svého domovského adresáře. Když příkaz doplníte o název jiného adresáře, příkaz zajistí přechod do příslušného adresáře. Všimněte si, že adresáře mohou být absolutní nebo relativní: příkazem `cd boot` přejdete do podadresáře s názvem `boot` v rámci aktuálního adresáře, ale příkaz `cd /boot` umožní z libovolného místa v systému souborů přímo přejít do adresáře `/boot`.
- **mv** – příkaz *move* (přesunout) má v systému Linux dvě funkce: dovoluje přesunout soubor mezi dvěma adresáři a kromě toho umožňuje soubory přejmenovat. Druhá funkce sice vypadá nepatřičně, ale z hlediska systému Linux je soubor „přesunut“ z jednoho názvu do jiného. Příkaz lze volat zápisem `mv starý_soubor nový_soubor`.
- **rm** – příkaz `rm` (zkratka slova *remove* – odebrat) slouží k odstraňování souborů. Příkaz odstraní libovolný soubor nebo více souborů, jejichž seznam je uveden za názvem příka-

zu. Ekvivalentem v systému Windows je příkaz `del` a v obou případech je potřeba věnovat pozornost tomu, aby nebyl omylem odstraněn nesprávný soubor.

- **rm** – samotný příkaz `rm` obvykle nedokáže odstraňovat adresáře. Proto je k dispozici příkaz `rmdir`, kterým lze odstraňovat adresáře poté, co z nich byly příkazem `rm` vymazány všechny soubory.
- **mkdir** – opakem příkazu `rmdir` je příkaz `mkdir` na vytváření nových adresářů. Zadáte-li v okně terminálu například příkaz `mkdir moje_složka`, vytvoříte v aktuálním pracovním adresáři nový adresář s názvem `moje_složka`. Stejně jako u příkazu `cd` platí, že adresáře uvedené jako parametr příkazu mohou být relativní či absolutní.

Představení distribuce Debian

Debian je jednou z nejdéle vyvíjených distribucí systému Linux a díky své nenáročnosti představuje výhodnou volbu pro uživatele počítače Raspberry Pi. Proto jej nadace Raspberry Pi Foundation zvolila jako software, který doporučuje novým uživatelům. Z distribuce Debian vycházejí i příklady v této knize.

Aby bylo možné co nejvíce omezit velikost souboru ke stahování, obsahuje bitová kopie distribuce Debian pro Raspberry Pi (Raspbian) pouze část softwaru, který je obsažen v běžné verzi pro počítače PC. Patří k nim nástroje na procházení webu, programování v jazyce Python a grafické rozhraní pro práci s počítačem Pi. Další software lze rychle nainstalovat pomocí správce balíčků `apt` této distribuce.

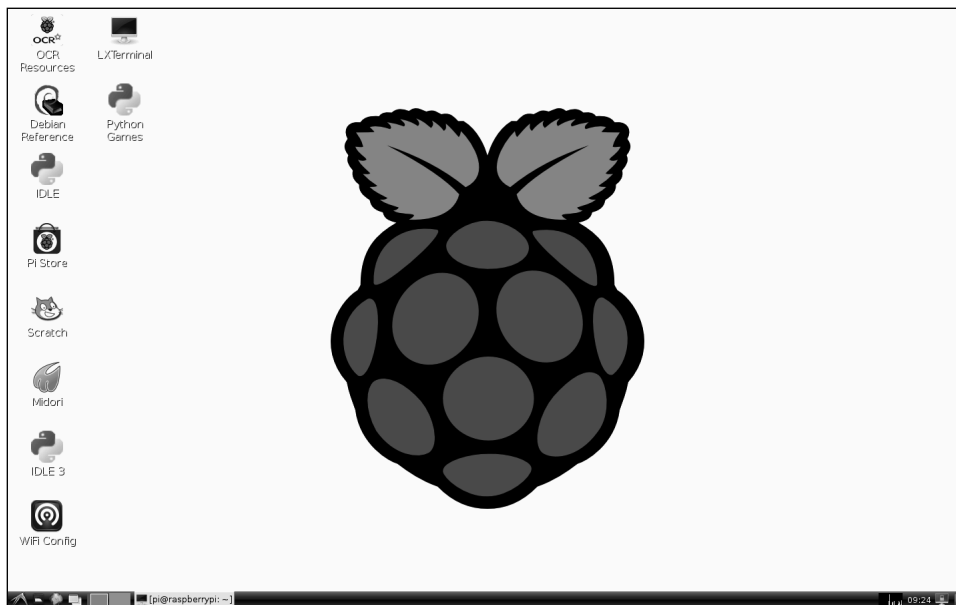
Sestavení distribuce Debian pro počítač Raspberry Pi zahrnuje prostředí plochy, které se nazývá *LXDE (Lightweight X11 Desktop Environment)*. Prostředí LXDE je navrženo tak, aby poskytovalo atraktivní uživatelské rozhraní na základě systému *X Window*. Poskytuje intuitivní rozhraní pro obsluhu myši, se kterým se snadno dokáže seznámit každý, kdo již pracoval se systémem Windows, OS X nebo jiným operačním systémem s grafickým uživatelským rozhraním.



Tip: U většiny distribucí pro Raspberry Pi se grafické uživatelské rozhraní nespouští automaticky. Chcete-li je rychle načíst a opustit textovou konzolu, přihlaste se, zadejte příkaz `startx` a stiskněte klávesu Enter. Pokud chcete ponechat grafické uživatelské rozhraní spuštěné na pozadí a přejít zpět do konzoly, stiskněte a podržte klávesy Ctrl a Alt a stiskněte klávesu F1. Poté všechny klávesy pusťte.

Jestliže používáte doporučenou distribuci Debian, zjistíte, že máte hned po instalaci systému k dispozici mnoho připravených programů. Tato sada softwaru pro počítač Pi sice ani zdaleka není úplná (uživatelé mohou nainstalovat tisíce různých *balíčků*), ale poměrně dobře představuje možnosti systému.

Software dodávaný s distribucí Debian je rozdělen do tematických kategorií. Chcete-li tyto kategorie zobrazit, můžete klepnout levým tlačítkem myši na ikonu nabídky, která se v prostředí LXDE nachází v levém dolním rohu obrazovky (viz obrázek 2.1).



Obrázek 2.1: Plocha prostředí LXDE po spuštění distribuce Debian pro Raspberry Pi

Popis softwarových balíčků seskupených podle kategorií naleznete v následujících seznamech.

Accessories (Příslušenství)

- **Debian Reference** – integrovaná referenční příručka, která obsahuje podrobný rozbor linuxové distribuce Debian a vysvětluje, jak mohou programátoři přispět k jejímu vývoji.
- **File Manager (Správce souborů)** – správce souborů PCManFM je grafický prohlížeč souborů, které jsou umístěny na kartě v počítači Pi nebo na připojeném úložném zařízení.
- **Image Viewer (Prohlížeč obrázků)** – prohlížeč GPicView umožňuje zobrazovat obrázky, například v digitálním fotoaparátu nebo na připojeném úložném zařízení.
- **Leafpad** – jednoduchý textový editor, který se hodí k psaní krátkých poznámek nebo tvorbě jednoduchých programů.
- **LXTerminal** – tento terminálový balíček prostředí LXDE umožňuje pracovat s příkazovým řádkem systému Linux v okně bez opouštění grafického uživatelského rozhraní.
- **Root Terminal (Terminál uživatele root)** – tento terminál se podobá oknu LXTerminal, ale automaticky vás přihlásí k účtu superuživatele *root*, abyste mohli provádět úkoly údržby systému, které nejsou v běžném uživatelském účtu dostupné.
- **Xarchiver** – tento nástroj umožňuje vytvářet nebo rozbalovat komprimované soubory, jako jsou archivy ve formátu Zip.

Education (Vzdělávání)

- **Scratch** – grafický programovací jazyk určený pro malé děti. Další informace o jazyce Scratch a jeho možnostech naleznete v kapitole 10, „Úvod do jazyka Scratch“.

- **Squeak** – platforma, na které funguje jazyk Scratch. Tuto položku nabídky obvykle není potřeba používat a místo ní lze klepnout na výše zmíněnou položku Scratch.

Internet

- **Midori** – rychlý, ale jednoduchý webový prohlížeč Midori odpovídá prohlížeči Internet Explorer v systému Windows či prohlížeči Safari v systému OS X.
- **Midori Private Browsing (Midori – soukromé prohlížení)** – klepnutím na tuto položku spustíte prohlížeč Midori v soukromém režimu, což znamená, že navštívené stránky nebudou uloženy do historie prohlížeče.
- **NetSurf Web Browser (NetSurf Webový prohlížeč)** – alternativa k prohlížeči Midori s názvem NetSurf poskytuje u některých typů webových stránek lepší výkon. Když vyzkoušíte oba prohlížeče, můžete určit ten, který vám lépe vyhovuje.

Programming (Programování)

- **IDLE** – *integrované vývojové prostředí (IDE – integrated development environment)*, které je speciálně napsáno pro jazyk Python. Další informace o psaní vlastních programů jazyka Python v prostředí IDLE naleznete v kapitole 11 „Úvod do jazyka Python“.
- **IDLE 3** – po klepnutí na tuto položku spustíte prostředí IDLE konfigurované k použití novější verze programovacího jazyka Python 3 namísto výchozího jazyka Python 2.7. Oba jazyky ve většině aspektů vzájemně kompatibilní, ale některé programy mohou vyžadovat funkce jazyka Python 3.
- **Scratch** – tento zástupce spustí výukový jazyk Scratch a je totožný s položkou Scratch, která je dostupná v kategorii Education. Program je možné spustit kteroukoli z nich.
- **Squeak** – stejně jako u jazyka Scratch se jedná o duplicitní položku k zástupci z kategorie Education. Tato položka obvykle není potřebná a místo ní je vhodné použít zástupce Scratch.

System Tools (Systémové nástroje)

- **Task Manager (Správce úloh)** – tímto nástrojem lze kontrolovat velikost dostupné paměti počítače Pi a aktuální zatížení procesoru a ukončovat programy, které selhaly nebo neodpovídají.

Preferences (Volby)

- **Customise Look and Feel (Upravit vzhled a chování)** – sada nástrojů na úpravy vzhledu grafického uživatelského rozhraní včetně stylu a barvy oken.
- **Desktop Session Settings (Nastavení relace pracovního prostředí)** – tento nástroj umožňuje měnit chování systému po přihlášení uživatele včetně automaticky spuštěných programů a použitého správce oken – softwaru, který vykresluje okraje a záhlaví oken.
- **Keyboard and Mouse (Klávesnice a myš)** – nástroj k přizpůsobení vstupních zařízení. Pokud klávesnice po stisknutí určitých kláves poskytuje chybné znaky nebo je myš příliš citlivá, můžete příslušná nastavení změnit pomocí tohoto programu.

- **Monitor Settings (Nastavení monitoru)** – na tomto místě můžete změnit některá nastavení monitoru nebo televizoru připojeného k počítači Pi, ačkoli složitější změny mohou vyžadovat úpravy konfiguračních souborů. Další příslušné informace naleznete v kapitole 6, „Konfigurace počítače Raspberry Pi“.
- **Openbox Configuration Manager** – grafické uživatelské prostředí LXDE je založeno na prostředí plochy označovaném jako *Openbox*, které lze nastavit na tomto místě. Pomocí tohoto nástroje je možné změnit vzhled grafického uživatelského rozhraní pomocí nových motivů nebo upravit způsob fungování některých prvků rozhraní.
- **Preferred Applications (Preferované aplikace)** – tímto nástrojem můžete změnit aplikace, které se budou spouštět pro určité typy souborů. Jestliže se rozhodnete používat alternativní webový prohlížeč, můžete výchozí nastavení systému změnit zde.

Získání nápovědy

Systém Linux je navržen tak, aby co nejvíce vycházel vstříc novým uživatelům – dokonce i v okně terminálu s příkazovým řádkem. V této kapitole se sice naučíte nejobvyklejší varianty použití každého příkazu, ale nebudeme se zabývat všemi jejich možnostmi – v takovém případě by kniha musela být mnohem tlustší.

Pokud se dostanete do situace, kdy si nebudete vědět rady nebo se budete chtít o některém nástroji popsaném na následujících stranách dozvědět více, měli byste znát následující příkaz: **man**.

Ke všem linuxovým aplikacím patří soubor nápovědy, který se označuje jako *manuálová stránka* (man page). Popisuje vlastnosti softwaru a zároveň uvádí podrobnosti o tom, jaké má parametry a jak se tyto parametry používají.

Chcete-li otevřít manuálovou stránku pro určitý nástroj, stačí zadat příkaz **man** následovaný názvem příkazu. Pokud chcete zobrazit manuálovou stránku programu **ls**, což je nástroj na výpis obsahu adresářů, zadejte **man ls**.

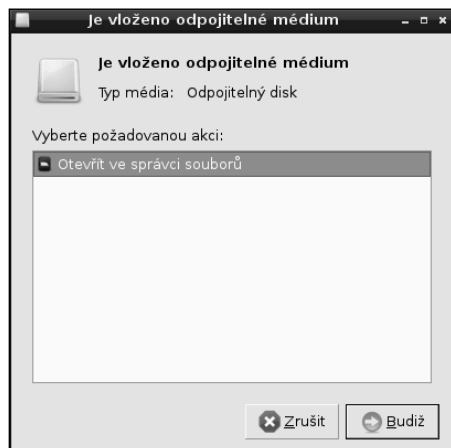
Použití externích úložných zařízení

Karta SD počítače Pi, na které jsou uloženy různé soubory a adresáře operačního systému, není příliš velká. Největší dostupná karta SD v době psaní této knihy má kapacitu 64 GB, což je značně nižší hodnota oproti 3 000 GB (3 TB), které nabízí největší pevné disky plného formátu určené pro stolní počítače.

Pokud v počítači Pi přehráváte soubory videa (viz kapitola 7, „Počítač Pi jako domácí kino“), pravděpodobně budete potřebovat více místa, než které dokáže poskytnout karta SD. V kapitole 1, „Seznámení s počítačem Raspberry Pi“, jsme již uvedli, že k počítači Pi lze připojit zařízení UMS (USB Mass Storage), která mohou zvýšit dostupnou úložnou kapacitu.

Aby však bylo možné tato externí zařízení zpřístupnit, musí o nich operační systém vědět. Příslušný proces se v systému Linux označuje jako *připojení* (mounting). Pracujete-li s verzí systému Linux, kde je načteno prostředí plochy (jako je prostředí LXDE doporučené distribuce Debian, které se spouští z konzoly příkazem **startx**), probíhá tento proces automaticky. Stačí

zařízení připojit k volnému portu USB v počítači Pi nebo rozbočovači USB a zařízení spolu se svým obsahem bude okamžitě dostupné (viz obrázek 2.2).



Obrázek 2.2: Prostředí LXDE automaticky připojuje zařízení UMS

Postup v okně konzoly je jen o málo složitější. Chcete-li v systému Linux zpřístupnit zařízení, když není spuštěno prostředí plochy, postupujte takto:

1. Připojte úložné zařízení pro USB k počítači Pi buď přímo, nebo pomocí připojeného rozbočovače USB.
2. Příkazem `sudo fdisk -l` zobrazte seznam jednotek připojených k počítači Pi a vyhledejte úložné zařízení pro USB podle jeho velikosti. Poznamenejte si název zařízení: `/dev/sdXN`, kde *X* je písmeno jednotky a *N* znamená číslo oddílu. Pokud je k počítači Pi připojeno jen jediné zařízení, bude označeno jako `/dev/sda1`.
3. Než bude úložné zařízení pro USB přístupné, musí pro něj systém Linux nastavit přípojný bod. Tento bod vytvoříte příkazem `sudo mkdir /media/externí_jednotka`.
4. Zařízení je aktuálně přístupné pouze pro uživatele root. Chcete-li je zpřístupnit všem uživatelům, zadejte následující řetězec na jediný řádek:

```
sudo chgrp -R users /media/externí_jednotka && ↵
sudo chmod -R g+w /media/externí_jednotka
```
5. Připojte úložné zařízení pro USB pomocí příkazu `sudo mount /dev/sdXN /media/externí_jednotka -o=rw`. Tím získáte přístup k zařízení a jeho obsahu.

Vytvoření nového uživatelského účtu

Na rozdíl od systémů Windows a OS X, které jsou v zásadě určeny k tomu, aby s nimi pracoval jednotlivec, systém Linux je od základů společenský operační systém vytvořený tak, aby podporoval více uživatelů. Systém Debian má ve výchozím nastavení nakonfigurovány dva

uživatelské účty: *pi*, což je normální uživatelský účet, a *root*, tj. účet superuživatele s dodatečnými oprávněními.



Tip: Nepodléhejte pokušení pravidelně se přihlašovat k účtu **root**. Při práci s uživatelským účtem bez zvýšených oprávnění jste chráněni před náhodným poškozením operačního systému a před nebezpečím virů a jiných škodlivých programů, které mohou pocházet z Internetu.

Samozřejmě je možné, abyste pracovali s účtem *pi*, ale vhodnější je vytvořit vlastní vyhrazený uživatelský účet. Můžete vytvořit také další účty pro všechny přátele či členy rodiny, kteří by mohli chtít počítač Pi používat.

Nový účet v počítači Pi vytvoříte docela snadno a postup je přibližně stejný ve všech distribucích. Jediný rozdíl spočívá v uživatelském jménu a heslu, které slouží k prvnímu přihlášení k počítači Pi. Postupujte takto:



Tip: Symbol ↵ označuje, že ačkoli je text v knize uveden na více řádcích, měli byste příkaz zadávat na jediný řádek a klávesu Enter stisknout teprve na konci.

1. Přihlaste se k počítači Pi pomocí existujícího uživatelského účtu (používáte-li doporučenou distribuci Debian, jedná se o uživatelské jméno *pi* a heslo *raspberry*).

2. Zadejte následující příkaz na jediný řádek:

```
sudo useradd -m -G adm,dialout,cdrom,audio,plugdev,users, ↵
lpadmin,sambashare,vchiq,powerdev uživatelské_jméno
```

Tím vytvoříte nový prázdný uživatelský účet. Nezapomeňte, že příkaz je potřeba zadat na jediný řádek a za čárky se nesmí vkládat žádné mezery. Jestliže ve své distribuci nemáte příkaz `sudo`, přejděte nejdříve na účet `root` příkazem `su`.

3. Chcete-li nastavit heslo nového účtu, zadejte příkaz `sudo passwd uživatelské_jméno` a na výzvu zadejte nové heslo.

Uživatelé a skupiny

Každý uživatel systému Linux má tři hlavní atributy: své *ID uživatele* (UID – User ID), *ID skupiny* (GID – Group ID) a seznam členství v dalších skupinách. Uživatel může být členem libovolného počtu skupin, ačkoli pouze jedna z nich může sloužit jako uživatelská primární skupina. Obvykle se jedná o skupinu se stejným názvem jako uživatelské jméno.

Členství ve skupinách je důležité. Uživatelům lze sice udělit přímý přístup k souborům a zařízením v systému, ale častěji uživatelé získávají přístup k těmto prostředkům na základě členství ve skupinách. Skupina **audio** například uděluje všem svým členům právo přístupu k hardwaru počítače Pi na přehrávání zvuku. Pokud uživatel není členem této skupiny, nemá možnost poslouchat hudbu.

Chcete-li zobrazit členství uživatele ve skupinách, zadejte v okně terminálu `groups uživatelské_jméno`. Pokud tento příkaz spustíte pro výchozího uživatele *pi*, zobrazí se seznam skupin, jejichž členem by měl být každý nový uživatel počítače Pi. Tímto způsobem jsme zjistili skupiny, které jsme uvedli v příkazu z kroku 2 předchozího postupu.

Vysvětleme, co se právě stalo: Příkaz `sudo` sdělí operačnímu systému, že právě zadávaný příkaz by měl spustit tak, jako byste byli přihlášení k účtu *root*. Příkaz `useradd` znamená, že chcete vytvořit nový uživatelský účet. Sekce `-m`, která se označuje jako *příznak* (flag) nebo *parametr* (option), požádá program `useradd`, aby vytvořil domovský adresář, kam může nový uživatel ukládat své soubory. Dlouhý seznam za příznakem `-G` uvádí skupiny, jejichž má být uživatel členem.

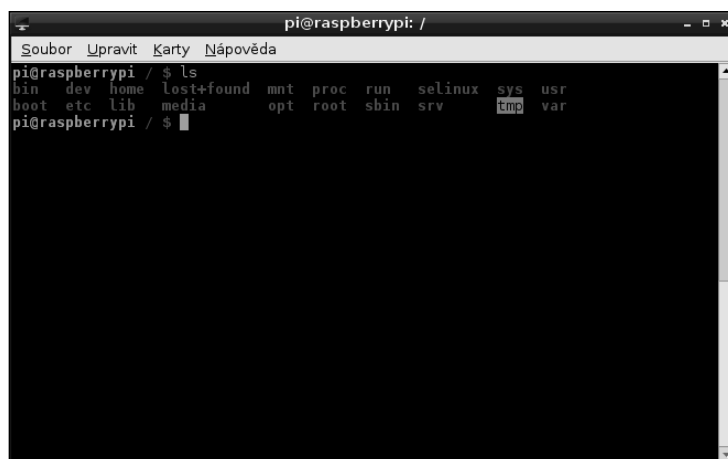
Struktura systémů souborů

Obsah karty SD se označuje jako *systém souborů* a dělí se na různé části, z nichž každá má určitou funkci. Chcete-li počítač Raspberry Pi pouze používat, nepotřebujete vědět, k čemu každá část systému souborů slouží. Tyto znalosti se vám však mohou hodit v případě potíží.

Logická struktura

Systém Linux pracuje s jednotkami, soubory, složkami a zařízeními poněkud odlišně než jiné operační systémy. Místo více jednotek označených písmeny se vše zobrazuje jako větve takzvaného *kořenového systému souborů*.

Když se přihlásíte k počítači Pi a zadáte příkaz `ls /`, dostanete seznam různých adresářů (viz obrázek 2.3). Některé z nich odpovídají oblastem pro ukládání souborů na kartě SD, zatímco jiné jsou *virtuální adresáře*, které umožňují přístup k různým částem operačního systému nebo hardwaru.



Obrázek 2.3: Výpis adresářů kořenového systému souborů počítače Pi

Ve výchozí distribuci Debian jsou zobrazeny následující adresáře:

- **boot** – obsahuje jádro systému Linux a jiné balíčky, které jsou nezbytné ke spuštění počítače Pi.

- **bin** – v tomto adresáři jsou uloženy binární soubory operačního systému, například soubory potřebné k činnosti grafického uživatelského rozhraní.
- **dev** – jedná se o virtuální adresář, který na kartě SD ve skutečnosti neexistuje. Z tohoto adresáře lze získat přístup ke všem zařízením připojeným k systému, včetně úložných zařízení, zvukové karty a portu HDMI.
- **etc** – uchovává různé konfigurační soubory, mimo jiné seznam uživatelů a jejich zašifrovaná hesla.
- **home** – v rámci tohoto adresáře dostane každý uživatel svůj podadresář, kam může ukládat své osobní soubory.
- **lib** – funguje jako úložiště *knihoven*, což jsou sdílené části kódu, které slouží mnoha různým aplikacím.
- **lost+found** – v případě havárie systému se do tohoto speciálního adresáře ukládají *fragменты souborů*.
- **media** – je to speciální adresář pro vyměnitelná úložná zařízení, jako jsou USB flash disky nebo externí jednotky CD.
- **mnt** – tato složka se používá k ručnímu *připojení* úložných zařízení, například externích pevných disků.
- **opt** – tento adresář obsahuje volitelný software, který není součástí samotného operačního systému. Pokud do počítače Pi nainstalujete nové programy, obvykle budou umístěny sem.
- **proc** – jedná se o další virtuální adresář, který obsahuje informace o běžících programech, označovaných v systému Linux jako *procesy*.
- **selinux** – soubory související se sadou bezpečnostních nástrojů *Security Enhanced Linux*, kterou původně vyvinula americká agentura NSA (National Security Agency).
- **sbin** – tento adresář uchovává speciální binární soubory, které primárně slouží uživateli root (superuživateli) k údržbě systému.
- **sys** – v tomto adresáři se nacházejí speciální soubory operačního systému.
- **tmp** – sem se automaticky zapisují dočasné soubory.
- **usr** – tento adresář poskytuje úložiště uživatelským programům.
- **var** – do tohoto virtuálního adresáře programy ukládají měnící se hodnoty neboli *proměnné*.

Fyzická struktura

Předchozí seznam sice informuje o tom, jak vypadá systém souborů v operačním systému Linux, ale neodpovídá tomu, jak jsou soubory umístěny na vlastní kartě SD. V případě výchozí distribuce Debian je karta SD rozdělena na dvě hlavní části. Označují se jako *oddíly* (partition), protože dělí zařízení na různé oblasti, podobně jako kapitoly pomáhají uspořádat obsah knihy.

Prvním oddílem na disku je malý oddíl (75 MB) naformátovaný systémem souborů VFAT, což je stejný formát oddílu, jaký se používá pro vyměnitelné jednotky v systému Microsoft Windows. Tento oddíl je v systému Linux *připojen* neboli zpřístupněn v adresáři `/boot` a za-

hrnuje všechny soubory, které jsou potřebné ke konfiguraci počítače Raspberry Pi a načtení samotného operačního systému.

Druhý oddíl je mnohem větší a naformátován je pomocí systému souborů EXT4, což je nativní linuxový systém souborů, který zajišťuje vysokou přístupovou rychlost a bezpečnost dat. Tento oddíl obsahuje převážnou část celé distribuce. Jsou zde uloženy všechny programy, obsah plochy, uživatelské soubory a uživatelsky nainstalovaný software. Uvedený oddíl zabírá většinu karty SD.

Instalování a odinstalování softwaru

Výchozí software nainstalovaný v distribuci Debian by měl zpočátku postačovat, ale většina uživatelů chce svůj počítač Pi přizpůsobit vlastním potřebám.

Nový software se do počítače Pi instaluje snadno. Distribuce Debian obsahuje nástroj s názvem **apt**, což je výkonný *správce balíčků*. Jako balíčky se v systému Linux označují části softwaru nebo kolekce různých částí softwaru, které jsou navrženy tak, aby vzájemně spolupracovaly.

Nástroj **apt** se sice ovládá z příkazového řádku, ale používá se velmi snadno a naučíte se jej rychle. Existují sice grafická uživatelská rozhraní pro tento nástroj, jako je oblíbený Synaptic Package Manager, ale tato rozhraní lze v počítači Pi kvůli omezené paměti často spouštět jen obtížně. Doporučujeme proto, abyste software instalovali pomocí terminálu.

Jiné distribuce

V distribuci Debian a odvozených distribucích obvykle jako správce balíčků slouží nástroj **apt**. Existují však i další obdobné nástroje a autoři jiných distribucí se mohou rozhodnout pro jiný z nich. Fedora Remix například používá nástroj **pacman**.

S nástrojem **pacman** se nepracuje obtížněji než se správcem **apt**, ale má odlišnou syntaxi (způsob zápisu instrukcí při instalaci nového softwaru nebo odebrání stávajícího). Chcete-li získat pokyny k použití nástroje **pacman** místo správce **apt**, zadejte v terminálu distribuce Fedora Remix příkaz **man pacman**.

V jiných distribucích můžete najít správce balíčků **yum**. Testujete-li distribuci, která obsahuje nástroj **yum**, můžete zobrazit pokyny zadáním příkazu **man yum** v okně terminálu.

Úkolem správce balíčků je sledovat veškerý software nainstalovaný v systému. Správce se neomezuje na instalaci nového softwaru, ale kontroluje také, které programy jsou aktuálně nainstalovány, umožňuje odebrat starý software a instaluje zpřístupněné aktualizace.

Správa balíčků představuje jednu z oblastí, kde se systém Linux značně liší od operačních systémů jako Windows či OS X. Nový software k instalaci sice můžeme stáhnout ručně, ale mnohem častější je instalace pomocí integrovaných nástrojů správy balíčků.



Tip: Než se pokusíte instalovat nový software nebo upgradovat stávající, měli byste aktualizovat *mezipaměť* nástroje **apt**. Za tímto účelem stačí zadat příkaz **sudo apt-get update**.

Hledání softwaru

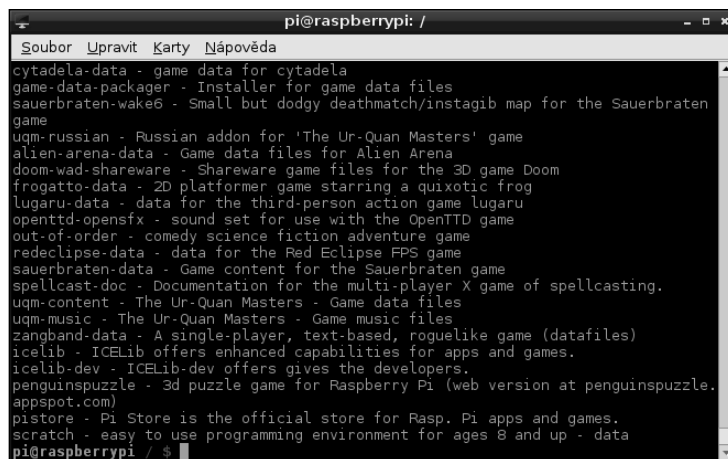
V prvním kroku instalace nové softwarové položky je potřeba zjistit, jak se nazývá. Nejjednodušší je přitom prohledat *mezipaměť* (cache) dostupných softwarových balíčků. Mezipaměť obsahuje seznam všech programů, které lze nainstalovat pomocí správce `apt`. Tyto programy jsou uloženy na internetových serverech, které se označují jako *repozitáře* (repository).

Software `apt` zahrnuje nástroj na správu mezipaměti, který se nazývá `apt-cache`. Pomocí uvedeného programu lze vyhledávat určité slovo nebo frázi ve všech dostupných softwarových balíčcích.

Chcete-li si například zahrát nějakou hru, můžete zadat následující příkaz:

```
apt-cache search game
```

Tento příkaz požádá nástroj `apt-cache`, aby ve všech dostupných programech vyhledal všechny položky, které mají ve svém názvu nebo popisu slovo „game“. V případě běžných vyhledávacích termínů můžete dostat poměrně dlouhý seznam (viz obrázek 2.4), takže se pokuste být co nejkonkrétnější.



```

pi@raspberrypi: /
Soubor  Upravit  Karty  Nápověda
cytadela-data - game data for cytadela
game-data-packager - Installer for game data files
sauerbraten-wake6 - Small but dodgy deathmatch/instagib map for the Sauerbraten
game
uqm-russian - Russian addon for 'The Ur-Quan Masters' game
alien-arena-data - Game data files for Alien Arena
doom-wad-shareware - Shareware game files for the 3D game Doom
frogatto-data - 2D platformer game starring a quixotic frog
lugaru-data - data for the third-person action game lugaru
openttd-opensfx - sound set for use with the OpenTTD game
out-of-order - comedy science fiction adventure game
redeclipse-data - data for the Red Eclipse FPS game
sauerbraten-data - Game content for the Sauerbraten game
spellcast-doc - Documentation for the multi-player X game of spellcasting.
uqm-content - The Ur-Quan Masters - Game data files
uqm-music - The Ur-Quan Masters - Game music files
zangband-data - A single-player, text-based, roguelike game (datafiles)
icelib - ICElib offers enhanced capabilities for apps and games.
icelib-dev - ICElib-dev offers gives the developers.
penguinspuzzle - 3d puzzle game for Raspberry Pi (web version at penguinspuzzle.
appspot.com)
pistore - Pi Store is the official store for Rasp. Pi apps and games.
scratch - easy to use programming environment for ages 8 and up - data
pi@raspberrypi / $

```

Obrázek 2.4: Několik posledních výsledků hledání slova „game“ nástrojem `apt-cache`



Tip: Jestliže hledaný termín vypíše příliš mnoho různých balíčků, takže je nelze zobrazit na jediné obrazovce, můžete systému Linux sdělit, že chcete výpis po každé obrazovce pozastavit. Přitom je potřeba *zřetěžit* (pipe) výstup nástroje `apt-cache` s nástrojem, který se nazývá `less`. Pouze upravte příkaz do podoby `apt-cache search game | less` a poté můžete seznam procházet kurzorovými klávesami. Výpis zavřete stisknutím klávesy `Q`.

Instalace softwaru

Když zjistíte název požadovaného balíčku, přejděte k jeho instalaci, kterou zajišťuje příkaz `apt-get`. K instalaci softwaru je oprávněn pouze uživatel *root*, protože tato operace ovlivnu-

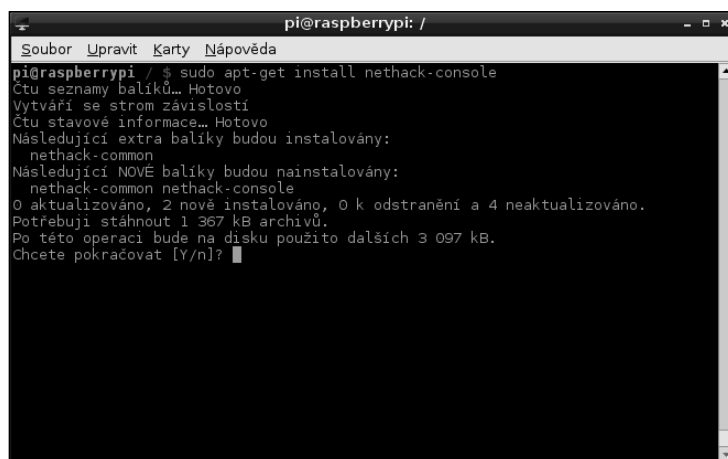
je všechny uživatele počítače Pi. Příkazy je proto nutné uvést příkazem `sudo`, který operační systém informuje o tom, že má tyto příkazy spustit jako uživatel *root*.

Chcete-li například nainstalovat balíček `nethack-console` (konzolová náhodně generovaná hra RPG), stačí zadat příkaz `install` nástroje `apt-get` takto:

```
sudo apt-get install nethack-console
```

Některé balíčky ke svému fungování vyžadují jiné balíčky. Programovací jazyk může záviset na kompilátoru, herní modul na grafických souborech nebo zvukový přehrávač na kodecích, které umožňují přehrávat různé formáty. V terminologii Linux to označujeme jako *závislosti* (dependency).

Závislosti představují jeden z hlavních důvodů, proč se místo ruční instalace softwaru používá správce balíčků, jako je `apt`. Jestliže balíček závisí na jiných balíčcích, nástroj `apt` je automaticky vyhledá (viz obrázek 2.5) a připraví jejich instalaci. V takových případech se zobrazí dotaz, zda chcete pokračovat. Pokud ano, zadejte písmeno `Y` a poté stiskněte klávesu `Enter`.



```
pi@raspberrypi: /  
Soubor Upravit Karty Nápověda  
pi@raspberrypi / $ sudo apt-get install nethack-console  
Ctu seznamy balíčků... Hotovo  
Vytváří se strom závislosti  
Ctu stavové informace... Hotovo  
Následující extra balíky budou instalovány:  
  nethack-common  
Následující NOVE balíky budou nainstalovány:  
  nethack-common nethack-console  
0 aktualizováno, 2 nově instalováno, 0 k odstranění a 4 neaktualizováno.  
Potřebuji stáhnout 1 367 kB archivů.  
Po této operaci bude na disku použito dalších 3 097 kB.  
Chcete pokračovat [Y/n]?
```

Obrázek 2.5: Nástroj `apt` zobrazuje závislosti balíčku

Odinstalování softwaru

Jestliže zjistíte, že některý program nadále nepotřebujete, poskytuje nástroj `apt-get` také příkaz `remove`. Tento příkaz čistým způsobem odinstaluje balíček spolu s případnými závislostmi, které již nejsou potřebné. Pokud v počítači Pi používáte kartu SD s menší kapacitou, je možnost vyzkoušet software a rychle jej odebrat mimořádně užitečná.

Chcete-li odebrat program `thrust`, stačí otevřít terminál a zadat následující příkaz:

```
sudo apt-get remove thrust
```

Účinnější alternativou příkazu `remove` je příkaz `purge`. Podobně jako příkaz `remove` také příkaz `purge` odstraní software, který již nechcete používat. Zatímco příkaz `remove` ponechá konfi-

gurační soubory softwaru beze změny, příkaz `purge` odebere celou instalaci. Příkaz `purge` je ideální v situacích, kdy se vám určitý balíček nepodaří nastavit a přestane fungovat. Chcete-li například úplně odstranit program `thrust`, zadejte pouze:

```
sudo apt-get purge thrust
```

Upgrade softwaru

Kromě instalování a odinstalování balíčků dokáže nástroj `apt` také udržovat balíčky aktuální. Při upgradu balíčku pomocí nástroje `apt` máte jistotu, že dostanete nejnovější aktualizace, opravy chyb a bezpečnostní opravy.

Než se pokusíte balíček upgradovat, aktualizujte mezipaměť nástroj `apt`, aby obsahovala co nejnovější informace:

```
sudo apt-get update
```

Při upgradech softwaru máte dvě možnosti: můžete upgradovat všechny součásti systému najednou nebo upgradovat jednotlivé programy. Chcete-li udržet aktuální celou distribuci, zadejte tento příkaz:

```
sudo apt-get upgrade
```

Jestliže chcete upgradovat jednotlivý balíček, jednoduše požádejte nástroj `apt`, aby jej nainstaloval znovu. Chcete-li například nainstalovat upgrade programu `thrust`, zadejte tento příkaz:

```
sudo apt-get install thrust
```

Pokud je balíček již nainstalován, nástroj `apt` tento požadavek zpracuje jako zástupný upgrade. V případě, že již pracujete s nejnovější dostupnou verzí, nástroj `apt` pouze sdělí, že software nelze aktualizovat, a ukončí se.



Tip: Chcete-li získat další informace o správě balíčků nástrojem `apt`, konkrétně o tom, jak lze určité balíčky „pozdržet“ a vyloučit z upgradů, zadejte v okně terminálu příkaz `man apt`.

Řešení potíží

Věci občas nefungují tak hladce, jak by měly. Čím je zařízení složitější, tím komplexnější problémy mohou nastat – a počítač Pi patří mezi hodně složitá zařízení.

Mnoho běžných problémů lze naštěstí diagnostikovat a opravit poměrně snadno. V této kapitole se zaměříme na některé nejčastější důvody, proč se počítač Pi nechová podle očekávání, a ukážeme, jak potíže vyřešit.

Diagnostika klávesnice a myši

Nejspíš nejobvyklejší problém, s nímž se uživatelé počítače Raspberry Pi setkávají, spočívá v tom, že klávesnice opakuje některé znaky. Pokud se například příkaz `startx` na obrazovce vypíše jako `stttttttttartxxxxxxxxxxxxx`, pochopitelně po stisknutí klávesy Enter nebude fungovat.

Jestliže klávesnice USB po připojení k počítači Raspberry Pi nepracuje správně, je to typicky způsobeno jedním ze dvou důvodů: má příliš vysoké nároky na napájení nebo dochází ke konfliktu její interní čipové sady s modulem USB počítače Pi.

V dokumentaci ke své klávesnici nebo na štítku na její dolní straně se pokuste najít její požadavky na napájení v *miliampérech (mA)*. Takový proud se klávesnice při své činnosti pokouší získávat z portu USB.

K portům USB počítače Pi je připojena komponenta, která se označuje jako *vratná pojistka* a chrání počítač Pi v případě, že se zařízení pokouší odebírat příliš velký proud. Pokud se tato vratná pojistka asi při 150 mA aktivuje, způsobí vypnutí portu USB. Jestliže klávesnice odebírá přibližně takový proud, může fungovat nespolehlivě nebo vůbec. Může se jednat o omezení pro klávesnice s integrovaným osvětlením LED, které ke svému provozu vyžadují mnohem více energie než standardní klávesnice.

Zjistíte-li, že vaše klávesnice USB čerpá příliš vysoký proud, nepřipojujte ji přímo k počítači Pi, ale pokuste se ji připojit k napájenému rozbočovači USB. Klávesnice tak dokáže odebírat proud z napájecí jednotky rozbočovače, a nikoli z vlastního počítače Pi. Případně můžete klávesnici nahradit modelem, který má nižší požadavky na napájení. Problém s opakováním písmen je také možné vystopovat k nedostatečné dodávce napájení samotného počítače Pi, čímž se budeme zabývat v části „Diagnostika napájení“.

Potíže s kompatibilitou se bohužel diagnostikují obtížněji. Naprostá většina klávesnic sice s počítačem Pi spolupracuje docela dobře, ale několik málo z nich vykazuje podivné příznaky. Patří k nim například občasné vynechávání, syndrom opakovaného písmene, nebo dokonce havárie, které znemožňují činnost počítače Pi. Tyto potíže se někdy projevují teprve poté, kdy jsou k počítači Pi připojena další zařízení USB. Pokud klávesnice přestane správně fungovat poté, co připojíte jiné zařízení USB (zejména bezdrátový adaptér pro USB), může se jednat o potíže s kompatibilitou.

Je-li to možné, pokuste se klávesnici zaměnit za jiný model. Jestliže nová klávesnice funguje, předchozí model pravděpodobně nebyl kompatibilní s počítačem Pi. Seznam klávesnic, které jsou známy svou nekompatibilitou, naleznete na stránce eLinux wiki: http://elinux.org/RPi_VerifiedPeripherals#Problem_USB_Keyboards.

Stejná rada ohledně preventivní kontroly kompatibility se vztahuje i na problémy s myši: většina myši a trackballů pro USB funguje správně, ale u některých z nich se projevuje nekompatibilita s modulem sběrnice USB počítače Pi. Obvykle se přitom objevují příznaky typu poskakujícího nebo zablokovaného ukazatele myši, ale jindy mohou vést k tomu, že počítač Pi nelze spustit nebo v nepředvídatelných intervalech selhává. Pokud se chystáte k nákupu nové myši, naleznete na stránce eLinux wiki aktuální seznam modelů, u kterých je ověřeno, že s počítačem Pi spolupracují: http://elinux.org/RPi_VerifiedPeripherals#Working_USB_Mouse_Devices.

Diagnostika napájení

Mnoho problémů s počítačem Raspberry Pi se odvíjí od nevyhovujícího zdroje napájení. Model A vyžaduje napájení s napětím 5 V a schopností dodávat proud 500 mA, zatímco dodatečné komponenty modelu B zvyšují jeho požadavky na napájení až k 700 mA. Konstrukce některých napájecích adaptérů USB takový proud nedokáže poskytovat, i když to jejich technická specifikace tvrdí.



Tip: Formální standard USB požaduje, aby zařízení neodebírala více než 500 mA, přičemž i tento proud je pro zařízení k dispozici jen poté, co proběhne proces zvaný *vyjednání* (negotiation). Vzhledem k tomu, že počítač Pi ohledně svého napájení nevyjednává, po svém připojení k portu USB stolního počítače nebo notebooku nejspíše nebude fungovat.

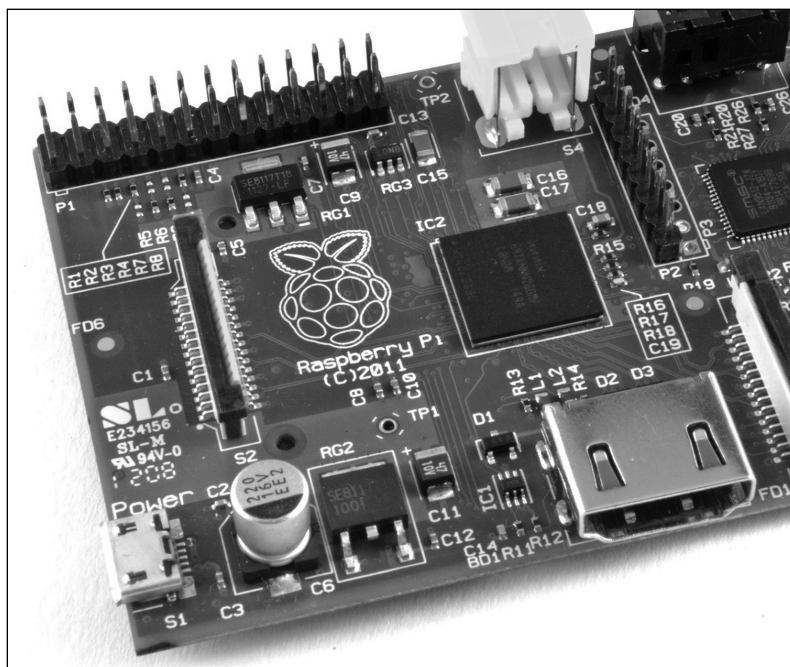
Pokud se při používání počítače Pi projevují občasné problémy (konkrétně funguje-li jen do té doby, než něco připojíte k portu USB nebo spustíte operaci náročnou na procesor, jako je přehrávání videa), je možné, že počítač nemá dostatečnou dodávku napájení. Počítač Pi to umožňuje relativně snadno ověřit díky dvěma *testovacím bodům napětí*.

Chcete-li testovací body napětí použít, potřebujete *voltmetr* nebo *multimetr* se schopností měřit stejnosměrné (DC) napětí. Pokud má voltmetr více vstupů pro různá napětí, použijte odpovídající nastavení.



Upozornění: Snažte se, abyste se testovacími hroty nedotkli počítače nikde jinde než v místě testovacího bodu. Mohli byste propojit napětí 5 V dodávané do počítače Pi s obvody pod interním napětím 3,3 V. Přitom by došlo ke zkratu, který by mohl zařízení poškodit. Dbejte opatrnosti zejména kolem vodivých pinů na konektorech.

Jako dva testovací body slouží malé otvory s měděným povrchem, tzv. *průchodky* (via), které jsou spojeny s obvody počítače Pi pod napětím 5 V a se zemí. Kladný (červený) hrot voltmetru přiložte k testovacímu bodu TP1, který se nachází na levé straně desky přímo nad malou černou komponentou zvanou *regulátor* a označenou RG2. Černý (záporný) hrot voltmetru připojte k testovacímu bodu TP2, který je umístěn mezi měděnými piny portu GPIO a žluto-stříbrným konektorem RCA v levé horní části desky (viz obrázek 3.1).



Obrázek 3.1: Dva testovací body napětí, které jsou označeny TP1 a TP2

Voltmetr by měl ukázat hodnotu mezi 4,8 a 5 V. Pokud naměříte nižší hodnotu než 4,8 V, znamená to, že počítač Pi nedostává dostatek energie. Pokuste se vyměnit adaptér USB za jiný model. Přitom zkontrolujte, zda podle svého štítku dokáže dodávat proud 700 mA nebo více. Lze doporučit modely s jmenovitým proudem 1 A, ale dejte si pozor na levné modely – jejich značení bývá často nepřesné a slibované hodnoty nedokáží poskytnout. Tento problém se u pravých značkových nabíječek mobilních telefonů téměř nevyskytuje, ale raději se vyhněte levným neznačkovým zařízením – která se často prodávají jako *kompatibilní adaptéry*.

Pokud voltmetr ukáže záporné číslo, nelekejte se: znamená to pouze, že jste zaměnili kladný a záporný hrot. Hroty můžete přehodit, ale stačí, když budete při odečtu napětí záporné znaménko ignorovat.

Diagnostika zobrazení

Počítač Pi je sice zkonstruován tak, aby spolupracoval téměř s libovolným zobrazovacím zařízením s připojením HDMI, DVI nebo kompozitního videa, ale po připojení nemusí fungovat podle očekávání. Můžete například zjistit, že je obraz posunut na stranu nebo jej část chybí, je viditelná jen jeho malá část uprostřed obrazovky, je pouze černobílý, případně se neobjevuje vůbec.

Nejdříve zkontrolujte, k jakému typu zařízení je počítač Pi připojen. To je důležité zejména v případech, kdy jste počítač Pi připojili k televizoru pomocí kompozitního připojení RCA. V různých zemích se používají různé standardy televizního obrazu. To znamená, že počítač Pi s konfigurací pro jednu zemi nemusí v jiné zemi fungovat správně. Takto lze obvykle vysvětlit stav, kdy počítač Pi poskytuje černobílý obraz. Postup změny tohoto nastavení naleznete v kapitole 6, „Konfigurace počítače Raspberry Pi“.

Při použití výstupu HDMI je typ zobrazení zpravidla detekován automaticky. Jestliže však připojíte počítač Pi k počítačovému monitoru pomocí adaptéru HDMI na DVI, občas dochází k potížím. Běžně se objevují příznaky jako „sněžení“, chybějící části obrazu nebo úplně černá obrazovka. Chcete-li tyto problémy vyřešit, poznamenejte si rozlišení a obnovovací frekvenci připojené obrazovky a poté přejděte na kapitolu 6, kde najdete postup, jak tyto hodnoty nastavit ručně.

Další problém má podobu příliš velkého nebo příliš malého obrazu, kdy buď chybí části na okraji obrazovky, nebo má kompletní obraz příliš velký černý okraj. To je způsobeno nastavením označovaným jako *overscan*, které v případě připojení počítače Pi k televizoru zajišťuje, aby se k zobrazení nepoužívaly části displeje, které mohou být skryté pod rámečkem. Stejně jako u jiných nastavení souvisejících se zobrazením se v kapitole 6 naučíte parametr *overscan* upravit, nebo dokonce úplně vypnout.

Diagnostika spouštění

Pokud počítač Pi nelze spustit, je to nejčastěji způsobeno problémem s kartou SD. Na rozdíl od stolního počítače nebo notebooku počítač Pi naprosto závisí na souborech, které jsou uloženy na této kartě. Jestliže počítač Pi nedokáže s kartou komunikovat, nezobrazí nic na obrazovce ani neprojeví žádné známky aktivity.

Pokud se kontrolka napájení počítače Pi po připojení napájecího kabelu mikro USB rozsvítí, ale jinak se nic nestane a kontrolka OK zůstane zhasnutá, jedná se o problém s kartou SD. Nejdříve zkontrolujte, zda karta je karta přístupná v PC a zda obsahuje oddíly a soubory, kte-

ré lze očekávat na správně nahané kartě. (Další podrobnosti naleznete v kapitole 2, „Správa systému Linux“, zejména v její části „Struktura systému souborů“.)

V případě, že karta funguje v PC, ale nikoli v počítači Pi, může to znamenat problém s kompatibilitou. Některé karty SD – zejména vysokorychlostní karty označené jako karty typu *Class 10* – při svém vložení do integrované čtečky karet SD v počítači Pi nefungují správně. Seznam karet, u kterých se vyskytly problémy s kompatibilitou v počítači Pi, naleznete na stránce eLinux wiki: http://elinux.org/RPi_VerifiedPeripherals#Problem_SD_Cards.

Máte-li některou z karet uvedených na tomto seznamu, budete ji bohužel muset vyměnit za jinou kartu, aby počítač Pi fungoval. Při vývoji softwarové základny počítače Pi se však programátoři snaží zajistit, aby s počítačem Pi správně spolupracovalo více typů karet. Než svou vysokorychlostní kartu zcela odepíšete, ověřte, zda není k dispozici aktualizovaná verze vaší linuxové distribuce. (Další informace o distribucích naleznete v kapitole 1, „Seznámení s počítačem Raspberry Pi“.)

Diagnostika sítě

Nejužitečnější nástroj na diagnostiku síťových problémů se nazývá `ifconfig`. Používáte-li bezdrátové připojení k síti, přejděte na kapitolu 4, „Konfigurace sítě“, kde naleznete informace o podobném nástroji určeném pro bezdrátová zařízení. Jinak pokračujte ve čtení.

Nástroj `ifconfig`, který má poskytovat informace o připojených síťových portech, nabízí mnoho možností kontroly a konfigurace síťových portů počítače Pi. Chcete-li nástroj začít používat, stačí v okně terminálu zadat samotný jeho název:

```
ifconfig
```

Program `ifconfig` volaný tímto způsobem poskytne informace o všech síťových portech, které nalezne (viz obrázek 3.2). Standardní model B počítače Raspberry Pi je vybaven dvěma porty: fyzickým portem Ethernet na pravé straně desky a virtuálním rozhraním *zpětné smyčky* (loopback), které programům v počítači Pi umožňuje komunikovat mezi sebou.

Výstup příkazu `ifconfig` je rozdělen na následující sekce:

- **Link encap** – typ *zapouzdření* (encapsulation), který se uplatňuje v síti. U modelu B bude uvedeno Ethernet v případě fyzického síťového portu a `Local Loopback` u adaptéru virtuální smyčky.
- **Hwaddr** – *adresa MAC (Media Access Control)* síťového rozhraní zapsaná v šestnáctkové soustavě. Tato adresa je pro všechna zařízení v síti jedinečná a každý počítač Pi má vlastní adresu MAC, která je nastavena při jeho výrobě.
- **inet addr** – *IP adresa (Internet Protocol)* síťového rozhraní. Umožňuje vyhledat počítač Pi v síti, pokud je v počítači spuštěna služba přístupná z jiných počítačů, například webový nebo souborový server.
- **Bcast** – *adresa všesměrového vysílání* (broadcast) sítě, ke které je počítač Pi připojen. Libovolná data odeslaná na tuto adresu budou přijata všemi zařízeními v síti.

- **Mask** – *maska sítě*, která určuje maximální velikost sítě, k níž je počítač Pi připojen. U většiny domácích uživatelů zde bude uvedeno 255.255.255.0.
- **MTU** – velikost parametru *MTU (Maximum Transmission Unit)*, což je maximální velikost dat, která lze přenést v jediném paketu.
- **RX** – tato sekce poskytuje informace o přijatém síťovém provozu včetně zjištěného počtu chyb a zahozených paketů. Pokud se v této sekci začnou objevovat chyby, ukazuje to na potíže se sítí.
- **TX** – tato sekce obsahuje stejné informace jako sekce RX, ale pro odesílané pakety. Opět platí, že všechny zde uvedené chyby znamenají síťové problémy.
- **collisions** – jestliže se dva systémy v síti pokusí komunikovat současně, dojde ke *kolizi*, která vyžaduje, aby obě zařízení své pakety odeslala znovu. Malý počet kolizí nepředstavuje problém, ale vyšší hodnota svědčí o potížích se sítí.
- **txqueuelen** – *délka přenosové fronty*, která je obvykle nastavena na hodnotu 1 000 a zpravidla není nutné ji měnit.
- **RX bytes, TX bytes** – souhrn objemu dat přenesených přes síťové rozhraní.

```

pi@raspberrypi / $ ifconfig
eth0      Link encap:Ethernet  HWaddr b8:27:eb:c7:83:2a
          inet addr:192.168.202.61  Věsměr:192.168.202.255  Mask:255.255.255.0
          AKTIVOVÁNO VĚSMĚROVÉ VYSÍLÁNÍ BĚŽÍ MULTICAST  MTU:1500  Metrika:1
          RX packets:3013 errors:0 dropped:0 overruns:0 frame:0
          TX packets:52 errors:0 dropped:0 overruns:0 carrier:0
          kolizi:0 délka odchozí fronty:1000
          RX bytes:293185 (286.3 KiB)  TX bytes:4164 (4.0 KiB)

lo        Link encap:Místní smyčka
          inet addr:127.0.0.1  Mask:255.0.0.0
          AKTIVOVÁNO SMYČKA BĚŽÍ  MTU:16436  Metrika:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          kolizi:0 délka odchozí fronty:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

pi@raspberrypi / $

```

Obrázek 3.2: Výstup příkazu ifconfig v modelu B počítače Raspberry Pi

Máte-li problémy se síťovým připojením počítače Pi, pokuste se nejdříve vypnout a znovu zapnout síťové rozhraní. Nejjednodušší způsob nabízejí dva nástroje zvané `ifup` a `ifdown`.

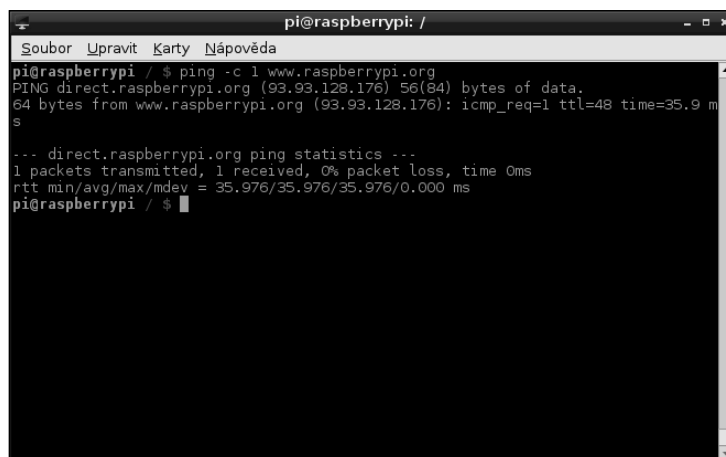
Pokud je síť zapnutá, ale nefunguje správně (když například nástroj `ifconfig` nic nezobrazuje v sekci `inet addr`), začněte tím, že vypnete síťový port. V okně terminálu zadejte následující příkaz:

```
sudo ifdown eth0
```

Jakmile je síť vypnuta, zkontrolujte, zda je kabel pevně zasunut na obou koncích a zda libovolné zařízení, k němuž je počítač Pi připojen (rozbočovač, přepínač či směrovač), je zapnuto a funguje. Potom rozhraní znovu zapněte tímto příkazem:

```
sudo ifup eth0
```

Síťové připojení můžete vyzkoušet příkazem `ping`, který odešle data vzdálenému počítači a vyčká na odpověď. Jestliže vše funguje, měli byste dostat stejnou odpověď, jaká je znázorněna na obrázku 3.3. V opačném případě může být nutné nastavení sítě nakonfigurovat ručně. Postup naleznete v kapitole 4, „Konfigurace sítě“.



```

pi@raspberrypi: /
Soubor  Upravit  Karty  Nápověda
pi@raspberrypi / $ ping -c 1 www.raspberrypi.org
PING direct.raspberrypi.org (93.93.128.176) 56(84) bytes of data.
64 bytes from www.raspberrypi.org (93.93.128.176): icmp_req=1 ttl=48 time=35.9 ms
s

--- direct.raspberrypi.org ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 35.976/35.976/35.976/0.000 ms
pi@raspberrypi / $

```

Obrázek 3.3: Výsledek úspěšného testu sítě příkazem `ping`

Nouzové jádro

Jádro systému Linux tvoří základ operačního systému, díky kterému počítač Pi funguje. Odpovídá za mnoho základních funkcí: od zajištění přístupu k souborům až po komunikaci programů mezi sebou.

Počítač Pi při svém zapnutí načte normální výchozí jádro. Většina distribucí obsahuje také druhé jádro, které zůstává nepoužité. Jedná se o *nouzové jádro* (emergency kernel), a jak je zřejmé z jeho názvu, obvykle se uplatňuje jen tehdy, když nefunguje normální jádro.

Není příliš pravděpodobné, že byste kdykoli potřebovali spouštět počítač Pi pomocí nouzového jádra, ale pro všechny případy stojí za to se s příslušným postupem seznámit. Tyto znalosti jsou důležité zvláště v případech, kdy upgradujete jádro nebo používáte novou a potenciálně nedostatečně testovanou distribuci. Nově vydaný software může občas obsahovat *chyby*, které se před jeho vydáním nepodaří zachytit. Když se po upgradu setkáte s podivnými chybami, můžete pomocí nouzového jádra určit, že problém spočívá v nové verzi jádra.

Jádro systému Linux je tvořeno jediným souborem, který se nachází v adresáři `/boot` a nazývá se `kernel.img`. Když počítač Pi po svém zapnutí začne načítat operační systém, vyhledá tento soubor, a když jej nenalezne, nebude fungovat. Nouzové jádro je uloženo v druhém souboru, který je rovněž umístěn v adresáři `/boot` a má název `kernel_emergency.img`.

Nouzové jádro je ve většině případů téměř shodné se standardním jádrem. Pokud dojde ke změnám standardního jádra, například kvůli zvýšení výkonu nebo přidání nových funkcí, nouzové jádro zůstane beze změn. Pokud změny standardního jádra způsobí potíže se stabilitou, může díky tomu uživatel jednoduše požádat počítač Pi, aby místo standardního jádra zavedl nouzové jádro.

Existují dvě metody, jak spustit nouzové jádro, a pokud počítač Pi nelze spustit, obě tyto metody vyžadují použití počítače PC a čtečky karet SD. Jinak lze následující postup provést v samotném počítači Pi.

Nouzové jádro lze nejsnáze spustit takto: přejmenujte stávající soubor `kernel.img` na `kernel.img.bak` a poté přejmenujte soubor `kernel_emergency.img` na `kernel.img`. Po svém spuštění bude počítač Pi nadále ve výchozím nastavení načítat nouzové jádro. Chcete-li přejít zpět na standardní jádro, stačí celý proces obrátit: přejmenujte soubor `kernel.img` na `kernel_emergency.img` a soubor `kernel.img.bak` na `kernel.img`.

Alternativní metoda spuštění nouzového jádra je založena na úpravě souboru `cmdline.txt` (nachází se v adresáři `/boot`). Na konec existujícího seznamu příkazů zadejte následující položku:

```
kernel=kernel_emergency.img
```

Tato položka sdělí počítači Pi, že má místo obvyklého souboru `kernel.img` načítat jádro s názvem `kernel_emergency.img`. Chcete-li celý postup obrátit, stačí znovu otevřít soubor `cmdline.txt` a příslušnou položku vymazat.

Další informace o souboru `cmdline.txt` a o tom, jak ovlivňuje činnost počítače Raspberry Pi, naleznete v kapitole 6, „Konfigurace počítače Raspberry Pi“.

Konfigurace sítě

Většina uživatelů při konfiguraci síťového připojení počítače Pi vystačí se zasunutím kabelu do ethernetového portu modelu B nebo s připojením ethernetového adaptéru pro USB v případě modelu A. Jiní uživatelé se však neobejdou bez ruční konfigurace.

Pokud víte, že vaše síť neobsahuje *server DHCP* (*Dynamic Host Configuration Protocol*), tento systém poskytuje počítači Pi a jiným zařízením v síti informace o připojení, nebo pokud chcete počítač Pi připojit pomocí bezdrátového adaptéru, pokračujte ve čtení.

Kabelové připojení k síti

Jestliže síť stále nefunguje, možná ji bude nutné nakonfigurovat ručně. Domácí, školní či firemní sítě obvykle zahrnují server DHCP, který počítači Pi a jiným zařízením v síti sděluje, jak se mají připojit. V některých sítích však server DHCP není, a vyžadují proto ruční nastavení.

Seznam síťových rozhraní spolu s informacemi o tom, jak mají být nakonfigurována, je uložen v souboru s názvem `interfaces`, který se nachází v adresáři `/etc/network`. Tento soubor může upravovat pouze uživatel `root`, protože síťové rozhraní odstraněné z tohoto seznamu přestane fungovat.

Z terminálu můžete tento soubor začít upravovat pomocí různých textových editorů. Pro jednoduchost budeme v následujících ukázkách používat textový editor `nano`. Otevřete soubor pro úpravy následujícím příkazem:

```
sudo nano /etc/network/interfaces
```

Nano je výkonný, ale nenáročný textový editor s jednoduchým uživatelským rozhraním (viz obrázek 4.1). Kurzor můžete v dokumentu přesunovat pomocí šipkových kláves, uložit dokument stisknutím kláves `Ctrl+O` a ukončit editor stisknutím kláves `Ctrl+X`.

Řádek, který je potřeba při ruční konfiguraci upravit, začíná textem `iface eth0 inet`. Na konci tohoto řádku odstraňte položku `dhcp` a nahraďte ji klíčovým slovem `static`, začněte nový řádek stisknutím klávesy `Enter` a poté doplňte zbývající podrobnosti v následujícím formátu. Na začátku každého nového řádku vložte tabulátor:

```
[Tab] address xxx.xxx.xxx.xxx  
[Tab] netmask xxx.xxx.xxx.xxx  
[Tab] gateway xxx.xxx.xxx.xxx
```

Na začátku každého řádku je potřeba stisknout klávesu Tab – nezadávejte znaky [Tab]. Znaky x na řádcích konfigurace představují síťové adresy, které je nutné zadat. U položky address zadejte statickou IP adresu, kterou chcete počítači Pi přiřadit. V případě položky netmask zadejte masku sítě, která určuje velikost připojené sítě, v takzvaném *tečkovém desítkovém formátu*. V domácích sítích se obvykle používá maska 255.255.255.0. Na řádku s položkou gateway uveďte IP adresu svého směrovače nebo kabelového modemu.

Uvedme příklad nastavení běžné domácí sítě:

```
iface eth0 inet static
[Tab] address 192.168.0.10
[Tab] netmask 255.255.255.0
[Tab] gateway 192.168.0.254
```



Obrázek 4.1: Úpravy souboru /etc/network/interfaces pomocí editoru nano

Když dokončíte úpravy souboru, uložte jej stisknutím kláves Ctrl+O a poté stisknutím kláves Ctrl+X ukončíte editor nano a přejděte zpět do okna terminálu. Chcete-li, aby se nová síťová nastavení projevila, restartujte *síťovou službu* tímto příkazem:

```
sudo /etc/init.d/networking restart
```

Jestliže se chcete vrátit k automatickému nastavení pomocí protokolu DHCP, musíte znovu upravit soubor `interfaces` a odstranit nastavení `address`, `netmask` a `gateway`. Na konci řádku `iface` nahraďte klíčové slovo `static` za `dhcp` a poté opět restartujte síťovou službu.

Chcete-li svůj počítač Pi připojit k vnějšímu světu, samotné ruční nastavení IP adresy přitom nestačí. Počítače v moderních sítích mají jak číselné adresní identifikátory, které se označují jako IP adresy, tak názvy hostitele či doménové názvy. Právě díky těm druhým (srozumitelným) názvům můžete do svého prohlížeče jednoduše zadat adresu `www.raspberrypi.org` a nemusíte si pamatovat číselnou řadu 93.93.128.176.

Systémy zvané *server DNS (Domain Name Service)* mají za úkol vyhledat k zadaným srozumitelným názvům odpovídající čísla, která umožňují přístup k cílovému systému. Fungují podob-

ně jako automatizované telefonní adresáře. Abyste mohli k internetovým systémům přistupovat pomocí jejich doménových názvů, musíte počítači Pi sdělit, který server DNS má použít.

Seznam serverů DNS, které se v linuxové terminologii označují jako *názvové servery* (nameserver), je uložen v souboru `/etc/resolv.conf`. Když systém zjistí podrobnosti pomocí protokolu DHCP, automaticky je zapíše do tohoto souboru. Při ručním nastavení adresy musíte zadat adresy názvových serverů ve své síti. Obvykle se jedná o adresu směrovače, která je uvedena na řádce gateway souboru `interfaces` (viz popis v předchozí části kapitoly).

Chcete-li nastavit názvové servery, otevřete soubor v editoru nano zadáním následujícího příkazu v okně terminálu:

```
sudo nano /etc/resolv.conf
```

Každý názvový server uveďte na samostatném řádku, který začíná klíčovým slovem `nameserver` a mezerou. Příklad konfigurace sítě v souboru `resolv.conf`, který překládá doménové názvy pomocí veřejně dostupných názvových serverů společnosti Google, vypadá takto:

```
nameserver 8.8.8.8
nameserver 8.8.4.4
```

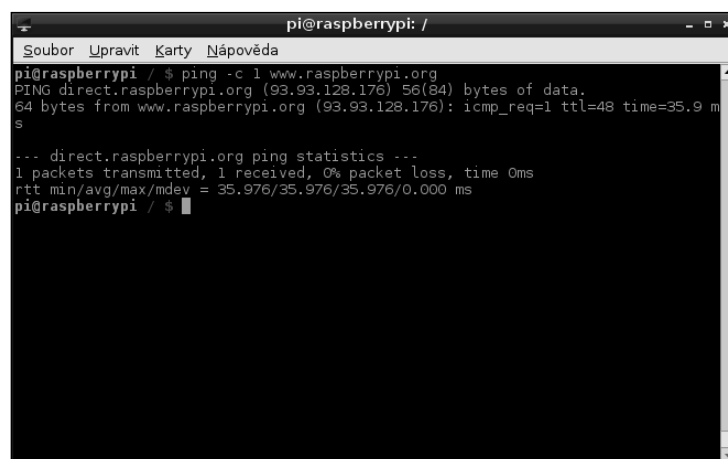
Je zřejmé, že adresy názvových serverů je nutné zadat ve formě IP adresy, a nikoli doménových názvů. Pokud byste uvedli doménové názvy, počítač Pi by se dostal do nekonečného cyklu, když by se pokoušel najít názvový server, aby zjistil, jak může najít názvové servery.

Uložte soubor stisknutím kláves `Ctrl+O` a ukončete editor nano stisknutím kláves `Ctrl+X`. Restartujte síťové rozhraní následujícím příkazem:

```
sudo /etc/init.d/networking restart
```

Potom nastavení vyzkoušejte ve webovém prohlížeči nebo pomocí tohoto příkazu `ping` (viz obrázek 4.2):

```
ping -c 1 www.raspberrypi.org
```



Obrázek 4.2: Úspěšný test připojení modelu B počítače Raspberry Pi k síti

Bezdrátové připojení k síti

Žádný ze současných modelů počítače Raspberry Pi sice neobsahuje integrovaný hardware sítě Wi-Fi, ale bezdrátovou konektivitu lze doplnit pomocí jednoduchého adaptéru Wi-Fi pro USB. Chcete-li však počítač Pi připojit tímto způsobem, musíte adaptér nejdříve nakonfigurovat.



Tip: Adaptéry Wi-Fi pro USB jsou velmi náročné na napájení. Pokud takový adaptér připojíte přímo do portu USB počítače Pi, je pravděpodobné, že nebude fungovat. Lepší je připojit k počítači Pi napájený rozbočovač USB a poté vložit adaptér Wi-Fi do tohoto adaptéru.

Než začnete nastavovat bezdrátové rozhraní, musíte zjistit *identifikátor SSID (Service Set Identifier)* bezdrátového směrovače, ke kterému se chcete připojit (tento identifikátor se také označuje jako *název sítě*). Kromě toho potřebujete znát typ použitého šifrování a požadované heslo. Dále musíte vědět, o jaký typ bezdrátové sítě se jedná. Adaptér USB určený pro síť Wi-Fi standardu 802.11a se nemusí připojit k síti typu 802.11g a naopak.

Aby mohl systém s bezdrátovým adaptérem pro USB komunikovat, je potřeba softwarový balíček označovaný jako *firmware*. Některé distribuce sice mají standardně nainstalováno několik balíčků firmwaru pro nejrozšířenější adaptéry Wi-Fi, ale jiné distribuce nikoli. Většina distribucí určených pro počítač Raspberry Pi v současnosti kvůli úspoře místa vyžaduje ruční instalaci souborů firmwaru pro bezdrátové síťové karty.

To může bohužel vést k situaci, která připomíná Hlavu 22: kvůli stažení souborů firmwaru je potřeba počítač Pi připojit k Internetu. Pokud můžete na pár minut využít kabelový port směrovače či brány, není to žádný problém. Jestliže však nemáte jinou možnost přístupu k Internetu než bezdrátové připojení, musíte instalační balíček firmwaru ručně stáhnout do jiného počítače a poté jej přenést do počítače Pi. Soubor buď můžete nakopírovat na kartu SD počítače Pi, nebo jej přenést na externím úložném zařízení, jako je USB flash disk.

Abyste dokázali najít správný soubor firmwaru ke stažení, musíte zjistit typ svého bezdrátového adaptéru. Na trhu jsou sice k dispozici bezdrátové adaptéry pro USB mnoha značek, ale společností, které skutečně vyrábějí jejich komponenty, je mnohem méně. Několik různých výrobců zpravidla ve svých bezdrátových adaptérech pro USB používá stejný typ čipu, takže jsou všechny jejich výrobky kompatibilní se stejným firmwarem. Chcete-li zjistit, který firmware nainstalovat, obvykle z tohoto důvodu nestačí informace, které jsou uvedeny přímo na zařízení nebo na jeho obalu. Místo toho je potřeba zařízení připojit k počítači Pi a v *cyklické vyrovnávací paměti jádra* (kernel ring buffer) najít chybové zprávy. Pokud jste již bezdrátový adaptér připojili podle pokynů v kapitole 1, „Seznámení s počítačem Raspberry Pi“, můžete pokračovat. V opačném případě připojte adaptér nyní.

Cyklická vyrovnávací paměť jádra je speciální část paměti, do které jádro systému Linux ukládá uživatelsky čitelný výstup. Jedná se o důležitou součást operačního systému Linux: text se při spouštění počítače Pi vypisuje příliš rychle, takže je klíčové, aby si uživatelé mohli zprávy prohlédnout později a vyhledat v nich případné chyby a diagnostikovat problémy.

Když je adaptér připojen, ale přitom nejsou nainstalovány žádné balíčky bezdrátového firmwaru, jádro do cyklické vyrovnávací paměti vypíše řadu chybových zpráv. Chcete-li si tyto zprávy přečíst, můžete vypsát obsah vyrovnávací paměti na obrazovku příkazem `dmesg`. Pokud chcete zobrazit obsah vyrovnávací paměti, zadejte v okně terminálu (nebo v konzole, jestliže jste nespustili prostředí plochy) následující příkaz:

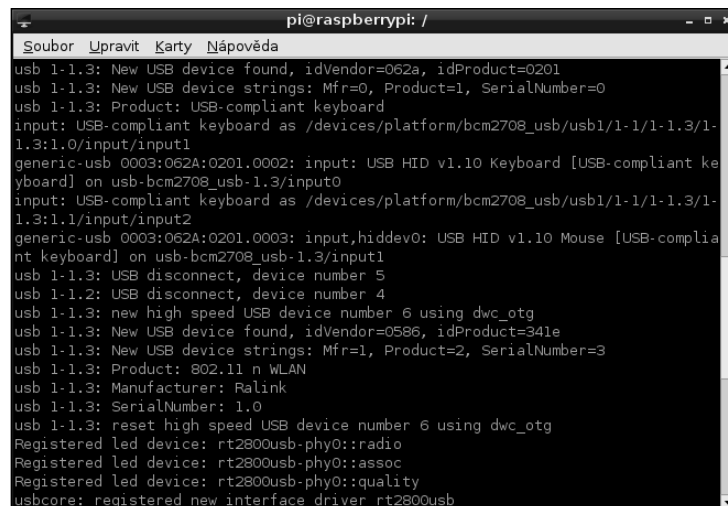
```
dmesg
```

Tento příkaz vypíše celou cyklickou vyrovnávací paměť jádra, která obsahuje všechny zprávy odeslané jádrem od zapnutí počítače Pi. Pokud počítač Pi již nějakou dobu běží, může být tento text poměrně dlouhý. Chcete-li vyhledat chybové zprávy týkající se bezdrátového adaptéru, můžete si usnadnit práci, když výstup příkazu `dmesg` zpracujete nástrojem, který se nazývá `grep`. Pomocí programu `grep` můžete ve vyrovnávací paměti vyhledat text týkající se chybějícího firmwaru. Když *zřetěžíte* výstup programu `dmesg` s nástrojem `grep` a uvedete hledaný termín, získáte mnohem čitelnější výstup. V okně terminálu zadejte následující příkaz:

```
dmesg | grep ^usb
```

Symbol `|` se označuje jako *roura* (pipe) a sděluje systému Linux, že má odeslat výstup jednoho programu (který by obvykle směřoval do souboru nebo na obrazovku) na vstup jiného programu. Tímto způsobem lze do takzvané kolony řetězit více programů. V tomto příkladu má program `grep` za úkol prohledat výstup příkazu `dmesg` (mnoho obrazovek textu z předchozího příkladu) a najít všechny řádky, kde se slovo `usb` vyskytuje na začátku (tuto pozici slova definuje znak `^`).

Konkrétní výsledek tohoto hledání závisí na výrobci použitého bezdrátového adaptéru pro USB. Na obrázku 4.3 je znázorněn výstup pro bezdrátový adaptér „Zyxel NWD2015 Wireless USB Adapter“ připojený k počítači Pi.



```
pi@raspberrypi: /
Soubor  Upravit  Karty  Nápověda
usb 1-1.3: New USB device found, idVendor=062a, idProduct=0201
usb 1-1.3: New USB device strings: Mfr=0, Product=1, SerialNumber=0
usb 1-1.3: Product: USB-compliant keyboard
input: USB-compliant keyboard as /devices/platform/bcm2708_usb/usb1/1-1.3/1-1.3:1.0/input/input1
generic-usb 0003:062A:0201.0002: input: USB HID v1.10 Keyboard [USB-compliant keyboard] on usb-bcm2708_usb-1.3/input0
input: USB-compliant keyboard as /devices/platform/bcm2708_usb/usb1/1-1.3/1-1.3:1.1/input/input2
generic-usb 0003:062A:0201.0003: input,hiddev0: USB HID v1.10 Mouse [USB-compliant keyboard] on usb-bcm2708_usb-1.3/input1
usb 1-1.3: USB disconnect, device number 5
usb 1-1.2: USB disconnect, device number 4
usb 1-1.3: new high speed USB device number 6 using dwc_otg
usb 1-1.3: New USB device found, idVendor=0586, idProduct=341e
usb 1-1.3: New USB device strings: Mfr=1, Product=2, SerialNumber=3
usb 1-1.3: Product: 802.11 n WLAN
usb 1-1.3: Manufacturer: Ralink
usb 1-1.3: SerialNumber: 1.0
usb 1-1.3: reset high speed USB device number 6 using dwc_otg
Registered led device: rt2800usb-phy0::radio
Registered led device: rt2800usb-phy0::assoc
Registered led device: rt2800usb-phy0::quality
usbcore: registered new interface driver rt2800usb
```

Obrázek 4.3: Hledání slova `usb` v cyklické vyrovnávací paměti jádra při připojení bezdrátového adaptéru Zyxel

V tomto výstupu je důležitý řádek s klíčovým slovem **Manufacturer** (výrobce). V případě adaptéru Zyxel NWD2105 je zde uvedena hodnota **Ralink**, což je skutečný výrobce čipu, který se nachází v bezdrátovém adaptéru Zyxel pro USB. Chcete-li bezdrátový adaptér zprovoznit, musíte instalovat právě firmware této společnosti.



Tip: Pokud se vám pomocí vyhledávacího termínu **usb** nepodaří nic najít, můžete hledání opakovat s klíčovými slovy **firmware**, **wlan** (bezdrátová síť) nebo **wireless** (bezdrátový). Jestliže ani potom nenajdete žádné užitečné informace, vypište pomocí příkazu **lsusb** seznam všech zařízení USB připojených k systému.

S použitím názvu výrobce, který jste zjistili ve výstupu příkazu **dmesg**, vyhledejte soubory firmwaru v nástroji **apt-cache** (viz druhá kapitola). V případě ukázkového adaptéru Zyxel NWD2015 by příkaz pro nástroj **apt-cache** vypadal takto:

```
apt-cache search ralink
```

Jestliže se nástroji **apt-cache** nepodaří firmware najít, musíte odhadem vybrat některou z možností uvedených v následujícím seznamu. Nebojte se, že omylem nainstalujete nesprávný balíček – každý firmware můžete rychle odinstalovat příkazem **apt-get remove** a ani více balíčků firmwaru nezpůsobí v systému žádné problémy. V doporučené distribuci Debian pro počítač Raspberry Pi jsou k dispozici následující balíčky firmwaru pro bezdrátové adaptéry:

- **atmel-firmware** – pro zařízení založená na čipových sadách Atmel AT76C50X
- **firmware-atheros** – pro zařízení založená na čipových sadách Atheros
- **firmware-brcm80211** – pro zařízení založená na čipových sadách Broadcom
- **firmware-intelwimax** – pro zařízení založená na čipových sadách Intel WiMAX
- **firmware-ipw2x00** – pro adaptéry Intel Pro Wireless (včetně řad 2100, 2200 a 2915)
- **firmware-iwlwifi** – pro jiné bezdrátové adaptéry Intel (včetně řad 3945, 4965 a 5000)
- **firmware-ralink** – pro zařízení založená na čipových sadách Ralink
- **firmware-realtek** – pro zařízení založená na čipových sadách Realtek
- **zd1211-firmware** – pro zařízení založená na čipové sadě ZyDAS 1211

Firmware pro ukázkový bezdrátový adaptér Zyxel je k dispozici v balíčku **firmware-ralink** na tomto seznamu. Tento balíček lze nainstalovat příkazem **apt-get**, ale pouze tehdy, je-li počítač Pi připojen k Internetu pomocí portu kabelové sítě Ethernet nebo pomocí ethernetového adaptéru pro USB. Po připojení k síti nainstalujte firmware tímto příkazem:

```
sudo apt-get install balíček_firmwaru
```

Výraz *balíček_firmwaru* v tomto příkazu nahraďte názvem balíčku, který jste vyhledali příkazem **apt-cache**. V případě adaptéru Zyxel NWD2105 by kompletní příkaz vypadal takto:

```
sudo apt-get install firmware-ralink.
```

Instalace bezdrátového firmwaru offline

Pokud počítač Pi nemůžete připojit k Internetu jinak než bezdrátově, musíte stáhnout firmware do jiného počítače. Ve webovém prohlížeči otevřete stránku vyhledávače a zadejte název balíčku firmwaru následovaný použitou distribucí a její verzí.

Jestliže pracujete s doporučenou distribucí Debian, můžete firmware pro čipovou sadu Ralink RT2x00 z výše uvedeného příkladu najít pomocí hledané fráze **firmware-ralink debian wheezy**. Výsledkem hledání bude *soubor balíčku* ke stažení. V případě distribuce Debian se bude jednat o soubor s příponou **.deb**. U distribuce Fedora Remix bude stejný firmware k dispozici jako soubor **.rpm**.

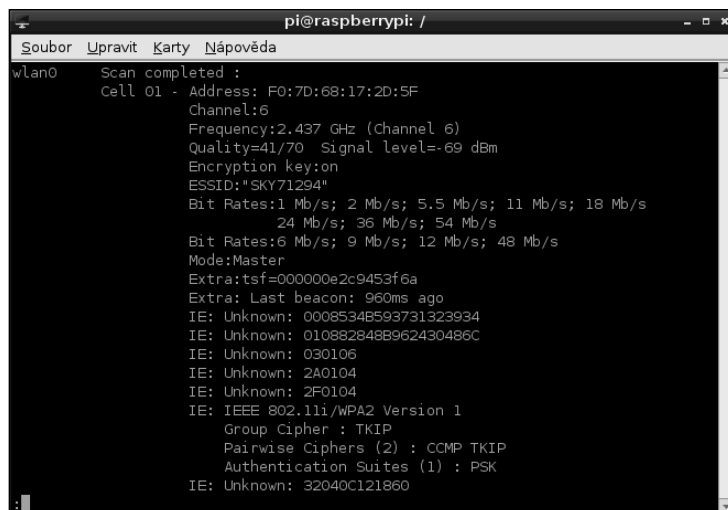
Stáhněte tento soubor a poté jej zkopírujte do adresáře **/home/pi** karty SD počítače Pi, případně na USB flash disk nebo jiné externí úložné zařízení. Spustěte počítač Pi a ve fázi instalace firmwaru nahradte název balíčku názvem staženého souboru. V případě ukázkové karty Zyxel NWD2105 by příkaz vypadal takto:

```
sudo apt-get install firmware-ralink_0.35_all.deb
```

Po instalaci firmwaru bezdrátový adaptér pro USB odpojte od počítače Pi a znovu jej připojte. Tím znovu spustíte hledání souborů firmwaru jádrem, ale nyní již budou soubory jádru k dispozici. Soubory zůstanou na svém místě a budou se po připojení bezdrátového adaptéru pro USB načítat automaticky. Proces instalace stačí provést pouze jednou.

Po instalaci firmwaru by mělo být nastavení bezdrátového připojení jednoduché. Nejdříve zkontrolujte, zda bezdrátový adaptér pro USB funguje očekávaným způsobem. Za tímto účelem prohledejte dostupné přístupové body bezdrátové sítě příkazem **iwlist**. Seznam se pravděpodobně nevejde na jednu obrazovku, takže výstup příkazu propojte s příkazem **less**, aby se výpis pozastavil po každé obrazovce:

```
sudo iwlist scan | less
```



Obrázek 4.4: Skenování bezdrátových sítí příkazem iwlist

Uvedený příkaz vrátí seznam všech bezdrátových sítí, které jsou z počítače Pi dosažitelné, spolu s příslušnými podrobnostmi (viz obrázek 4.4). Pokud se v této fázi zobrazí chybová zpráva, konkrétně informace o vypnuté síti nebo rozhraní, zkontrolujte, zda jste nainstalovali správný firmware a zda je bezdrátový adaptér pro USB připojen k napájenému rozbočovači USB.

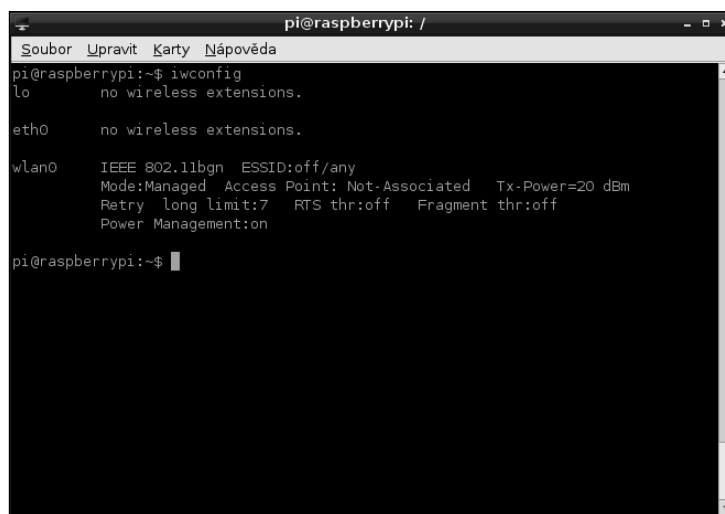
Aktuální stav sítě je možné zjistit pomocí příkazu `iwconfig`. Podobně jako program `ifconfig` také nástroj `iwconfig` dovoluje zkontrolovat stav síťového rozhraní a zadávat konfigurační příkazy. Na rozdíl od programu `ifconfig` je však nástroj `iwconfig` určen speciálně pro bezdrátové sítě a obsahuje příslušné funkce pro práci s nimi. V okně terminálu zadejte název příkazu:

```
iwconfig
```

Výstup příkazu `iwconfig` je rozdělen do několika sekcí, jak je zřejmé na obrázku 4.5:

- **Interface Name** – stejně jako u kabelových sítí má každé zařízení vlastní název rozhraní. Pokud se jedná o rozhraní bezdrátového připojení, budou uvedeny další podrobnosti. Bezdrátové připojení počítače Pi se standardně označuje jako `wlan0`.
- **Standard** – existují různé typy bezdrátových standardů IEEE 802.11, které se rozlišují dodatečným písmenem. V této sekci jsou uvedeny standardy, které bezdrátový adaptér pro USB podporuje. V případě ukázkového adaptéru jsou uvedeny typy sítě IEEE 802.11bgn, které dokáže adresovat.
- **ESSID** – hodnota SSID sítě, k níž je adaptér připojen. Jestliže adaptér aktuálně není připojen k síti, bude uvedena hodnota `off/any`.
- **Mode** – režim, ve kterém adaptér aktuálně funguje. Může se jednat o jeden z následujících režimů:
 - **Managed** – standardní bezdrátová síť, kde se klienti připojují k přístupovým bodům. Tento režim se používá téměř ve všech domácích i podnikových sítích.
 - **Ad-Hoc** – bezdrátová síť, kde zařízení komunikují přímo bez přístupových bodů.
 - **Monitor** – speciální režim, ve kterém karta naslouchá veškerému provozu nezávisle na tom, zda je adresován pro ni. Tento režim se obvykle používá při řešení potíží se sítí, aby bylo možné zachytávat kompletní provoz v bezdrátové síti.
 - **Repeater** – speciální režim, který vynutí, aby bezdrátová karta předávala provoz jiným síťovým klientům. Umožňuje zvýšit sílu signálu.
 - **Secondary** – podřízený režim režimu Repeater, ve kterém bezdrátová karta funguje jako záložní opakovač.
- **Access Point** – adresa přístupového bodu, k němuž je bezdrátový adaptér aktuálně připojen. Jestliže adaptér není připojen k žádnému bezdrátovému přístupovému bodu, bude zde uvedeno `Not-Associated`.
- **Tx-Power** – vysílací výkon bezdrátového adaptéru. Zde uvedená hodnota označuje sílu signálu vysílaného adaptérem: čím je číslo vyšší, tím je signál silnější.
- **Retry** – aktuální nastavení počtu opakování bezdrátového adaptéru. Tato hodnota se uplatňuje v přetížených sítích. Obvykle ji není nutné měnit a některé karty její změnu ani neumožňují.

- **RTS** – aktuální nastavení adaptéru pro proces handshaking označovaný jako *Ready To Send* a *Clear To Send (RTS/CTS)*, který zabráňuje kolizím v sítích s intenzivními přenosy. Tuto hodnotu obvykle nastavuje přístupový bod během připojení.
- **Fragment** – maximální velikost fragmentu, která v rušných sítích definuje dělení paketů na více fragmentů. Tuto hodnotu obvykle nastavuje přístupový bod během připojení.
- **Power Management** – aktuální stav funkce řízení spotřeby adaptéru, která omezuje požadavky zařízení na napájení v době, kdy je bezdrátová síť nečinná. U počítače Pi nemá velký význam, ale zpravidla se aktivuje u zařízení napájených bateriemi, jako jsou notebooky.



```

pi@raspberrypi: /
Soubor  Upravit  Karty  Nápověda
pi@raspberrypi:~$ iwconfig
lo        no wireless extensions.

eth0      no wireless extensions.

wlan0     IEEE 802.11bgn  ESSID:off/any
          Mode:Managed  Access Point: Not-Associated  Tx-Power=20 dBm
          Retry  long limit:7   RTS thr:off   Fragment thr:off
          Power Management:on

pi@raspberrypi:~$

```

Obrázek 4.5: Výstup příkazu `iwconfig`, když není připojena bezdrátová síť

Chcete-li počítač Pi připojit k bezdrátové síti, musíte doplnit několik řádků do souboru `/etc/network/interfaces`. (Podrobnosti o struktuře tohoto souboru naleznete v sekci „Kabelové připojení k síti“ v předchozí části této kapitoly.) Nejdříve soubor otevřete v textovém editoru `nano`:

```
sudo nano /etc/network/interfaces
```

Na konci souboru vytvořte novou položku pro bezdrátový adaptér pro USB, která bude vypadat následovně (viz obrázek 4.6):

```

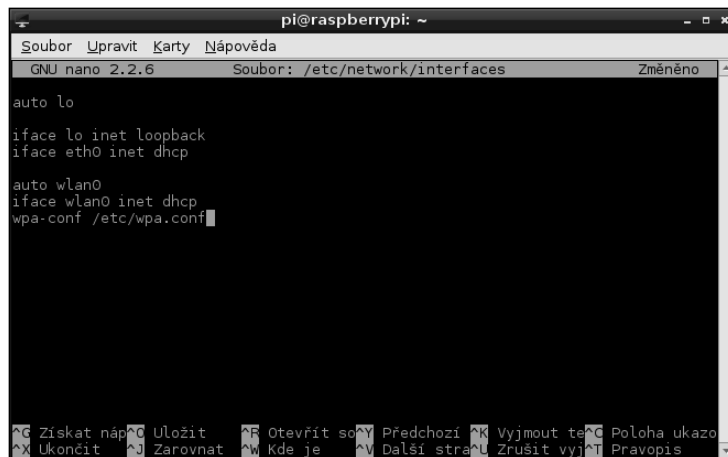
auto wlan0
iface wlan0 inet dhcp
wpa-conf /etc/wpa.conf

```

Když položku vytvoříte, uložte soubor stisknutím kláves `Ctrl+O` a ukončete editor `nano` stisknutím kláves `Ctrl+X`.



Tip: Identifikátor zařízení `wlan0` platí v případě, že se jedná o první bezdrátové zařízení, které jste v počítači Pi nastavili. Jinak se bude číslo na konci lišit. Zobrazte aktuální seznam bezdrátových zařízení příkazem `iwconfig` a odpovídajícím způsobem změňte řádky v předchozí ukázce kódu.



Obrázek 4.6: Úpravy souboru interfaces kvůli přístupu k bezdrátové síti

Poslední řádek souboru `interfaces` odkazuje na konfigurační soubor `wpa.conf`, který zatím neexistuje. Tento soubor používá nástroj s názvem `wpa_supplicant`, díky kterému se systém Linux může snadno připojit k sítím zabezpečeným pomocí šifrování WPA (*Wireless Protected Access*).

Pomocí nástroje `wpa_supplicant` lze počítač Pi připojit téměř k libovolné bezdrátové síti bez ohledu na to, zda je chráněna pomocí standardu WPA nebo jeho novější verze WPA2. Nástroj přitom podporuje režimy šifrování AES (*Advanced Encryption Standard*) i TKIP (*Temporal Key Integrity Protocol*). Navzdory svému názvu dovoluje program `wpa_supplicant` připojení i k bezdrátovým sítím, které používají starší šifrovací standard WEP (*Wired Equivalent Privacy*).

Program `wpa_supplicant` ukládá svou konfiguraci do souboru `wpa.conf`, který je umístěn v adresáři `/etc`. Chcete-li počítač Pi začít konfigurovat k bezdrátovému přístupu, nejdříve otevřete nový prázdný soubor pro úpravy tímto příkazem:

```
sudo nano /etc/wpa.conf
```

Zadejte následující dva řádky, které jsou opět stejné pro libovolný typ bezdrátové sítě. Parametr `Identifikátor_SSID` nahradte identifikátorem SSID bezdrátové sítě, k níž se chcete připojit, a na konec souboru vložte řádky, které odpovídají typu šifrování dané sítě.

```
network={
[Tab] ssid="Identifikátor_SSID"
```

Další část konfiguračního souboru závisí na typu konfigurované bezdrátové sítě. V následujících sekcích naleznete pokyny, jak dokončit konfiguraci pro nezabezpečené sítě a sítě s šifrováním WEP nebo WPA.

Bez šifrování

Jestliže se ve vaší bezdrátové síti nepoužívá žádné šifrování, dokončete soubor `wpa.conf` takto:

```
[Tab] key_mgmt=NONE
}
```

Uložte soubor stisknutím kláves Ctrl+O a ukončete editor nano stisknutím kláves Ctrl+X.

Šifrování WEP

Pokud bezdrátová síť pracuje s šifrováním WEP, dokončete soubor `wpa.conf` takto:

```
[Tab] key_mgmt=NONE
[Tab] wep_key0="Klíč_WEP"
}
```

Místo parametru `Klíč_WEP` zadejte klíč ASCII, který slouží k šifrování WEP v síti. Uložte soubor stisknutím kláves Ctrl+O a ukončete editor nano stisknutím kláves Ctrl+X.



Tip: Šifrování WEP je mimořádně slabé. Pomocí snadno dostupného softwaru lze zjistit hodnotu síťového šifrovacího klíče WEP za pouhých několik minut, takže se mohou do vaší sítě dostat neautorizovaní uživatelé. Pokud šifrování WEP dosud používáte, zvažte přechod na šifrování WPA nebo WPA2, které zajistí vyšší bezpečnost.

Šifrování WPA/WPA2

Pokud je bezdrátová síť chráněna šifrováním WPA nebo WPA2, dokončete soubor `wpa.conf` takto:

```
[Tab] key_mgmt=WPA-PSK
[Tab] psk="Klíč_WPA"
}
```

Parametr `Klíč_WPA` nahraďte šifrovací frází příslušné bezdrátové sítě. Na obrázku 4.7 je znázorněna ukázková konfigurace bezdrátové sítě s identifikátorem SSID „Private_Router“ a šifrovací frází WPA „Private Wireless Network“. Uložte soubor stisknutím kláves Ctrl+O a ukončete editor nano stisknutím kláves Ctrl+X.



Obrázek 4.7: Úpravy souboru `wpa.conf` pro síť s ochranou WPA

Připojení k bezdrátové síti

Připojení počítače Pi k bezdrátové síti je nyní nakonfigurováno a aktivuje se po dalším restartu počítače. Chcete-li se k bezdrátové síti připojit bez restartu, zadejte příkaz:

```
sudo ifup wlan0
```

Pokud chcete ověřit, zda je síť funkční, odpojte ethernetový kabel počítače Pi (je-li připojen) a zadejte následující příkaz:

```
ping -c 1 www.raspberrypi.org
```



Tip: Jestliže se po instalaci bezdrátového adaptéru pro USB začnou objevovat problémy s počítačem Pi, mohou být příčinou konflikty s jinými zařízeními USB. O některých modelech adaptérů je známo, že způsobují problémy s jistými klávesnicemi USB. Aktuální seznam adaptérů, které byly úspěšně vyzkoušeny, i těch, u kterých se vyskytly konflikty, naleznete na stránce <http://www.element14.com/community/docs/DOC-44703/1/raspberry-pi-wifi-adapter-testing> nebo na stránce eLinux wiki http://elinux.org/RPi_VerifiedPeripherals#Working_USB_Wifi_Adapters.

Správa oddílů

Získáte-li operační systém počítače Raspberry Pi jako bitovou kopii cizí karty SD, je to sice pohodlné, ale poněkud omezující. Většina bitových kopií distribucí určených ke stažení předpokládá uložení na kartu SD s kapacitou 2 GB nebo 4 GB. To znamená, že uživatelé karet velikosti 8 GB nebo větších nemohou využít minimálně polovinu místa na své kartě.

Vytvoření nového oddílu

Jeden způsob, jak maximálně využít větší kartu SD v počítači Raspberry Pi, spočívá ve vytvoření nového oddílu v prázdném prostoru na konci karty. Tento oddíl umožňuje ukládat libovolné velké soubory, k nimž chcete ze svého počítače Pi přistupovat, aniž byste museli připojovat externí úložné zařízení.

Všechny fáze této operace lze provést přímo z počítače Pi, aniž byste kartu SD museli vyjmát a připojovat ji k počítači PC. Stačí mít k dispozici kartu SD, na kterou je nahrána jedna z bitových kopií systému Linux pro Raspberry Pi (viz popis v kapitole 1, „Seznámení s počítačem Raspberry Pi“), a trochu volného místa.

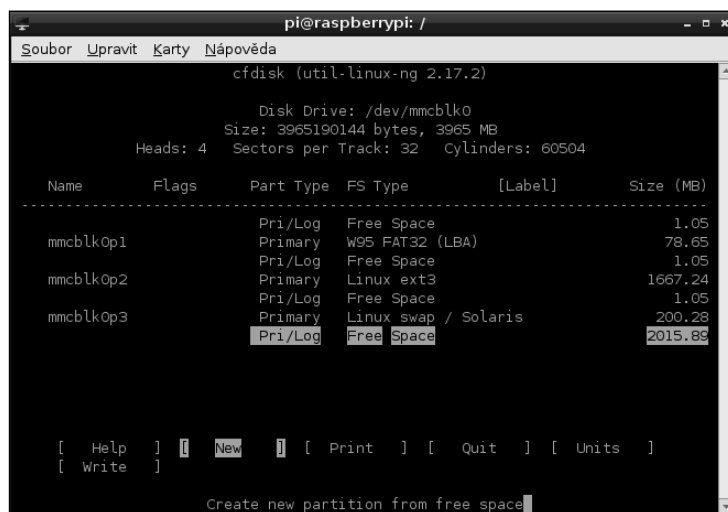


Upozornění: Při práci s nástroji, které dokáží modifikovat oddíly na disku, musíte dbát opatrnosti, abyste neodstranili žádný existující oddíl. Konkrétně dávejte pozor, abyste neodstranili oddíl / **boot** nebo oddíly kořenového systému souborů, protože by počítač Pi nebylo možné spustit.

Chcete-li na kartě SD vytvořit nový oddíl, postupujte takto:

1. Otevřete okno terminálu a příkazem `sudo fdisk -l` vypíšete seznam úložných zařízení, která jsou připojena k počítači Pi, a jejich aktuálních oddílů. Karta SD bude uvedena jako zařízení `/dev/mmcblk0` s řadou oddílů označených `p0`, `p1` atd.
2. Příkazem `sudo cfdisk /dev/mmcblk0` spusíte nástroj na správu oddílů `cfdisk`, který lze ovládat pomocí nabídek. Tento příkaz použije kartu SD jako cílové zařízení. Jestliže chcete vytvořit oddíl na zařízení USB Mass Storage, nahraďte výrazem `mmcblk0` identifikátor příslušného úložného zařízení (např. `sda`).
3. Pomocí kurzorových kláves přesuňte zvýraznění výběru oddílu – které zobrazuje aktuálně vybraný oddíl inverzně, obvykle černou barvou na bílé – po seznamu oddílů směrem dolů až k položce s názvem Free Space na konci seznamu.

- Opět použijte kurzorové klávesy a přesuňte zvýraznění výběru oddílu, které se nachází v dolní části obrazovky a má inverzní vzhled, na položku New a stiskněte klávesu Enter (viz obrázek 5.1).



Obrázek 5.1: Vytvoření nového oddílu nástrojem cfdisk

- Zobrazí se výzva k vytvoření primárního (Primary) nebo logického (Logical) oddílu. Ve výchozím nastavení se vytváří primární oddíl, takže stačí tuto hodnotu potvrdit stisknutím klávesy Enter.
- Další výzva požádá o zadání velikosti oddílu v megabajtech (MB). Standardně se vytváří oddíl, který zaplní veškeré dostupné volné místo na zařízení, takže opět stačí stisknout klávesu Enter.
- V dalším kroku je potřeba zapsat na disk nové údaje, které se označují jako *tabulka oddílů* (partition table). Pomocí kurzorových kláves přesuňte zvýraznění výběru možnosti na položku Write a stiskněte klávesu Enter.
- Zobrazí se výzva, abyste ověřili, zda jsou provedené změny správné. Pečlivě zkontrolujte, zda jste vytvořili oddíl podle uvedených pokynů, a poté odpovězte zadáním yes a stiskněte Enter.
- Chcete-li nástroj cfdisk ukončit, přesuňte pomocí kurzorových kláves zvýraznění výběru možnosti v dolní části obrazovky na položku Quit a stiskněte klávesu Enter.

Tabulka oddílů byla sice nyní aktualizována tak, aby obsahovala nově vytvořený oddíl, ale pro operační systém nebude oddíl viditelný, dokud jej znovu nenačtete. To lze nejsnáze zajistit restartem počítače Pi pomocí následujícího příkazu:

```
sudo reboot
```

Když se počítač Pi restartuje a znovu se přihlásíte, můžete nástrojem fdisk ověřit, že je nový oddíl připraven k použití. Zadejte následující příkaz:

```
sudo fdisk -l
```

Než však můžete na nový oddíl začít ukládat soubory, musíte jednotku pomocí nástroje `mkfs` naformátovat. Postupujte přitom takto:

1. Příkazem `sudo fdisk -l` vypíšete seznam úložných zařízení, která jsou připojena k počítači Pi, a jejich aktuálních oddílů. Všimněte si nového oddílu, který bude zobrazen jako `/dev/mmcblk0pN`, kde `N` je číslo oddílu. Pokud pracujete s doporučenou distribucí Debian, bude mít oddíl označení `/dev/mmcblk0p3`.
2. Vytvořte na oddílu nový systém souborů EXT4 zadáním příkazu `sudo mkfs.ext4 /dev/mmcblk0pN`, přičemž parametr `N` nahraďte číslem nového oddílu. Zkontrolujte, zda jste vybrali správný oddíl: příkaz `mkfs` (*make file system* – vytvoř systém souborů) smaže na formátovaném oddílu všechna data.
3. Abyste mohli nový systém souborů začít používat, musíte jej *připojit*. Vytvořte *přípojný bod* (prázdný adresář) zadáním příkazu `sudo mkdir /uloziste` v okně terminálu.
4. Pomocí příkazu `mount` zpřístupněte nový oddíl v přípojném bodě, který jste právě vytvořili. Zadejte příkaz `sudo mount /dev/mmcblk0pN /uloziste`, kde `N` je číslo oddílu.



Upozornění: Pokud na oddílu vytváříte nový prázdný systém souborů pomocí příkazu `mkfs`, nejdříve vždy pečlivě zkontrolujte podrobnosti o oddílu. Jestliže v příkazu `mkfs` uvedete chybný oddíl, příkaz vymaže všechny soubory, které byly na oddílu uloženy. V případě, že smažete oddíl `/boot` nebo kořenový systém souborů, počítač Pi nebude možné spustit, dokud znovu nenahrajete systém na kartu SD.

Tímto způsobem sice získáte úložné zařízení, které je přístupné pro superuživatele `root`, ale standardní uživatel `pi` ani žádné uživatelské účty, které jste vytvořili sami, aktuálně nemají k ukládání souborů na tomto oddílu dostatečná oprávnění. To lze napravit pomocí trojice příkazů: `chown`, `chgrp` a `chmod`.

Název prvního příkazu `chown` je zkratkou slov *change ownership* (změnit vlastnictví) a umožňuje soubory vytvořené jedním uživatelem předat jinému uživateli. Příkaz `chgrp` nastaví skupinu, které má soubor patřit, aby k němu mohli přistupovat všichni členové této skupiny, a příkaz `chmod` upraví oprávnění k souboru nebo adresáři.

Chcete-li poskytnout přístup k novému oddílu všem uživatelům, změňte příkazem `chgrp` členství ve skupině ze skupiny `root` na skupinu `users`. Uvedte přitom příznak `-R` (rekurzivní), aby se příkaz vztahoval na celý obsah adresáře:

```
sudo chgrp -R users /uloziste
```

Potřebujete také, aby všichni členové skupiny dokázali do adresáře zapisovat. Za tímto účelem zadejte příkaz `chmod` s parametrem `g+w`, který příkaz `chmod` instruuje, aby členům skupiny povolil přístup pro zápis:

```
sudo chmod -R g+w /uloziste
```

Nový oddíl je nyní připraven k použití, ale čeká vás ještě jeden úkol. Aktuálně je po každém restartu počítače Pi potřeba oddíl připojovat ručně (pomocí příkazu `mount` doplněného o pří-

kaz sudo kvůli spuštění jako uživatel *root*). Chcete-li ušetřit čas, můžete počítač Pi nastavit tak, aby nový oddíl připojoval automaticky. Zajistíte to úpravou souboru *fstab*.

Soubor *fstab* (zkratka slov *file system table* – tabulka systému souborů), který se nachází v adresáři */etc*, informuje systém Linux, které systémy má připojit a ke kterým přípojným bodům. Tato tabulka může na první pohled vypadat složitě, ale používá logickou sloupcovou strukturu.

Ze sloupců zleva doprava systém Linux postupně zjišťuje umístění připojovaného zařízení, adresář, kde má být zařízení přístupné (přípojný bod), typ systému souborů, případné požadované parametry a nakonec dvě čísla, která určují, zda má být systém souborů *vypsán* (dump) v případě systémového problému a zda jej má zkontrolovat nástroj *fsck* (file system check – kontrola systému souborů).

Pokud chcete, aby systém připojoval nový oddíl automaticky, nejdříve otevřete soubor *fstab* v editoru *nano*:

```
sudo nano /etc/fstab
```

Na konec souboru přidejte nový řádek, kde budou definovány různé potřebné parametry nového oddílu. Jednotlivé položky přitom oddělujte tabulátory:

```
/dev/mmcblk0pN [Tab]/u1oziste[Tab] ext4[Tab] defaults[Tab] 0[Tab] 2
```

Nezapomeňte změnit parametr *N* na číslo nového oddílu (viz obrázek 5.2). Jestliže pomocí souboru *fstab* připojujete externí úložná zařízení, použijte název zařízení */dev/sdXN*, kde *X* je písmeno jednotky a *N* znamená číslo oddílu. Uložte soubor stisknutím kláves *Ctrl+O* a ukončete editor *nano* stisknutím kláves *Ctrl+X*. Když přejdete zpět do okna terminálu, restartujte systém a zadáním příkazu *mount* zkontrolujte, zda je adresář */u1oziste* automaticky připojen. Pokud nikoli, pečlivě si prohlédněte svůj nový záznam v souboru *fstab*. Nezapomeňte, že po ukončení každé položky je nutné stisknout klávesu *Tab*.



Obrázek 5.2: Úpravy souboru *fstab*, aby se nový oddíl připojoval automaticky

Změna velikosti stávajících oddílů

Vytvoření nového oddílu představuje jeden způsob, jak využít místo na větší kartě SD, ale tento způsob není zrovna nejpružnější. Lepší metoda je založena na změně velikosti existujících oddílů tak, aby obsadily volné místo. Chcete-li to provést bezpečně, musíte odpojit počítač Pi od napájení, vyjmout z něj kartu SD a vložit ji do čtečky karet připojené ke stolnímu počítači nebo notebooku.

Automatická změna velikosti

Linuxová distribuce Debian pro počítač Raspberry Pi obsahuje nástroj zvaný `raspi-config`, který se načítá při prvním spuštění systému. Lze jej také kdykoli spustit ručně zadáním příkazu `sudo raspi-config` v konzole nebo okně terminálu. Tento nástroj poskytuje rozhraní pro mnoho běžných úkolů konfigurace, včetně automatické změny velikosti kořenového systému souborů tak, aby se plně využilo dostupné místo na kartě SD.

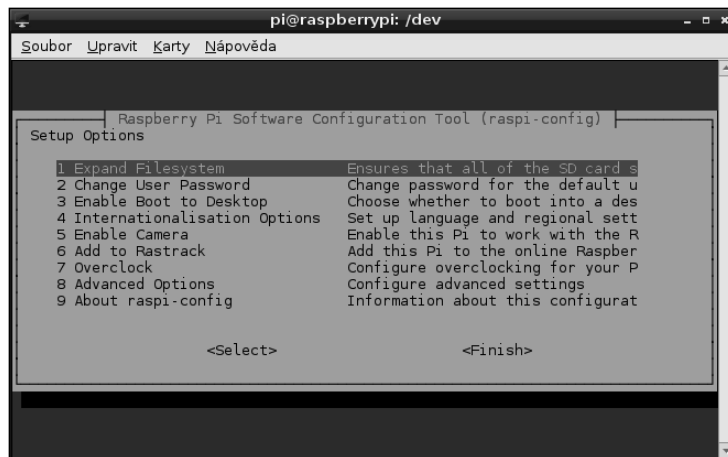


Upozornění: Změna velikosti kořenového systému souborů nástrojem `raspi-config` může ve výjimečných případech způsobit poškození dat. Pokud máte v počítači Pi uložena data, o něž nechtete přijít, nejdříve tato data zálohujte nebo postupujte podle spolehlivějších pokynů na ruční změnu velikosti oddílů, které naleznete v další části kapitoly.

Chcete-li změnit velikost kořenového systému souborů nástrojem `raspi-config`, postupujte takto:

1. Pokud jste systém Debian v počítači Raspberry Pi spustili poprvé, nástroj `raspi-config` bude načten automaticky. Jinak jej spusťte ručně příkazem `sudo raspi-config` v konzole nebo okně terminálu.
2. V nabídce nástroje `raspi-config` (viz obrázek 5.3) stiskněte klávesu se šipkou dolů, abyste zvýraznili možnost `expand_rootfs` (rozšířit kořenový systém souborů), a poté stiskněte klávesu `Enter`.
3. Operace změny velikosti trvá pouze několik sekund a poté se zobrazí zpráva, že proces bude dokončen po dalším restartu počítače Pi. Zprávu zavřete stisknutím klávesy `Enter`.
4. Dvojitým stisknutím klávesy `Tab` zvýrazněte položku `Exit` a ukončete nástroj `raspi-config` stisknutím klávesy `Enter`.
5. Restartujte počítač Pi pomocí příkazu `sudo reboot`. Proces restartu bude trvat déle než obvykle, protože je nutné změnit velikost systému souborů. Tento proces proběhne při každé změně velikosti jen jednou. Další restart počítače Pi již bude trvat standardně dlouho.

Po dokončení restartu počítače Pi bude mít kořenový systém souborů takovou velikost, jakou karta SD umožňuje. Chcete-li to ověřit, zadejte v okně terminálu příkaz `df -h`, který zobrazí volné místo na všech připojených úložných zařízeních.



Obrázek 5.3: Obrazovka s nabídkou nástroje raspi-config

Ruční změna velikosti

Nejspolehlivější způsob, jak změnit velikost oddílů systému Linux ve stolním počítači nebo notebooku, představuje nástroj s názvem *Parted Magic*. Toto bezplatně dostupné spustitelné CD je určeno k úpravám systémů souborů.

Disk funguje v počítačích PC i Mac a spouští se výhradně z operační paměti. Díky tomu se nepokouší nahradit stávající operační systém. Na rozdíl od nástroje *raspi-config*, specifického pro distribuci Debian, je také kompatibilní se všemi distribucemi pro počítač Raspberry Pi.



Tip: Pokud používáte systém Linux, můžete si nainstalovat grafický nástroj pro správu oddílů **gparted**, který je součástí disku Parted Magic, a nemusíte spouštět tento disk. V distribucích založených na systému Debian stačí k načtení programu zadat příkaz `sudo apt-get install gparted` následovaný příkazem `sudo gparted`.

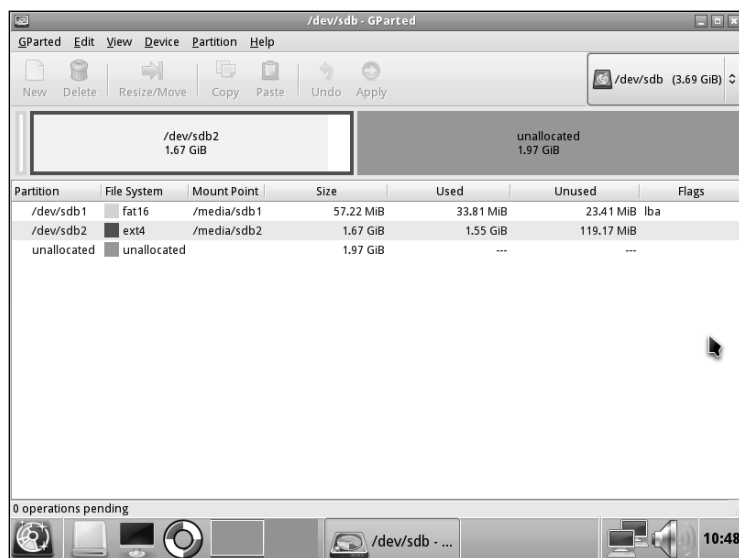
Soubor bitové kopie ISO disku Parted Magic stáhněte z webu <http://partedmagic.com> a zapište jej na prázdný disk CD či DVD pomocí dostupného nástroje na vypalování disků. Ponechte disk v jednotce a restartujte počítač. Načte se systém nabídek disku Parted Magic. Chcete-li zavést vlastní software, zvolte možnost Standard Settings.

Parted Magic je přizpůsobený operační systém Linux, který obsahuje nástroje speciálně určené ke správě úložných zařízení. Připojte kartu SD počítače Raspberry Pi ke svému počítači a po klepnutím na ikonu na ploše spusťte nástroj Partition Editor (editor oddílů – viz obrázek 5.4).

Editor oddílů ve výchozím nastavení vyhledá první jednotku nalezenou v systému, což je obvykle pevný disk počítače PC. Tuto jednotku není vhodné měnit, takže nezapomeňte klepnout na nabídku s výběrem zařízení v pravém horním rohu a zvolit zařízení, které odpovídá kartě SD. U systému s jediným diskem je karta obvykle označena `/dev/sdb` (viz obrázek 5.5).



Obrázek 5.4: Plocha disku Parted Magic



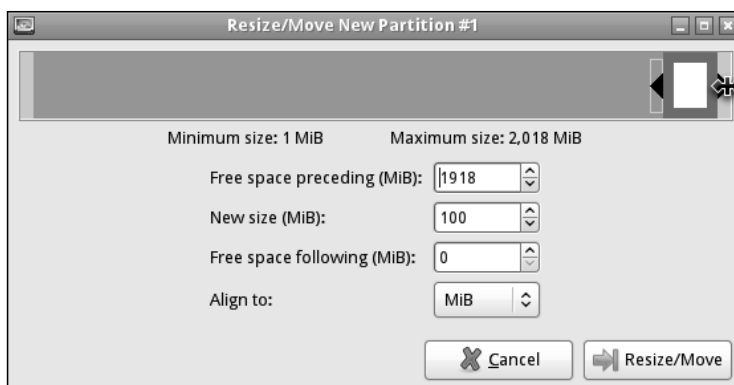
Obrázek 5.5: Nástroj Partition Editor disku Parted Magic před změnou velikosti oddílu



Upozornění: Proces změny velikosti a přesunu oddílů je riskantní. Pokud během změny velikosti kartu SD vyjmete ze čtečky nebo je přerušeno napájení počítače, bude obsah karty poškozen. Před úpravou oddílů vždy vytvořte zálohu všech nenahraditelných osobních souborů, které máte na kartě uloženy.

Konkrétní oddíly, které je nutné upravit a přesunout, závisí na zvolené distribuci. V případě distribuce Debian je potřeba změnit velikost druhého oddílu, což je zpravidla `sdb2`. Po spuštění editoru oddílů postupujte následovně:

1. Některé linuxové distribuce obsahují na konci bitové kopie oddíl `swap` (odkládací oddíl). V editoru oddílů se zobrazuje jako malý oddíl typu `linux-swap`. Pokud tento oddíl najdete, pokračujte dalším krokem postupu. V opačném případě přejděte přímo na krok 5.
2. Klepněte na odkládací oddíl, který bude v seznamu uveden jako poslední, a na panelu nástrojů vyberte možnost `Resize/Move`.
3. Ve zobrazeném dialogovém okně klepněte na rámeček v levém horním rohu a přetáhněte jej do pravého horního rohu (viz obrázek 5.6). Poté by se v poli `Free Space Following (MiB)` měla objevit hodnota 0.



Obrázek 5.6: Přesun odkládacího oddílu v editoru oddílů

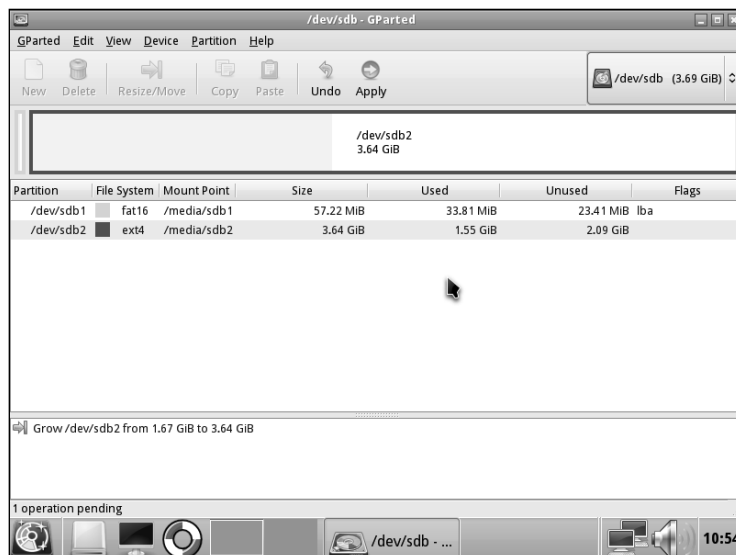
4. Potvrďte změnu klepnutím na tlačítko `Resize/Move`. Při přesunu oddílu se zobrazí upozornění, že nová tabulka oddílů může způsobit problémy při spouštění systému. V případě této změny takové riziko nehrozí, protože nepřesunujete spouštěcí oddíl. Stačí tedy klepnout na tlačítko `OK`.
5. Klepněte na největší oddíl v seznamu, který je obvykle označen `sdb2`, a znovu klepněte na tlačítko `Resize/Move` na panelu nástrojů.
6. Tentokrát klepněte na pravou šipku zbarveného rámečku a přetáhněte ji na pravý okraj šedého rámečku (viz obrázek 5.7). Přitom oddíl pouze nepřesunete, ale zvětšíte jej.
7. Potvrďte změnu klepnutím na tlačítko `Resize/Move` a po zobrazení zprávy s upozorněním znovu klepněte na tlačítko `OK`.



Obrázek 5.7: Změna velikosti kořenového systému souborů v editoru oddílů

8. Klepněte na tlačítko Apply na panelu nástrojů a poté ještě v zobrazeném dialogovém okně. V závislosti na rychlosti karty SD může proces změny velikosti oddílu trvat několik minut.
9. Po dokončení změny velikosti můžete PC vypnout a vložit kartu SD zpět do počítače Pi.

Díky změnám provedeným v editoru oddílů (viz obrázek 5.8) je nyní hlavní oddíl na kartě SD tak velký, jak to umožňuje kapacita karty. Vzhledem k tomu, že do tohoto oddílu většina linuxových distribucí ukládá jak vlastní soubory, tak uživatelská data, měl by mít nyní počítač Pi k dispozici dostatek volného místa.



Obrázek 5.8: Nástroj Partition Editor disku Parted Magic po změně velikosti oddílu

Přechod na větší kartu SD

Jestliže budete počítač Pi nějakou dobu používat, můžete zcela zaplnit kartu SD velikosti 4 GB, i když jste se původně domnívali, že pro vaše potřeby bude určitě stačit. Nové karty SD s kapacitou 8 GB, 16 GB nebo více jsou sice poměrně levné, ale při změně karty byste neradi přišli o své soubory.

Naštěstí lze obsah existující karty SD poměrně snadno přenést na větší kartu. Druhá polovina procesu se neliší od postupu nahrávání systému na kartu SD z kapitoly 1, „Seznámení s počítačem Raspberry Pi“. Rozdíl však spočívá v tom, že místo staženého souboru bitové kopie budete jako zdroj používat existující kartu SD.



Upozornění: Proces *klonování* karty SD je nedestruktivní a po jeho dokončení budou na obou kartách uložena stejná data. Pokud máte na staré kartě SD osobní soubory, nezapomeňte je bezpečně odstranit dříve, než kartu poskytnete jiné osobě. Kromě původní karty SD a novější větší karty budete potřebovat přístup k počítači PC nebo Mac, čtečku karet SD a – jste-li uživatelem systému Windows – také disk Parted Magic, který také umožňuje měnit velikosti oddílů (viz předchozí část kapitoly).

První krok přechodu na větší kartu SD spočívá ve vytvoření bitové kopie stávající karty. Máte-li přístup ke dvěma čtečkám karet SD, můžete tento krok přeskočit a v pozdějších pokynech nahradit název bitové kopie adresou zařízení čtečky karet SD, kde je vložena původní karta z počítače Pi.

Vytvoření bitové kopie v Linuxu

Vytvoření bitové kopie karty SD v systému Linux není o nic těžší než nahrání systému na kartu, které jsme popsali v kapitole 1. Než začnete, zkontrolujte, zda je v počítači dost místa na uložení souboru ve velikosti karty SD. Potom postupujte takto:

1. Z nabídky aplikací své distribuce spusťte terminál.
2. Vložte menší kartu SD z počítače Pi do čtečky karet připojené k PC.
3. Zobrazte seznam disků příkazem `sudo fdisk -l`. Vyhledejte kartu SD podle její velikosti a poznamenejte si adresu zařízení (`/dev/sdX`, kde `X` je písmeno odpovídající příslušnému zařízení). U některých počítačů s integrovanými čtečkami karet SD může být uvedeno `/dev/mmcblkX`, kde `X` je písmeno odpovídající příslušnému zařízení. V takovém případě použijte adresu v následujících pokynech.
4. Příkazem `sudo dd of=dočasná_bitová_kopie.img if=/dev/sdX bs=2M` načtete obsah karty SD a zapíšete jej do souboru s názvem `dočasná_bitová_kopie.img` (viz obrázek 5.9).

```

blacklaw@xerxes-linux: ~
File Edit View Terminal Help
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x24282427

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1  *           1         12748    102398278+   7   HPFS/NTFS
/dev/sda2             12749        25496    102398310   5   Extended
/dev/sda3             25497        77825    420332692+   7   HPFS/NTFS
/dev/sda5             12749        12997      2000061   82  Linux swap / Solaris
/dev/sda6             12998        25496    100398186   83   Linux

Disk /dev/sdb: 3965 MB, 3965190144 bytes
4 heads, 32 sectors/track, 60504 cylinders
Units = cylinders of 128 * 512 = 65536 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00062d20

   Device Boot      Start         End      Blocks   Id  System
/dev/sdb1             17          1216       76800    c   W95 FAT32 (LBA)
/dev/sdb2             1233        57392    3594240    83   Linux
/dev/sdb3             57393        60448     195584    82  Linux swap / Solaris
blacklaw@xerxes-linux:~$ sudo dd of=temporaryimage.img if=/dev/sdb bs=2M

```

Obrázek 5.9: Vytvoření bitové kopie existující karty SD pomocí nástroje dd

Vytvoření bitové kopie v systému OS X

V systému OS X se při vytvoření bitové kopie karty SD z počítače Pi postupuje téměř stejně jako při nahrávání systému na kartu SD, které jsme ukázali v kapitole 1. I v tomto případě zkontrolujte, zda je k dispozici dost místa na uložení souboru ve velikosti karty SD. Potom postupujte takto:

1. Z nabídky Application (Aplikace) vyberte příkaz Utilities (Nástroje) a klepněte na aplikaci Terminal (Terminál).
2. Vložte menší kartu SD z počítače Pi do čtečky karet připojené k PC.
3. Zobrazte seznam úložných zařízení příkazem `diskutil list`. Vyhledejte kartu SD podle její velikosti a poznamenejte si adresu zařízení (`/dev/diskX`, kde `X` je písmeno odpovídající příslušnému zařízení).
4. Pokud byla karta SD automaticky připojena (příkazem `mount`) a zobrazena na ploše, nejprve ji odpojte příkazem `diskutil unmountdisk /dev/diskX`.
5. Příkazem `dd of=dočasná_bitová_kopie.img if=/dev/diskX bs=2M` načtete obsah karty SD a zapíšete jej do souboru s názvem `dočasná_bitová_kopie.img`.

Vytvoření bitové kopie v systému Windows

Nástroj Win32 Disk Imager, pomocí nějž jste nahráli systém na kartu SD v kapitole 1, „Seznámení s počítačem Raspberry Pi“, neumožňuje vytvářet bitové kopie. Místo toho budete potřebovat disk Parted Magic, který poskytne přístup k nástroji dd systému Linux. Postupujte takto:

1. Vložte do svého PC disk Parted Magic, restartujte počítač a zvolte možnost Standard Settings.
2. Pomocí třetí ikony zleva na dolním panelu nástrojů, která vypadá jako počítačový monitor, otevřete okno terminálu (viz obrázek 5.10).



Obrázek 5.10: Ikona terminálu v systému Parted Magic

3. Příkazem `fdisk -l` zobrazíte seznam jednotek v počítači PC a podle velikosti vyhledejte hlavní pevný disk. Poznamenejte si název zařízení: `/dev/sdXN`, kde `X` je písmeno jednotky a `N` znamená číslo oddílu. U některých počítačů s integrovanými čtečkami karet SD může být uvedeno `/dev/mmcblkX`, kde `X` je písmeno odpovídající příslušnému zařízení. V takovém případě použijte adresu v následujících pokynech.
4. Pomocí příkazu `mkdir /media/pevny_disk` vytvořte na pevném disku svého PC přípojný bod. Chcete-li poté jednotku zpřístupnit, připojte ji příkazem `mount /dev/sdXN /media/pevny_disk -o=rw`.
5. Vložte do čtečky karet SD v počítači PC menší kartu SD z počítače Pi a poté zadáním příkazu `fdisk -l` vyhledejte jeho uzel zařízení (`/dev/sdY`, kde `Y` je písmeno jednotky).

6. Příkazem `dd of=/media/pevny_disk/dočasná_bitová_kopie.img if=/dev/sdY bs=2M` načtete obsah karty SD a zapíšete jej do souboru s názvem `dočasná_bitová_kopie.img` na pevném disku.

Vytvořenou bitovou kopii disku můžete zapsat na novou kartu pomocí pokynů k nahrání systému na kartu SD z kapitoly 1, „Seznámení s počítačem Raspberry Pi“. Nezapomeňte, že zápis nějakou dobu trvá, takže buďte trpěliví a nechejte operaci řádně dokončit.

Po dokončení zápisu bitové kopie budete mít dvě karty SD, které budou obsahovat přesně stejná data včetně identické tabulky oddílů. To znamená, že ačkoli může mít nová karta kapacitu 16 GB či 32 GB, počítač Pi dokáže přistupovat pouze ke stejným 2 GB nebo 4 GB jako na původní kartě.

Chcete-li, aby počítač Pi mohl využít veškeré dostupné místo na nové kartě, postupujte podle pokynů v sekci „Změna velikosti stávajících oddílů“ v předchozí části této kapitoly.

Konfigurace počítače Raspberry Pi

Vzhledem ke svému původu v oblasti integrovaných počítačů nemá čip BCM2835, který je jádrem počítače Raspberry Pi, nic podobného nabídce BIOS počítačů PC, kde by bylo možné konfigurovat různá nízkourovňová systémová nastavení. Místo toho se spoléhá na textové soubory s konfiguračními řetězci, které načítá při zapnutí napájení.

Než se začnete seznamovat s různými možnostmi, které jsou k dispozici v souborech `config.txt`, `cmdline.txt` a `start.elf`, věnujte pozornost následujícímu upozornění: změny výchozích hodnot některých těchto nastavení mohou vést k tomu, že počítač Pi se v nejlepším případě nespustí, dokud neobnovíte původní verzi souborů, nebo dokonce může dojít k fyzickému poškození systému. Taková potenciálně nebezpečná nastavení budou v této kapitole označena.

Nastavení hardwaru – soubor `config.txt`

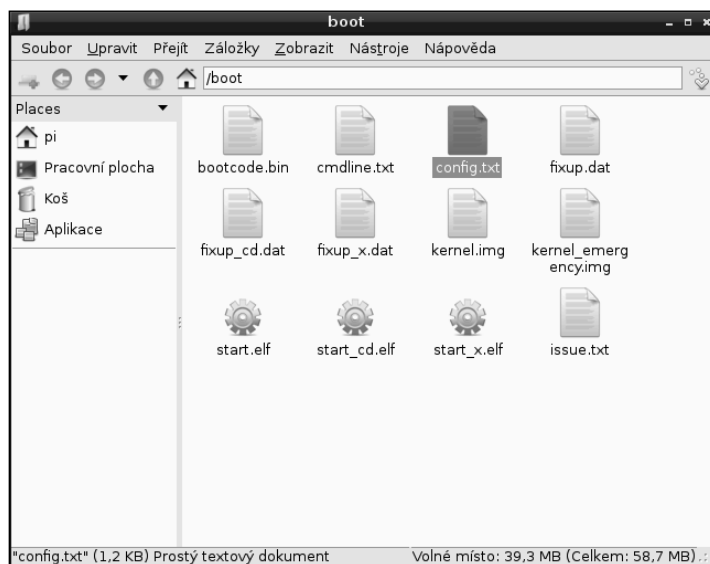
Hardwarová nastavení počítače Pi závisí na hodnotách v souboru `config.txt`, který je umístěn v adresáři `/boot` (viz obrázek 6.1). Na základě tohoto souboru počítač Pi nastavuje různé vstupy a výstupy a taktování čipu BCM2835 a připojeného paměťového modulu.

Máte-li problémy s grafickým výstupem (pokud například obraz nevyplňuje celou obrazovku nebo přesahuje její okraj), můžete je vyřešit změnou parametrů v souboru `config.txt`. Soubor je obvykle prázdný nebo v některých distribucích vůbec neexistuje. To pouze znamená, že počítač Pi funguje s předem nastavenými výchozími hodnotami. Jestliže chcete provést změny a soubor nenajdete, stačí vytvořit nový textový soubor s názvem `config.txt` a vyplnit nastavení, která chcete změnit.

Pomocí souboru `config.txt` lze kontrolovat téměř všechny aspekty hardwaru počítače Pi s výjimkou přidělování paměti hlavním procesorem (CPU) a grafickou částí procesoru BCM2835 (GPU). Postup, jak změnit rozdělení paměti, naleznete v části „Rozdělení paměti – soubor `start.elf`“ dále v této kapitole.

Soubor `config.txt` se načítá pouze při prvním spuštění systému. Případné změny provedené během činnosti počítače Pi se projeví teprve po jeho restartu nebo vypnutí a opakovaném za-

pnutí. Pokud některé změny způsobují problémy, stačí k obnovení výchozích hodnot pouze odstranit soubor z adresáře /boot. Jestliže počítač Pi nelze s novými nastaveními spustit, jednoduše vyjměte kartu SD a v počítači PC odstraňte soubor `config.txt` ze spouštěcího oddílu. Potom kartu znovu vložte do počítače Pi a opakujte akci.



Obrázek 6.1: Obsah adresáře /boot se zvýrazněným souborem `config.txt`

Úpravy zobrazení

Počítač Raspberry Pi obvykle detekuje typ připojeného displeje a podle toho přizpůsobí svá nastavení. Tato automatická detekce však někdy nefunguje. K takovým situacím dochází často tehdy, je-li počítač Raspberry Pi z jedné země připojen k staršímu televizoru z jiné země. Pokud jste svůj počítač Pi připojili k televizoru a obraz se neobjevuje, může být nutné tato výchozí nastavení změnit.

Při zlepšování nebo úpravách výstupu videa lze použít různá nastavení v souboru `config.txt`. Tato nastavení spolu se svými možnými hodnotami jsou popsána v následujícím seznamu.



Upozornění: Při ručních úpravách nastavení výstupu HDMI nebo kompozitního videa se může stát, že počítač Pi nedokáže komunikovat s monitorem. S výjimkou situací, kdy se nic nezobrazuje, je obvykle nejlepší použít automaticky detekovaná nastavení.

- **overscan_left** – toto nastavení posune obraz dovnitř o daný počet pixelů, aby se kompenzovala hodnota *overscan* televizoru. Jestliže textový výstup počítače Pi není na jednom okraji obrazovky úplný, můžete tento problém vyřešit úpravou nastavení `overscan`. Hodnoty je nutné uvádět jako počet vynechaných pixelů.

- **overscan_right** – toto nastavení má stejnou funkci jako nastavení **overscan_left**, ale projevuje se na pravém okraji obrazovky.
- **overscan_top** – uvedené nastavení opět ignoruje určitý počet pixelů, ale v tomto případě na horním okraji obrazovky.
- **overscan_bottom** – pomocí tohoto nastavení lze přeskočit daný počet pixelů od dolního okraje obrazovky. Hodnoty všech nastavení **overscan_** bývají obvykle stejné, takže se kolem displeje vytváří pravidelný okraj.
- **disable_overscan** – pokud připojíte monitor nebo televizor přes rozhraní HDMI, může se kolem obrazu objevit černý okraj. Chcete-li se tohoto okraje zbavit, můžete vypnout všechna výchozí nastavení **overscan**, když nastavíte tento parametr na hodnotu 1.
- **framebuffer_width** – tato hodnota se měří v pixelech a její úprava změní šířku konzoly. Jestliže je text zobrazený na obrazovce příliš malý, pokuste se nastavit menší hodnotu tohoto parametru, než je výchozí šířka připojeného displeje.
- **framebuffer_height** – toto nastavení ovlivňuje velikost konzoly stejným způsobem jako nastavení **framebuffer_width**, ale místo vodorovného rozměru se projevuje svisle.
- **framebuffer_depth** – řídí barevnou hloubku konzoly v bitech na pixel. Výchozí hodnota 16 bitů na pixel poskytuje 65 536 barev. Jiné hodnoty, včetně 8 bitů na pixel (256 barev), 24 bitů na pixel (přibližně 16,7 milionu barev) a 32 bitů na pixel (asi 1 miliarda barev), jsou sice platné, ale mohou způsobit chybný grafický výstup.
- **framebuffer_ignore_alpha** – tento parametr nastavený na hodnotu 1 vypne *alfa kanál*, který řídí průhlednost v konzole. Obvykle není nutné alfa kanál zakazovat, ale někdy lze tímto způsobem napravit poškozené zobrazení, které je způsobeno nastavením parametru **framebuffer_depth** na 32 bitů na pixel.
- **sdtv_mode** – tato hodnota ovlivní analogový kompozitní výstup videa z počítače Pi, aby jej bylo možné použít v různých zemích. Počítač Pi ve výchozím nastavení pracuje se severoamerickou verzí standardu videa NTSC. Uživatelé v jiných zemích, kteří chtějí připojit svůj analogový televizor, musí tuto hodnotu v některých případech změnit. Dostupné jsou tyto hodnoty:
 - 0 – NTSC, standard videa v severní Americe
 - 1 – NTSC-J, japonský standard videa
 - 2 – PAL, standard videa v Británii a dalších zemích včetně ČR
 - 3 – PAL-M, brazilský standard videa
- **sdtv_aspect** – řídí poměr stran analogového kompozitního výstupu. Pokud obraz vypadá roztažený nebo zdeformovaný, upravte tuto hodnotu tak, aby odpovídala poměru stran použitého televizoru. Dostupné jsou tyto hodnoty:
 - 1 – poměr stran 4:3, typický pro starší přístroje
 - 2 – poměr stran 14:9, který je běžný u menších širokoúhlých televizorů
 - 3 – poměr stran 16:9, obvyklý u moderních širokoúhlých televizorů
- **hdmi_mode** – kromě nastavení režimu videa pro analogový kompozitní výstup je také možné přepsat automatickou detekci rozlišení u portu HDMI. Tato možnost je praktická,

chcete-li nastavit menší rozlišení počítače Pi, než je nativní rozlišení displeje, aby byly zobrazované položky viditelné z větší vzdálenosti. Možné hodnoty tohoto nastavení jsou uvedeny v příloze B, „Režimy zobrazení HDMI“.

- **hdmi_drive** – lze upravit i napěťový výstup portu HDMI. To je důležité v případech, kdy používáte adaptér z HDMI na DVI, protože napětí těchto rozhraní se mírně liší. Jestliže se v obraze objevuje „sněžení“ nebo je příliš jasný, zkuste nastavit tento parametr. Dostupné jsou tyto hodnoty:
 - **1** – výstupní napětí DVI. V tomto režimu se po kabelu HDMI nepřenáší žádný zvuk.
 - **2** – výstupní napětí HDMI. V tomto režimu kabel HDMI přenáší i zvuk.
- **hdmi_force_hotplug** – vynutí, aby počítač Raspberry Pi použil port HDMI, i když nedetekuje žádný připojený displej. Hodnota 0 umožní, aby se počítač Pi pokusil detekovat displej, zatímco hodnota 1 vynutí použití portu HDMI v každém případě.
- **hdmi_group** – nastaví režim skupiny HDMI na CEA nebo DMT. Podle typu displeje, který se pokoušíte připojit, byste nejdříve měli změnit toto nastavení. Poté můžete pomocí parametru `hdmi_mode` určit rozlišení a frekvenci výstupu. K dispozici jsou dvě hodnoty:
 - **1** – nastaví skupinu HDMI podle definice, kterou vytvořila asociace CEA (Consumer Electronics Association of America). Toto nastavení použijte v případech, kdy je počítač Pi připojen k televizoru s vysokým rozlišením (HDTV) pomocí rozhraní HDMI. Přitom zvolte nastavení z prvního seznamu v příloze B, „Režimy zobrazení HDMI“.
 - **2** – nastaví skupinu HDMI podle definice, kterou uvádí asociace VESA (Video Electronics Standards Association) ve specifikaci DMT (Display Monitor Timings). Toto nastavení použijte v případech, kdy je počítač Pi připojen k počítačovému monitoru kabelem s rozhraním DVI. Přitom zvolte nastavení z druhého seznamu v příloze B, „Režimy zobrazení HDMI“.
- **hdmi_safe** – vynutí, aby počítač Pi pracoval s předem nastavenou sadou nastavení HDMI, která mají poskytnout maximální kompatibilitu s displeji připojenými k portu HDMI. Nastavení tohoto parametru na hodnotu 1 odpovídá následujícímu nastavení: `hdmi_force_hotplug=1, config_hdmi_boost=4, hdmi_group=1, hdmi_mode=1 a disable_overscan=0`.
- **config_hdmi_boost** – některé monitory vyžadují ke správnému fungování vyšší výkon výstupu HDMI. Pokud se v obraze objevuje „sněžení“, pokuste se tuto hodnotu postupně zvyšovat od 1 (u krátkých kabelů) do 7 (v případě dlouhých kabelů).

Každý parametr v souboru `config.txt` by měl být uveden na samostatném řádku. Za názvem možnosti přitom následuje symbol rovná se (=) a poté požadovaná hodnota. Chcete-li například nastavit, aby počítač Pi použil analogový televizor standardu PAL s poměrem stran 4:3 a hodnotou overscan 20 pixelů na všech stranách, zadejte do souboru `config.txt` následující řádky:

```
sdtv_mode=2
sdtv_aspect=1
overscan_left=20
overscan_right=20
overscan_top=20
overscan_bottom=20
```

Jestliže chcete, aby počítač Pi odesílal výstup na displej DVI pomocí portu HDMI ve formátu 720p60 bez jakéhokoli nastavení overscan, použijte místo toho tyto hodnoty:

```
hdmi_group=1
hdmi_mode=4
hdmi_drive=1
disable_overscan=1
```

Změny se projeví teprve po restartu počítače Pi. Pokud se v důsledku provedených změn přestane na monitoru zobrazovat grafický výstup počítače Pi, stačí vložit kartu SD do jiného počítače a buď nastavení v souboru `config.txt` opravit, nebo soubor úplně odstranit a obnovit tím výchozí hodnoty.

Možnosti spouštění

Pomocí souboru `config.txt` lze také řídit způsob zavádění systému Linux v počítači Raspberry Pi. Ke kontrole načítání jádra systému Linux sice nejčastěji slouží samostatný soubor s názvem `cmdline.txt` (kterým se budeme zabývat v další části kapitoly), ale stačí použít jen soubor `config.txt`. Proces spouštění ovlivňují následující možnosti:

- **disable_commandline_tags** – tato možnost nastaví modul `start.elf` (viz popis v další části kapitoly, aby před načítáním jádra systému Linux přeskočil vyplňování paměťových umístění za hodnotou `0x100`). Tuto možnost není vhodné vypínat, protože to může způsobit nesprávné načtení systému Linux a jeho selhání.
- **cmdline** – parametry příkazového řádku, které budou předány jádru systému Linux. Tato možnost dokáže nahradit soubor `cmdline.txt`, který se obvykle nachází v adresáři `/boot`.
- **kernel** – název načítaného souboru jádra. Tímto způsobem lze načíst nouzové jádro (viz kapitola 3, „Řešení potíží“).
- **ramfsfile** – název počátečního systému souborů RAM (RAMFS – RAM file system), který bude načten. Tuto hodnotu zpravidla nemusíte měnit, pokud nevytváříte nový počáteční systém souborů, se kterým byste chtěli experimentovat.
- **init_uart_baud** – rychlost sériové konzoly v bitech za sekundu. Výchozí hodnota je 115200, ale nižší hodnoty mohou usnadnit připojení, pokud se počítač Pi používá se starším hardwarovým sériovým terminálem.

Přetaktování počítače Raspberry Pi

Kromě toho, že soubor `config.txt` definuje grafický výstup procesoru BCM2835 počítače Pi, umožňuje také manipulovat s čipem jinými způsoby. Konkrétně dovoluje změnit rychlost fungování čipu a zvýšit jeho výkon na úkor životnosti součástky. Tento proces se označuje jako *přetaktování* (overclocking).

Multimediální procesor BCM2835, který je jádrem počítače Pi, je navržen jako procesor typu SoC (system-on-chip) a skládá se ze dvou hlavních částí: grafického procesoru (GPU) a hlavní-

ho procesoru (CPU). Zjednodušeně řečeno hlavní procesor zajišťuje všechny běžné výpočetní operace, zatímco grafický procesor se stará o vykreslování na obrazovce jak v 2D, tak ve 3D.



Upozornění: Změna kteréhokoli nastavení uvedeného v této části může způsobit poškození počítače Pi. Konkrétně změna nastavení, která souvisí s napětím paměti, grafického nebo hlavního procesoru, přepne určitou pojistku v čipu. Tím je zrušena záruka na počítač Raspberry Pi, i když vrátíte zpět normální hodnotu nastavení dříve, než dojde k jakémukoli poškození. Za poškození způsobené použitím těchto nastavení nenese odpovědnost ani nadace Raspberry Pi Foundation, ani prodejce, od něhož jste počítač Pi zakoupili. Pokud si nejste jisti, *tato nastavení neměňte*: zvýšení výkonu dosažené díky přetaktování obvykle nestojí za riziko poškození počítače Pi.

Pomocí souboru `config.txt` lze přetaktovat jednu nebo obě části čipu BCM2835. Je také možné zvýšit rychlost fungování paměťového modulu, který je nasazen na čipu v montážním formátu PoP (package-on-package).

Zvýšení provozní frekvence těchto komponent zajistí mírné zvýšení výkonu počítače Pi: vyšší taktovací frekvence grafického procesoru znamená, že se bude rychleji vykreslovat 3D grafika (například prvky hry) a rychleji bude probíhat i dekódování videa, což zajistí plynulejší přehrávání. Nárůst taktovací frekvence hlavního procesoru stejně jako zvýšení frekvence paměti pak posílí celkový výkon zařízení.

Důvod, proč počítač Pi nemá nastaveny vyšší provozní rychlosti již z výroby, spočívá zejména v životnosti čipů. Společnost Broadcom, která čip BCM2835 vyrábí, stanovila jeho provozní frekvenci na 700 MHz. Po zvýšení této oficiálně určené rychlosti sice čip může fungovat, ale tato změna se rovněž negativně projeví na jeho životnosti. Čipy konstrukce SoC – na rozdíl od procesorů stolních počítačů – obvykle k přetaktování neposkytují příliš velký prostor.

Nastavení přetaktování

Pokud jste ochotni přijmout riziko, že kvůli malému výkonnostnímu zisku počítač Pi zničíte (tento jev se v oboru integrovaných zařízení označuje jako *bricking*), můžete využít některá nastavení souboru `config.txt`. Výkon procesoru SoC počítače Pi závisí na následujících nastaveních:

- **arm_freq** – nastaví základní taktovací frekvenci části hlavního procesoru čipu BCM2835, což zvýší obecný výpočetní výkon. Výchozí frekvence je 700 MHz.
- **gpu_freq** – nastaví taktovací frekvenci grafické části čipu BCM2835. Tato změna urychlí všechny operace související s grafikou. Výchozí frekvence je 250 MHz. Kromě toho můžete nastavit jednotlivé části hardwaru grafického procesoru pomocí těchto možností:
 - **core_freq** – nastaví základní taktovací frekvenci grafického procesoru a ostatní frekvence ponechá beze změny, aby se zvýšil celkový výkon grafických operací. Výchozí frekvence je 250 MHz.
 - **h264_freq** – nastaví taktovací frekvenci hardwarového dekodéru videa grafického procesoru, aby se zlepšilo přehrávání dat videa v kódování H.264. Výchozí frekvence je 250 MHz.

- **isp_freq** – nastaví taktovací frekvenci kanálu obrazového senzoru, což zvýší rychlost zachytávání připojeného hardwaru videa (např. kamery). Výchozí frekvence je 250 MHz.
- **v3d_freq** – nastaví taktovací frekvenci hardwaru 3D renderování grafického procesoru. Díky tomu se zlepší výkon při vizualizaci a hrách. Výchozí frekvence je 250 MHz.
- **sdram_freq** – nastaví taktovací frekvenci čipu paměti RAM (random access memory) počítače Pi, což mírně zlepší výkon celého systému. Výchozí frekvence je 400 MHz.
- **init_uart_clock** – nastaví výchozí taktovací frekvenci *modulu UART (Universal Asynchronous Receiver/Transmitter)*, který řídí sériovou konzolu. Výchozí hodnota 3000000 nastaví frekvenci 3 MHz. Změna tohoto nastavení se pravděpodobně projeví pouze poškozením výstupu sériové konzoly.
- **init_emmc_clock** – nastaví výchozí taktovací frekvenci řadiče karet SD. Výchozí hodnota 80000000 nastaví frekvenci 80 MHz. Zvýšení této hodnoty může urychlit čtení a zápis karty SD, ale může také způsobit poškození dat.

Chcete-li například přetaktovat hlavní procesor na 800 MHz, grafický procesor na 280 MHz a paměť RAM na 420 MHz, zadejte do souboru `config.txt` následující možnosti (každou na samostatný řádek):

```
arm_freq=800
gpu_freq=280
sdram_freq=420
```

Stejně jako v případě úprav konfigurace zobrazení platí, že všechny provedené změny týkající se přetaktování se projeví teprve po restartu počítače Pi. Pokud chcete vrátit normální nastavení, můžete buď odstranit celý soubor `config.txt`, nebo – jestliže jste pomocí něj změnili i nastavení zobrazení – odstranit pouze řádky, které se týkají přetaktování. Poté restartujte počítač Pi.

Nastavení přepětí

Pokud svůj počítač Pi zkoušíte přetaktovat, nakonec narazíte na bariéru, kterou zařízení nedokáže překonat. Přesný bod, za kterým se počítač Pi nepodaří spolehlivě přetaktovat, závisí na konkrétním kusu. Je to způsobeno přirozenými odchylkami v čipu, které vznikají během výroby. U některých uživatelů může být limit pouhých 800 MHz. Jiní zjistí, že mohou ze svého počítače Pi bez problémů vymáčkout až 1 GHz (1 000 MHz).

Chcete-li, aby počítač Pi podával o trochu vyšší výkon, existuje způsob, jak tento horní limit potenciálně posunout: procesem, který se označuje jako *přepětí* (overvoltage). Procesor BCM2835 typu SoC počítače Pi a připojený paměťový modul obvykle fungují s napětím 1,2 V. I když to nelze doporučit, toto výchozí nastavení je možné změnit a vynutit, aby komponenty pracovaly s vyšším nebo nižším napětím. Zvýšení napětí posílí úroveň signálu v čipu, který tak spíše dokáže pracovat s vyššími taktovacími frekvencemi. Znamená to také, že čip se bude více zahřívat a v porovnání se samotným přetaktováním se dramaticky zkrátí jeho životnost.

Na rozdíl od dříve popsanych nastavení, která jsou v souboru `config.txt` uvedena jako absolutní hodnoty, se napětí upravuje pomocí hodnot vztažených ke standardnímu nastavení 1,2 V počítače Pi. Každé celé číslo větší než nula znamená zvýšení napětí o 0,025 V oproti

standardnímu napětí. Každé celé číslo menší než nula označuje pokles napětí o 0,025 V oproti standardnímu napětí.



Upozornění: Nastavením libovolné možnosti změny napětí v souboru `config.txt` dojde v čipu BCM2835 k aktivaci speciální pojistky, kterou nelze resetovat. Jedná se o jednoduchý způsob, jak zjistit, zda se někdo nepokoušel přetaktovat čip mimo jeho jmenovitou specifikaci. Přitom dojde ke zrušení záruky i v případech, že důvod selhání s přetaktováním nesouvisí. Pokud během záruky požádáte o výměnu počítače Pi a pojistka přitom bude aktivovaná, *nový počítač nedostanete*. Nezkoušejte nastavit přepětí počítače Pi, pokud si nemůžete dovolit nákup nového kusu.

Nastavení na úpravu napětí mají horní a dolní limity 8 a -16, které odpovídají 0,2 V nad standardním napětím (v absolutní hodnotě 1,4 V) a 0,4 V pod standardním napětím (tj. absolutně 0,8 V). Napětí je nutné měnit v celých číslech a nelze nastavit nižší hodnotu než 0,8 V (-16) nebo vyšší než 1,4 V (8).

Soubor `config.txt` poskytuje následující nastavení:

- **over_voltage** – upraví standardní napětí procesoru BCM2835. Hodnoty se uvádějí jako celá čísla, která odpovídají nárůstu nebo poklesu napětí o 0,025 V oproti výchozí hodnotě (0), přičemž platí dolní limit -16 a horní limit 8.
- **over_voltage_sdram** – upraví napětí dodávané paměťovému čipu počítače Pi. Stejně jako u nastavení `over_voltage` se hodnoty udávají jako celá čísla, která reprezentují zvýšení či snížení o 0,025 V oproti základnímu napětí (0). Lze nastavit minimální hodnotu -16 a maximální hodnotu 8. Navíc lze pomocí následujících možností upravit napětí jednotlivých komponent paměti:
 - **over_voltage_sdram_c** – změni napětí dodávané řadiči paměti. Přípustné hodnoty jsou stejné jako v případě nastavení `over_voltage_sdram`.
 - **over_voltage_sdram_i** – upraví napětí dodávané paměťovému vstupně-výstupnímu (I/O) systému paměti. Přípustné hodnoty jsou stejné jako v případě nastavení `over_voltage_sdram`.
 - **over_voltage_sdram_p** – upraví napětí dodávané komponentám fyzické vrstvy (physical layer – PHY) paměti. Přípustné hodnoty jsou stejné jako v případě nastavení `over_voltage_sdram`.

Uvedme příklad: Následující řádky zadané do souboru `config.txt` zajistí mírné zvýšení napětí čipu BCM2835 o 0,05 V na 1,25 V a poněkud výraznější zvýšení napětí paměťového čipu o 0,1 V na 1,3 V:

```
over_voltage=2
over_voltage_sdram=4
```

Stejně jako u jiných nastavení platí, že smazáním řádků ze souboru `config.txt` nebo odstraněním samotného souboru lze obnovit normální konfiguraci. Na rozdíl od jiných nastavení však zůstane stopa v podobě přepálené pojistky procesoru BCM2835, která ruší záruku na počítač Pi i v případě, kdy jsou obnovena výchozí nastavení.

Vypnutí mezipaměti L2

Procesor BCM2835 typu SoC počítače Pi je vybaven 128 kB *mezipaměti vrstvy 2* (layer 2 cache). Tato paměť je sice malá, ale mimořádně rychlá. *Mezipaměť* umožňuje dočasně ukládat data a instrukce přenášené mezi pomalejší operační pamětí a procesorem, aby se zvýšil výkon počítače.

Vzhledem k původu procesoru BCM2835 jako multimediálního procesoru určeného pro set-top boxy je tato mezipaměť L2 navržena tak, aby k ní přistupovala pouze grafická část čipu. Na rozdíl od klasických procesorů nemá hlavní procesor k dispozici žádnou vlastní paměť L2.

Pomocí souboru `config.txt` lze nastavit, aby čip BCM2835 umožnil přístup k mezipaměti L2 své komponentě hlavního procesoru. V některých případech lze tímto způsobem zvýšit výkon. Jindy může výkon poklesnout, což je způsobeno tím, že mezipaměť se na čipu nachází poměrně daleko od části hlavního procesoru a blíže ke grafickému procesoru.

Použití mezipaměti L2 také předpokládá linuxovou distribuci, která byla kompilována s ohledem na mezipaměť. Povolení mezipaměti L2 v distribuci, která byla vytvořena pro standardní počítač Pi s vypnutou mezipamětí, může vést k podivnému a nežádoucímu chování.

Chcete-li zapnout přístup hlavního procesoru k mezipaměti L2, stačí do souboru `config.txt` přidat následující řádek:

```
disable_l2cache=0
```

Stejně jako u jiných nastavení v souboru `config.txt` je nutné nejdříve restartovat systém, aby se změna projevila. Jestliže chcete přístup hlavního procesoru k mezipaměti vypnout, nahraďte hodnotu 0 hodnotou 1.

Zapnutí testovacího režimu

Tuto poslední možnost v souboru `config.txt` nemusí naprostá většina uživatelů počítače Pi vůbec zkoušet, ale pro úplnost se o ní zmíníme: jedná se o testovací režim. Testovací režim se používá při výrobě počítače Raspberry Pi. V kombinaci se speciálním hardwarem, kterým lze elektricky kontrolovat základní desku, mohou pracovníci ověřovat, zda počítač Pi funguje tak, jak má.



Upozornění: Povolení testovacího režimu nezpůsobí žádné trvalé škody, ale počítač Pi spustí operační systém teprve poté, kdy režim znovu zakázete a vypnete a opět zapnete napájení počítače.

Jestliže vás zajímá, jak se počítač Pi chová v továrně, můžete zapnout testovací režim zadáním následující možnosti do souboru `config.txt`:

```
test_mode=1
```

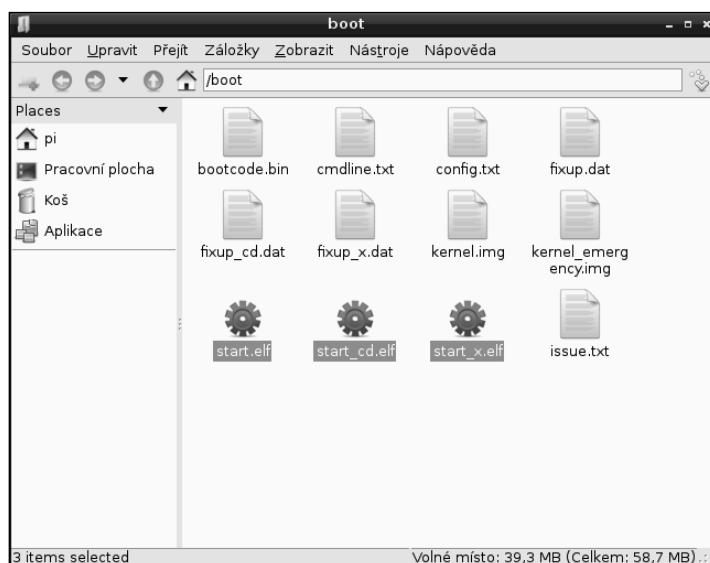
Tak jako u jiných nastavení v souboru `config.txt` bude testovací režim povolen teprve po restartu počítače Pi. Chcete-li testovací režim znovu zakázat, vymažte příslušný řádek v souboru `config.txt` a úplně odstraňte celý soubor nebo nahraďte hodnotu 1 hodnotou 0.

Rozdělení paměti – soubor start.elf

Počítač Raspberry Pi je sice vybaven pouze jediným paměťovým čipem s kapacitou 256 MB, ale tuto paměť lze hardwaru přiřadit různými způsoby. Čip BCM2835 se dělí na dvě hlavní části: univerzální hlavní procesor a grafický procesor orientovaný na grafické operace. Obě uvedené části čipu potřebují při své činnosti pracovat s pamětí. To znamená, že musí sdílet celkových 256 MB paměti počítače Raspberry Pi. Rozdělení paměti závisí na souboru, který se nazývá `start.elf`.

Typické rozdělení volí správci linuxových distribucí, které se instalují do počítače Pi. Někteří z nich se rozhodují pro stejnoměrné rozdělení po 128 MB jak pro hlavní, tak grafický procesor. Tím je zajištěno, že grafický hardware dokáže plně využít svůj potenciál. Autoři jiných distribucí poskytují větší část paměti hlavnímu procesoru, aby bylo možné zvýšit obecný výpočetní výkon.

Většina distribucí obsahuje tři kopie souboru `start.elf` kromě té jeho verze, která je načtena při spuštění počítače Pi: `arm128_start.elf`, `arm192_start.elf` a `arm224_start.elf` (viz obrázek 6.2). Tyto tři soubory jsou identické s výjimkou jedné malé změny: velikosti paměti, která je vyhrazena pro hlavní procesor čipu BCM2835.



Obrázek 6.2: Různé soubory `start.elf` v adresáři `/boot`

První soubor, `arm128_start.elf`, je nakonfigurován tak, aby rozdělil paměť na dvě stejné části, kdy má 128 MB k dispozici jak hlavní procesor ARM čipu BCM2835, tak jeho grafický procesor VideoCore IV.

Druhý a třetí soubor pak postupně zmenšují část paměti určenou pro grafický procesor. Soubor `arm192_start.elf` poskytuje 192 MB hlavnímu procesoru a 64 MB grafickému procesoru, zatímco při použití souboru `arm224_start.elf` má hlavní procesor k dispozici 224 MB

a grafický procesor pouze 32 MB. Když počítač Pi nastavíte tak, aby pracoval s těmito soubory, můžete zvětšit část paměti určenou pro obecné výpočetní operace.



Upozornění: Aplikace, které provádějí náročné grafické zpracování (například 3D hry a software na přehrávání videa ve vysokém rozlišení) obvykle pro grafický procesor potřebují 128 MB paměti. Pokud tuto kapacitu omezíte, může to způsobit výrazný pokles výkonu.

Většina univerzálních distribucí pracuje s rozdělením 192 MB/64 MB, ale vynutíte-li, aby počítač Pi používal rozdělení 224 MB/32 MB, můžete programům uvolnit další paměť. Tím lze zvýšit výkon v univerzálních aplikacích bez rizik, která souvisejí s přetaktováním počítače Pi.

Chcete-li změnit způsob přidělení paměti, stačí odstranit soubor `start.elf` z adresáře `/boot` a zkopírovat na jeho místo jednu ze tří alternativních verzí. Nezapomeňte nový soubor přejmenovat na `start.elf`, jinak se počítač Pi nespustí. Nejsnáze to zajistíte zadáním následujícího příkazu v okně terminálu:

```
sudo cp /boot/arm224_start.elf /boot/start.elf
```

Když počítač Pi restartujete, bude mít jeho hlavní procesor ARM k dispozici více paměti. Pokud chcete zkontrolovat kapacitu dostupné paměti, zadejte v okně terminálu příkaz `free`. Jestliže chcete přejít na jiné rozdělení paměti, pouze opakujte předchozí příkaz, kde uvedete jinou verzi zdrojového souboru `start.elf`: `arm_192_start.elf` či `arm128_start.elf`.

Nastavení softwaru – soubor `cmdline.txt`

Kromě souboru `config.txt`, který umožňuje nastavovat různé funkce hardwaru počítače Pi, existuje v adresáři `/boot` také jiný důležitý soubor: `cmdline.txt` (viz obrázek 6.3). Tento soubor obsahuje takzvaný *řádek režimu jádra* (kernel mode line), což jsou možnosti předávané jádru systému Linux při spouštění počítače Pi.

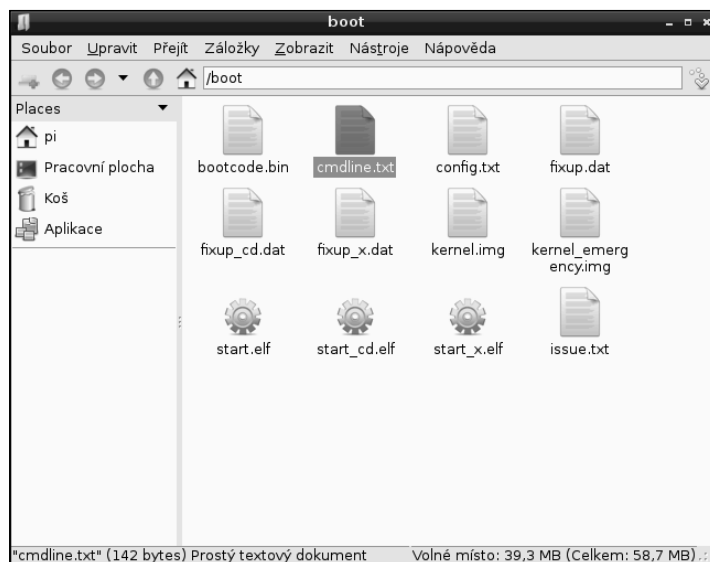
V linuxových distribucích určených pro stolní počítače nebo notebooky se tyto parametry obvykle jádru předávají pomocí nástroje označovaného jako *zavaděč* (bootloader), který má svůj vlastní konfigurační soubor. V počítači Pi se tyto možnosti jednoduše zadávají přímo do souboru `cmdline.txt`, který počítač Pi načítá během svého spouštění.

Do souboru `cmdline.txt` lze zapsat téměř každou možnost jádra, kterou systém Linux podporuje. Lze tak například upravit vzhled konzoly nebo zvolit načítané jádro. Jako příklad uveďme soubor `cmdline.txt` z distribuce Debian. Obsah tohoto souboru by měl být uveden na jednom nerozděleném řádku:

```
dwc_otg.lpm_enable=0 console=ttyAMA0,115200 ↵
kgdboc=ttyAMA0,115200 console=tty1 root=/dev/mmcblk0p2 ↵
rootfstype=ext4 rootwait
```

První možnost `dwc_otg.lpm_enable` zajistí, aby počítač Pi zakázal režim *OTG* (*On-The-Go*) svého řadiče USB. Tím lze předcházet problémům, které by mohly nastat při povolení funkce

bez odpovídající podpory operačního systému. Většina distribucí systému Linux pro počítač Pi tento režim vypíná.



Obrázek 6.3: Soubor cmdline.txt v adresáři /boot

Možnost `console` sděluje systému Linux, že má vytvořit sériovou konzolu (zařízení `ttyAMA0`), a určuje, s jakou rychlostí má fungovat. Ve většině případů je vhodné ponechat výchozí nastavení rychlosti 115 200 kb/s (kilobitů za sekundu). Jestliže počítač Pi komunikuje se staršími zařízeními, lze tuto hodnotu odpovídajícím způsobem snížit.

Možnost jádra `kgdboc` povolí ladění jádra systému Linux prostřednictvím sériové konzoly, která byla vytvořena pomocí parametru `console`. Většina uživatelů tento parametr nepotřebuje. Možnost ladění jádra přes sériové připojení je nejužitečnější pro vývojáře. Většina distribucí ponechává tuto možnost pro všechny případy zapnutou.

Druhá položka `console` vytvoří zařízení `tty1`, což je textová obrazovka, která se zobrazuje při prvním spuštění počítače Pi. Bez této položky byste s počítačem Pi nemohli pracovat, pokud byste nepřipojili nějaké zařízení k sériové konzole vytvořené pomocí prvního parametru `console`.

Možnost `root` informuje jádro systému Linux, kde nalezne svůj *kořenový systém souborů*, který obsahuje všechny soubory a adresáře potřebné k fungování systému. V případě výchozí distribuce Debian leží na druhém oddílu karty SD – zařízení `mmcblk0p2`. Tuto možnost lze upravit tak, aby odkazovala na externí úložné zařízení připojené přes USB. Uvedeným způsobem je možné výrazně urychlit činnost počítače Pi oproti stavu, kdy je jeho kořenový systém souborů uložen na kartě SD.

Kromě umístění svého kořenového systému souborů potřebuje jádro také vědět, s jakým formátem byl oddíl vytvořen. Vzhledem k tomu, že systém Linux je kompatibilní s různými sys-

témy souborů, možnost `rootfstype` konkrétně udává, že distribuce Debian používá systém souborů EXT4.

Z posledního parametru `rootwait` může jádro zjistit, že by se nemělo pokoušet o další spouštění systému, dokud nebude dostupné zařízení, které obsahuje kořenový systém souborů. Bez této možnosti by se mohl počítač Pi zablokovat, pokud by se začal spouštět dříve, než by byla jeho poměrně pomalá karta SD plně připravena k použití.

S výjimkou nastavení `dwc_otg` není žádný z těchto parametrů jádra specifický pro počítač Pi. Konfigurace zavaděče libovolné linuxové distribuce bude obsahovat seznam možností, které budou velmi podobné možnostem v souboru `cmdline.txt`.

Soubor `cmdline.txt` je obvykle lepší neupravovat. Vytvořili jej správci linuxové distribuce právě pro svou konkrétní verzi systému a může se distribuci od distribuce lišit. Položky, které fungují v systému Fedora Remix, nemusí fungovat v distribuci Debian a naopak. Možnosti dostupné v souboru `cmdline.txt` závisí na tom, které jádro distribuce používá a které funkce do něj byly zahrnuty při jeho sestavování.

Pokud pracujete na vývoji jádra, můžete pomocí souboru `cmdline.txt` předávat parametry povolující či zakazující nové funkce, které jste do jádra kompilovali. Tak jako u souboru `config.txt` je pokaždé nutné restartovat počítač, aby se změny projevíly.

POUŽITÍ POČÍTAČE PI JAKO MEDIÁLNÍHO CENTRA, KANCELÁŘSKÉHO POČÍTAČE A WEBOVÉHO SERVERU

V této části:

- **KAPITOLA 7** – Počítač Pi jako domácí kino
- **KAPITOLA 8** – Počítač Pi jako kancelářský počítač
- **KAPITOLA 9** – Počítač Pi jako webový server

Počítač Pi jako domácí kino

Mezi nejoblíbenější funkce počítače Pi patří jeho nasazení jako počítače pro domácí kino (HTPC – home theatre PC). Procesor Broadcom BCM2835 v centru počítače Pi je navržen s ohledem na vysoký multimediální výkon, protože byl původně určen do počítačů typu HTPC.

Grafická část čipu BCM2835 typu SoC (system-on-chip), tj. modul Broadcom VideoCore IV, dokáže v reálném čase přehrávat video s vysokým rozlišením v oblíbeném formátu H.264. Čip rovněž umožňuje přehrávat zvukové soubory v různých formátech, jak přes analogový zvukový výstup s konektorem 3,5 mm, tak digitálně pomocí portu HDMI.

Díky svým malým rozměrům, nízkému odběru energie a tichému provozu je počítač Pi atraktivní pro uživatele, kteří chtějí zprovoznit domácí kino. Od uvedení počítače na trh se objevilo mnoho distribucí a softwarových balíčků, které jej umožňují změnit na uživatelsky přívětivý počítač typu HTPC. Chcete-li však domácí kino vyzkoušet, nemusíte se nutně vzdávat svého stávajícího operačního systému.

Přehrávání hudby v konzole

Pokud vyvíjíte software, pravděpodobně budete většinu času trávit v konzole počítače Pi. Vzhledem k tomu, že většina programů na přehrávání hudby je určena pro grafické prostředí, může být vaše pracovní prostředí příliš tiché – ale také nemusí.

Počítač Pi je kompatibilní s výkonným balíčkem na přehrávání hudby, který funguje v textovém režimu a nazývá se *moc* (což je zkratka slov *music on console*). Na rozdíl od jiných nástrojů (jako je *LXMusic*) lze program *moc* nainstalovat a používat i v případech, kdy není počítač Pi vybaven žádným grafickým uživatelským rozhraním.

Chcete-li program *moc* použít, nejdříve nainstalujte jeho balíček z repozitářů distribuce. V případě distribuce Debian přitom stačí zadat následující příkaz v konzole nebo okně terminálu:

```
sudo apt-get install moc
```



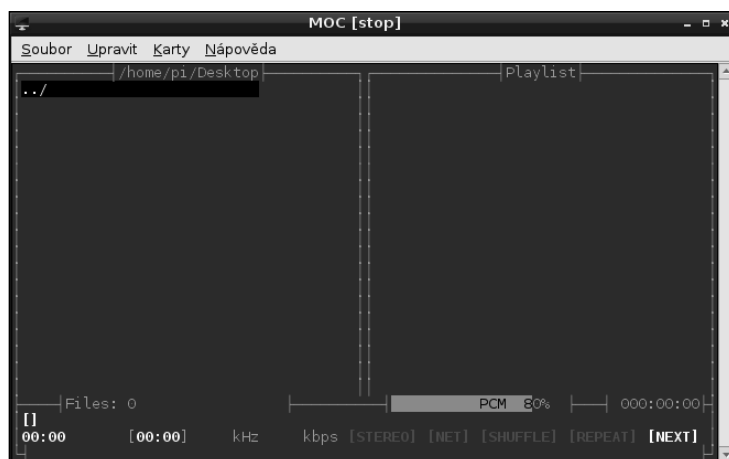
Tip: Některé distribuce již obsahují jiný nástroj s názvem *moc*, který s přehráváním zvuku nijak nesouvisí. Pokud po nainstalování balíčku *moc* nezískáte požadovaný efekt, pokuste se název balíčku opravit na *mocp*.

Jestliže trávíte hodně času mimo grafické uživatelské rozhraní při práci v konzole, představuje program `moc` vynikající volbu k přehrávání hudby. Oproti jiným nástrojům funguje na pozadí. Začnete-li tedy dělat něco jiného, přehrávání hudby tím nepřerušíte.

Chcete-li spustit program `moc`, použijte místo příkazu `moc` příkaz `mocp`. Důvod spočívá v tom, že existuje jiný nástroj s názvem `moc`. Proto byl zvolen jiný název, aby operační systém dokázal oba balíčky rozlišit.

V konzole (nebo okně terminálu, používáte-li prostředí plochy) stačí zadat následující příkaz:
`mocp`

Standardní rozhraní programu `mocp` je rozděleno na dvě podokna (viz obrázek 7.1). Levé podokno slouží jako prohlížeč souborů, který umožňuje vyhledat přehrávané hudební soubory. Seznam lze procházet pomocí kurzorových kláves a klávesa `Enter` spustí přehrávání od aktuálně vybrané skladby. Stisknete-li klávesu `Enter` na názvu adresáře, prohlížeč přejde do daného adresáře, zatímco po stisknutí této klávesy na položce `../` na začátku seznamu přejdete o jednu adresářovou úroveň zpět. Pravé podokno zobrazuje aktuální seznam stop (playlist).



Obrázek 7.1: Standardní rozhraní hudebního přehrávače `mocp` určeného pro konzolu

Síla nástroje `mocp` se projeví tehdy, když aplikaci ukončíte stisknutím klávesy `Q`. Jestliže program `mocp` právě přehrává hudbu, bude přehrávání pokračovat, i když budete v konzole nebo okně terminálu provádět jiné úkoly. Při novém spuštění nástroje `mocp` se obnoví jeho rozhraní, kde můžete změnit skladby nebo pozastavit či úplně zastavit přehrávání. Aplikaci `mocp` lze také řídit přímo z terminálu, aniž by bylo nutné pracovat s jejím rozhraním. Spustíte-li program `mocp` s příznaky (tj. možnostmi, které následují za příkazem a uvádějí se symbolem pomlčky), můžete spouštět, zastavovat, přeskakovat skladby a jinak měnit přehrávání, aniž byste museli přecházet do okna softwaru.

Nejčastěji se používají následující příznaky nástroje `mocp`:

- `-s` – zastaví aktuální přehrávání
- `-G` – pozastaví přehrávání nebo je obnoví, pokud je aktuálně pozastaveno
- `-f` – přeskočí na další skladbu v adresáři nebo seznamu stop
- `-r` – přejde zpět na předchozí skladbu v adresáři nebo seznamu stop
- `-i` – vypíše do okna terminálu nebo na konzolu informace o aktuální skladbě
- `-x` – zastaví přehrávání a zcela ukončí program `mocp`

Další informace o řízení programu `mocp` získáte po zadání příkazu `man mocp`.

Specializovaný počítač typu HTPC díky distribuci Raspbmc

Jedna věc je možnost přehrávat v počítači Raspberry Pi hudbu, ale čip BCM2835 toho dokáže mnohem více. Díky svému grafickému procesoru VideoCore IV může dekodovat a přehrávat video v plném rozlišení 1080p s kódováním H.264. Díky tomu lze počítač Pi navzdory jeho minimálním rozměrům a mimořádně nízkým požadavkům na napájení nasadit jako výkonné multimediální centrum.

Chcete-li však počítač Pi maximálně využít jako počítač typu HTPC, potřebujete k tomu dodatečný software. Tento software je sice možné nainstalovat do distribuce Debian, ale existuje snadnější cesta: přechod na distribuci *Raspbmc*.

Účelem distribuce Raspbmc, kterou vytvořil Sam Nazarko, je zejména změnit počítač Raspberry Pi na plnohodnotný systém multimediálního centra, který bude poskytovat kompletní funkce přehrávání videa a hudby, zobrazování fotografií a streamování z Internetu. Vychází z oblíbené distribuce Xbmc, kterou několik výrobců zařízení zvolilo jako základ svých komerčních systémů set-top boxů.

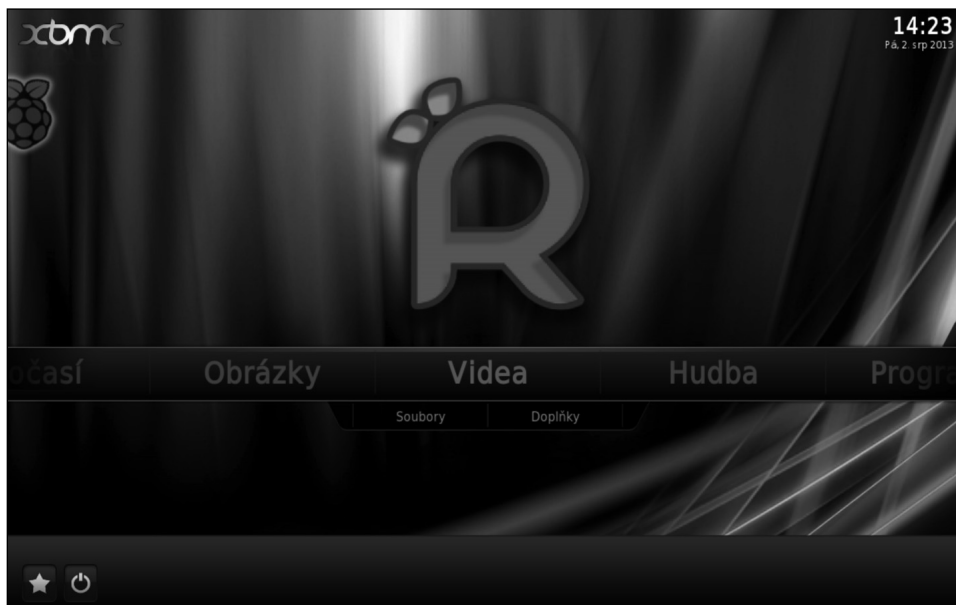
Pokud plánujete, že při tvorbě domácího kina využijete možnosti výstupu videa ve vysokém rozlišení a dekodování formátu H.264, které nabízí počítač Pi, rozhodně se vyplatí sáhnout po distribuci Raspbmc. Její instalace a spuštění v počítači Pi totiž není o nic složitější než v případě jiných linuxových distribucí. Nejdříve si stáhněte instalační program z oficiálního webu: <http://www.raspbmc.com/download/>. Tímto způsobem zprovozníte distribuci Raspbmc nejsnáze. K dispozici jsou instalační programy pro systémy Linux, OS X i Windows, které automaticky stáhnou bitovou kopii distribuce a zapíší ji na kartu SD připojenou k systému. Samozřejmě také můžete ručně stáhnout soubor bitové kopie a postupovat podle pokynů v kapitole 1, „Seznámení s počítačem Raspberry Pi“, ale není to nutné.



Upozornění: Pokud již máte kartu SD, kterou používáte v počítači Pi, mějte na paměti, že instalací distribuce Raspbmc na tuto kartu smažete celý její obsah. Zálohujte nejdříve všechny soubory, které chcete zachovat, případně chcete-li mít možnost přecházet mezi oběma distribucemi, kupte si druhou kartu SD a vyhraďte ji pro instalaci distribuce Raspbmc.

Po dokončení činnosti instalačního programu distribuce Raspbmc vložte kartu SD do počítače Pi a znovu připojte napájecí zdroj. Zkontrolujte však, zda je připojen i ethernetový kabel, protože distribuce Raspbmc potřebuje po svém prvním spuštění stáhnout určitá data z Internetu. První spuštění distribuce Raspbmc může trvat 10 až 15 minut, protože stahuje aktualizace a dělí kartu SD na oddíly. Podruhé se však spustí mnohem rychleji. Nepodléhejte panice, pokud se první spuštění zastaví ve fázi formátování oddílů. Delší prodleva je zde normální.

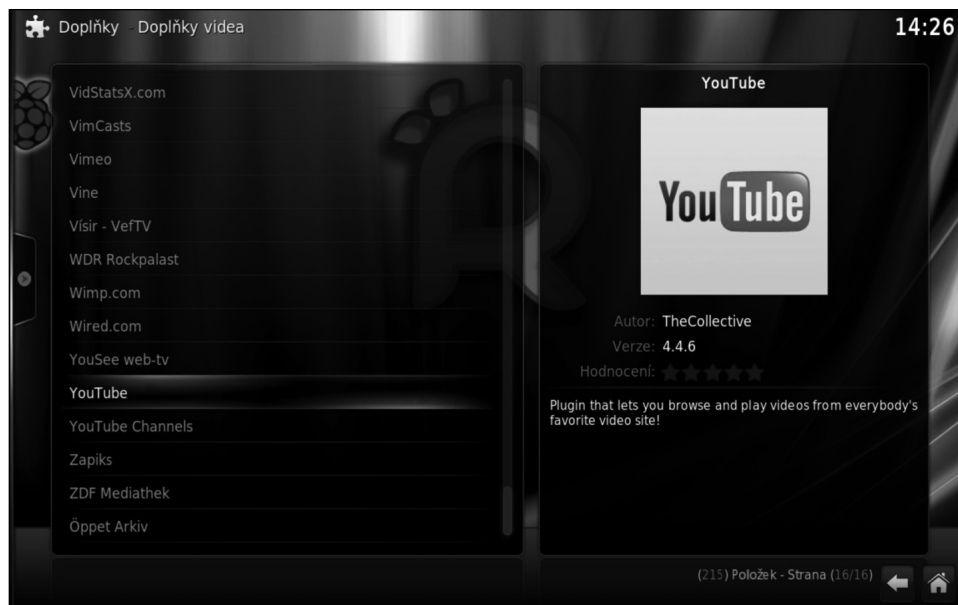
Po svém načtení distribuce Raspbmc automaticky spustí službu Xbmc (viz obrázek 7.2). Tato služba poskytuje přizpůsobené uživatelské rozhraní, které je určeno zejména pro domácí použití. Všechny prvky rozhraní jsou přístupné pomocí klávesnice nebo myši a díky velkému a snadno čitelnému písmu a hierarchickým nabídkám lze snáze vyhledat požadované funkce. Můžete si také zakoupit infračervený dálkový ovladač. Jeho součástí je přijímač, který se zapojuje do portu USB počítače Pi, a vysílač, který zajišťuje pohodlné ovládání domácího kina bez překážejících klávesnic nebo zamotaných kabelů.



Obrázek 7.2: Domovská obrazovka služby Xbmc načtené v distribuci Raspbmc

Streamování internetových médií

Služba Xbmc je ve výchozím nastavení nakonfigurována tak, aby přehrávala pouze soubory nalezené v samotném počítači Raspberry Pi. Pokud v nabídce Video zvolíte možnost Add-ons (Doplňky), můžete možnosti zařízení rozšířit o působivé funkce internetového streamování, včetně přístupu k různým televizním kanálům a službám streamování, které fungují výhradně v Internetu. Po klepnutí na položku Add-ons (Doplňky) zvolte možnost Get More (Získat více), která zpřístupní úplný seznam kompatibilních modulů plug-in (viz obrázek 7.3).

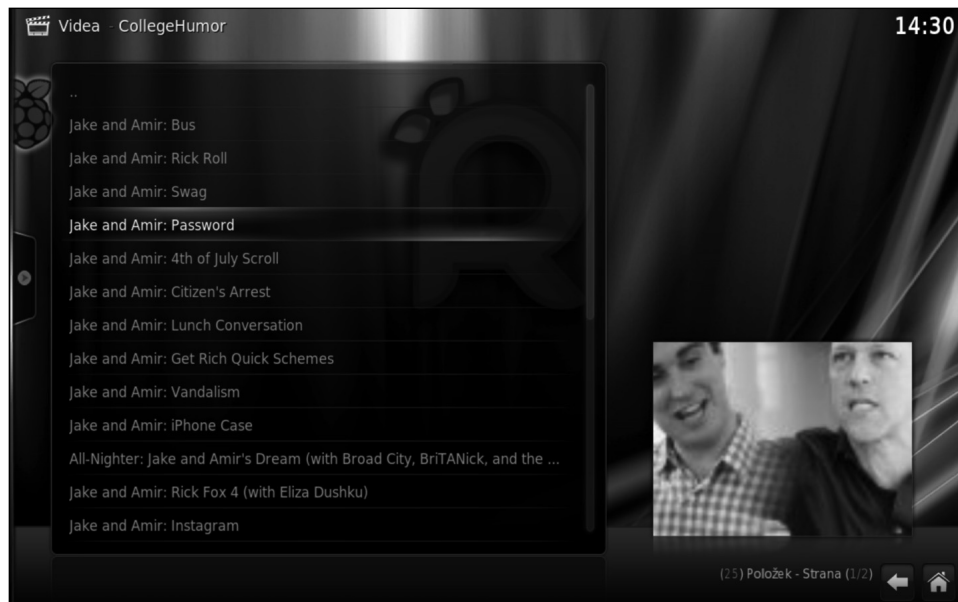


Obrázek 7.3: Přidání nových doplňků videa v distribuci Raspbmc

Posuňte obsah seznamu pomocí myši nebo kurzorových kláves. Chcete-li zobrazit další informace o položce, klepněte na ni nebo stiskněte klávesu Enter. Jestliže chcete nainstalovat doplněk, stačí klepnout na položku Install (Instalovat) v překryvném informačním okně, které se zobrazí po klepnutí na položku v seznamu. Tímto způsobem stáhnete doplněk z Internetu a automaticky jej nainstalujete do služby Xbmc. Všímejte si doplňků, které jsou označeny atributem Broken (Nefunkční). U těchto doplňků bylo zjištěno, že nefungují správně, a neměli byste je proto instalovat, dokud jejich vývojář příslušný problém neopraví.

Když dokončíte výběr a instalaci doplňků videa, přejděte zpět do hlavního rozhraní služby Xbmc klepnutím na tlačítko Home (Domů) v pravém dolním rohu obrazovky. Poté klepněte na slovo Video uprostřed obrazovky nebo stiskněte klávesu Enter a následně ze zobrazených možností zvolte položku Video Add-ons (Doplňky videa). Tím získáte přístup k nainstalovaným doplňkům. Klepnutím na některý z nich stáhnete seznam souborů k prohlížení. Pokud doplněk zahrnuje více kategorií, stáhne nejdříve tyto kategorie – v tomto případě klepněte na jednotlivé kategorie, abyste zobrazili soubory, které obsahují (viz obrázek 7.4).

Podobné doplňky jsou dostupné v nabídkách Music a Video a fungují stejným způsobem. Tyto doplňky umožňují prohlížet obrazový obsah a streamovat zvukový obsah z webů jako Flickr, Picasa, The Big Picture, Grooveshark, Sky.fm a SoundCloud.



Obrázek 7.4: Seznam videí, která jsou dostupná prostřednictvím doplňku CollegeHumor služby Xbmc

Streamování médií v místní síti

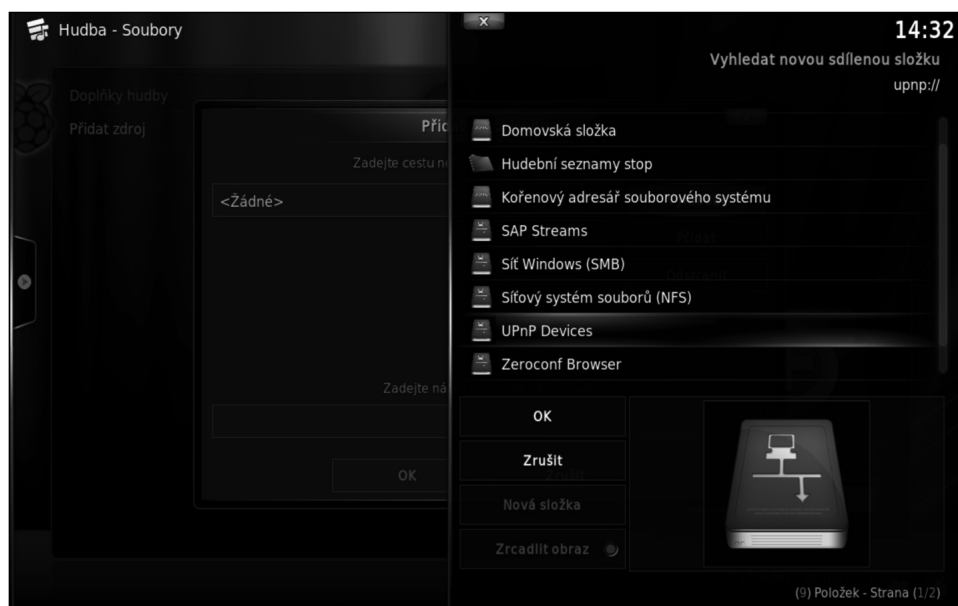
Software Xbmc podporuje standard streamování médií *UPnP* (*Universal Plug and Play*). Díky tomu se může připojit k mnoha různým zařízením v domácí síti. Podpora standardu UPnP je k dispozici u většiny moderních mobilních telefonů, herních konzol a síťových úložných zařízení *NAS* (*network-attached storage*), takže lze sdílet a streamovat video, hudbu a obrázky. Také mnoho notebooků a stolních počítačů je kompatibilních se standardem UPnP nebo úzce příbuzným standardem *DLNA* (*Digital Living Network Alliance*). Postup, jak tuto funkci povolit v konkrétním zařízení, naleznete v jeho dokumentaci.

Služba Xbmc se však neomezuje jen na připojení UPnP. Software se dokáže připojit i k síťovým serverům standardu *NFS* (*Network File System*), který je běžný u unixových systémů, standardu *SMB* (*Server Message Block*) obvyklému u serverů systému Windows a standardu *Zeroconf*, který je typický pro zařízení se systémem OS X. Bez ohledu na to, které síťové zařízení slouží k ukládání mediálního obsahu, je pravděpodobné, že služba Xbmc bude podporovat alespoň jeden způsob přístupu k němu.

Chcete-li nakonfigurovat službu Xbmc pro připojení k serveru v domácí síti, zvolte typ média – Video, Music (Hudba) nebo Pictures (Obrázky) a klepněte na možnost Add Source (Přidat zdroj). V zobrazeném okně zvolte možnost Browse (Vyhledat), která načte seznam zdrojových typů (viz obrázek 7.5). Tyto zdrojové typy zahrnují místní jednotky připojené k počítači Raspberry Pi, které jsou zvýrazněny modrou ikonou, stejně jako síťová zařízení označená

červenou ikonou. Ze seznamu vyberte typ serveru, ke kterému se pokoušíte připojit, a poté klepněte na zobrazený server.

Jestliže vybraný server poskytuje více složek (například složky pro různé žánry, interprety, alba nebo typy souborů), vyberte složku, ke které se pomocí služby Xbmc chcete připojit, a klepněte na tlačítko OK. Tím přejdete zpět na obrazovku Add Source (Přidat zdroj), kde budou vyplněny požadované informace. Pokud je nutné uvést další podrobnosti (například uživatelské jméno a heslo u chráněného serveru), musíte je zadat před klepnutím na tlačítko OK.



Obrázek 7.5: Volba síťového zdroje ve službě Xbmc

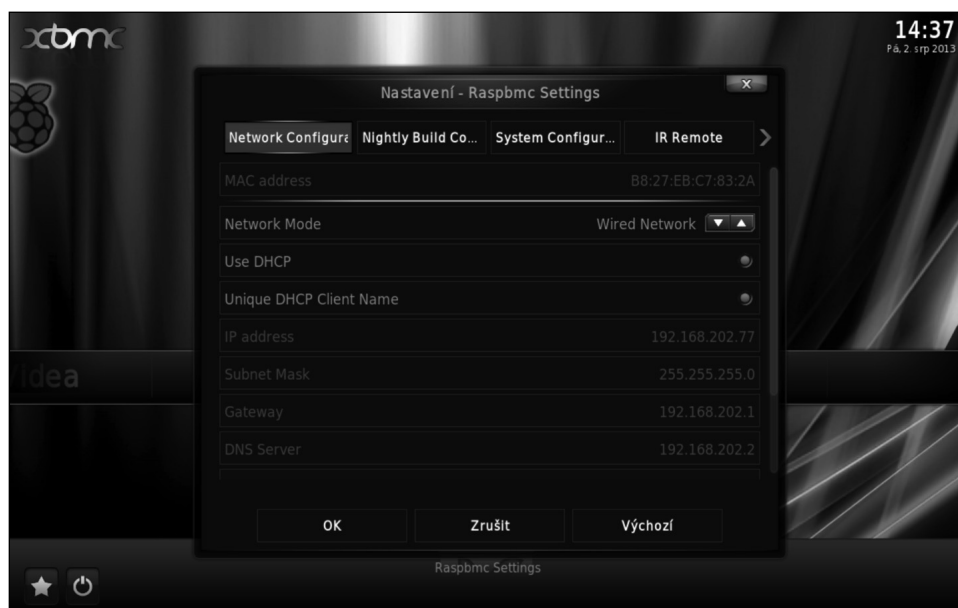
Pomocí stejné nabídky můžete také přidat jako zdroj služby Xbmc externí pevný disk, když vyberete jeho položku z počátečního seznamu. Většina externích jednotek se zobrazí automaticky a není nutné je explicitně přidávat. Jednotku je nutné přidat jako zdroj pouze v případě, že se její obsah neobjeví v nabídkách služby Xbmc.

Konfigurace distribuce Raspbmc

Nabídka Programs (Programy) na domovské obrazovce poskytuje přístup k nabídce Raspbmc Settings (viz obrázek 7.7), kde je možné upravit nastavení sítě a softwarových upgradů a různá systémová nastavení. Tuto nabídku můžete obvykle ignorovat, ale máte-li problémy s připojením k Internetu nebo neustále dochází k haváriím systému Raspbmc, můžete v nabídce Programs (Programy) najít užitečné nástroje, pomocí nichž lze potíže vyřešit.

Po klepnutí na možnost Raspbmc Settings v nabídce Programs (Programy) se zobrazí okno s několika kartami. Na první kartě lze upravit nastavení sítě, například vypnout DHCP a zvolit

ruční konfiguraci sítě (další podrobnosti naleznete v kapitole 4, „Konfigurace sítě“). Potřebujete-li počítač Raspberry Pi stejně jako v jiných distribucích nastavit tak, aby pracoval s ruční konfigurací sítě, můžete to v distribuci Raspbmc provést na tomto místě.



Obrázek 7.6: Konfigurace distribuce Raspbmc v nabídce Raspbmc Settings

Druhá karta s názvem Nightly Build Configuration umožňuje instalovat *noční* sestavení softwaru XBMC. Noční sestavení získalo svůj název podle toho, že vzniká ze zdrojových kódů programu automaticky každou noc a obsahuje poslední změny provedené vývojáři softwaru. V důsledku toho může zahrnovat opravy případných chyb, se kterými jste se mohli setkat, ale může také přinášet nevyzkoušené změny, které někdy způsobují nové problémy. Na stejné obrazovce lze přecházet mezi různými verzemi služby XMBC. Díky tomu je možné snadno vyzkoušet noční sestavení a při výskytu potíží přejít zpět ke standardní verzi.

Poslední karta System Configuration dovoluje nastavit nové heslo uživatelského účtu pi, aby byla distribuce Raspbmc zabezpečena před neoprávněným přístupem ze sítě. Stejně jako u univerzální distribuce Debian se ve výchozím nastavení používá heslo *raspberr*y. Na kartě System Configuration je rovněž možné vypnout automatické upgrady softwaru Raspbmc. Tuto funkci byste měli ponechat pokud možno zapnutou, protože zajišťuje, že budete automaticky dostávat nejnovější opravy zabezpečení a chyb. Pokud však máte internetové připojení s měřenými přenosy dat, můžete funkci zakázat.

Počítač Pi jako kancelářský počítač

Vzhledem ke své univerzálnosti může počítač Raspberry Pi zastávat roli univerzálního stolního počítače s nízkou spotřebou. Nikdy sice nedosáhne stejného výkonu jako standardní stolní počítač nebo notebook, ale případné problémy, které souvisejí s občasnými výkonnostními nedostatky, vynahrazuje svou nízkou cenou a ekologicky příznivou spotřebou energie.

Distribuce Debian určená pro počítač Raspberry Pi, která je k dispozici na oficiálním webu, sice neobsahuje žádný běžný kancelářský program, který bychom mohli u univerzálního počítače očekávat (například textový procesor, tabulkový procesor nebo program na úpravy obrázků), avšak uvedené programy lze nainstalovat příkazem `apt-get`. Můžete také vynechat místní instalaci a prostřednictvím webového prohlížeče používat software v cloudu, který může oproti místně nainstalovaným balíčkům nabídnout vyšší výkon za cenu nižší flexibility a chybějících pokročilých funkcí.

Pomocí jedné z metod popsaných v této kapitole (místně nainstalovaných aplikací nebo služeb založených na cloudu), je možné počítač Pi nasadit na běžnou kancelářskou a školní práci, což přitom nijak nenarušuje jeho použitelnost jako platformy k programování a experimentování.



Tip: Jestliže plánujete počítač Pi používat výhradně jako kancelářský stroj, je vhodné vyhradit pro univerzální výpočetní zpracování více než 256 MB paměti a snížit kapacitu paměti pro grafický procesor. Postup změny tohoto rozdělení naleznete v části „Rozdělení paměti – soubor `start.elf`“ v kapitole 6, „Konfigurace počítače Raspberry Pi“.

Použití cloudových aplikací

Pokud máte počítač Pi zpravidla připojen k Internetu, ať už pomocí integrovaného portu sítě Ethernet modelu B, nebo kabelového či bezdrátového adaptéru pro USB v případě modelu A, může *cloudový* (cloud-based) software poskytnout výkonný a přitom nenáročný způsob, jak v počítači Pi pracovat s kancelářskými aplikacemi.

Cloudový software odvozuje svůj název z toho, že není umístěn v lokálním počítači jako běžné programy. Místo toho je uložen na výkonných serverech, které se nacházejí v datových centrech na různých místech světa, a je dostupný přes Internet pomocí webového prohlížeče. Díky využití výpočetních a úložných kapacit mnohem výkonnějšího vzdáleného serveru je možné s počítačem Pi pracovat na mnohem složitějších dokumentech a úkolech, aniž by se přitom zpomaloval.

Cloudový software nabízí oproti místně nainstalovaným aplikacím také další výhody. Každá cloudová aplikace vypadá v každém přístupovém zařízení stejně. Mnoho z těchto aplikací zahrnuje také verze určené pro mobilní zařízení, k nimž lze přistupovat ze smartphonů a tabletů. Na vzdálených souborech se také ukládají soubory, které jsou díky tomu přístupné z libovolného zařízení a nezabírají žádné místo na kartě SD počítače Pi.

Cloudové aplikace však nejsou dokonalé. Za svými lokálně nainstalovanými protějšky obvykle zaostávají svými funkcemi a často jim chybí pokročilejší funkce nebo nepodporují tolik formátů souborů. Nejsou také dostupné v situaci, kdy není k dispozici připojení k Internetu, takže se příliš nehodí pro uživatele s nespolehlivou konektivitou.

Pokud se domníváte, že vyšší výkon a úspora místa na kartě SD vašeho počítače Pi za tyto nevýhody stojí, pokračujte ve čtení. V opačném případě přejděte na další část této kapitoly, kde se dozvíte, jak nainstalovat kancelářský balík typu open source s názvem OpenOffice.org, který umožňuje nahradit Microsoft Office.

Uvedme nejoblíbenější kancelářské balíky založené na cloudu:

- **Disk Google** (Google Drive) – služba Disk Google (dříve označovaná jako Dokumenty Google – Google Docs), kterou provozuje vedoucí poskytovatel vyhledávacích a reklamních služeb Google, zahrnuje textový procesor, tabulkový procesor a prezentační nástroj (viz obrázek 8.1). Podnikoví uživatelé si také mohou objednat účet Google Apps, který poskytuje zdokonalené funkce. Máte-li webový e-mailový účet Gmail, bude se službou Disk Google automaticky spolupracovat. Služba je dostupná na adrese <http://docs.google.com>.
- **Zoho** – Služba Zoho s pěti miliony registrovaných uživatelů představuje další oblíbenou volbu. Stejně jako u Disku Google je její součástí textový procesor, tabulkový procesor a prezentační balíček, ale služba Zoho poskytuje také vylepšené podnikové funkce, jako je systém znalostní báze založený na stránkách wiki, webové konference, správa financí, a dokonce i řízení vztahů se zákazníky. Mnoho z pokročilých funkcí však vyžaduje placený účet. Služba je přístupná na adrese <http://www.zoho.com>.
- **Office 365** – používáte-li sadu Microsoft Office, můžete s výhodou sáhnout po službě Office 365. Tato služba, která vychází ze stejného uživatelského rozhraní jako aktuální vydání sady Microsoft Office pro stolní počítače, je výkonná a flexibilní. Na rozdíl od služeb Zoho a Disk Google však služba Office 365 neposkytuje žádnou bezplatnou variantu a vyžaduje měsíční předplacení. Kromě toho v případě přístupu z počítače se systémem Linux nelze použít některé funkce. Službu si můžete objednat na adrese <http://office365.microsoft.com>.
- **ThinkFree Online** – toto webové rozhraní k softwaru Hancom ThinkFree Office nabízí textový procesor, tabulkový procesor a prezentační software zdarma s úložištěm velikos-

ti 1 GB. Systém je také propojen se službou ThinkFree Mobile pro tablety a smartphony a také se softwarem ThinkFree Server, který je určen pro podniky. Předplatné služby je možné získat na adrese <http://online.thinkfree.com>.

Mnoho z těchto webových služeb bohužel vyžaduje výkonnější prohlížeč než Midori, který je v doporučené distribuci Debian standardně k dispozici. Chcete-li tedy pracovat s libovolnou z těchto sad cloudových aplikací, musíte nainstalovat jiný prohlížeč. Uvedme pokyny k instalaci prohlížeče Chromium, což je projekt open source, na němž je založen prohlížeč Chrome společnosti Google. Díky poměrně nízkým paměťovým nárokům představuje prohlížeč Chromium pro počítač Pi dobrou volbu. Pokud používáte distribuci Fedora Remix, která jako svůj výchozí prohlížeč obsahuje program Mozilla Firefox, můžete tyto pokyny přeskočit a přejít k další části kapitoly.



Obrázek 8.1: Služba Disk Google spuštěná v prohlížeči Chromium v počítači Raspberry Pi

Chcete-li nainstalovat prohlížeč Chromium v distribuci Debian, otevřete okno terminálu a zadejte příkaz:

```
sudo apt-get install chromium-browser
```



Tip: Dbejte na to, abyste nainstalovali balíček **chromium-browser**, a nikoli balíček **chromium**. Druhý z nich patří střílecí hře, která je sice zábavná, ale při používání cloudových kancelářských balíčků v počítači Pi vám nijak nepomůže!

Po nainstalování prohlížeče Chromium stačí k použití cloudového kancelářského balíku přejít na příslušný web, zaregistrovat svůj účet (v případě placených služeb typu Microsoft Office 365 se zadáním údajů o platební kartě) a přihlásit se. Pokud budete mít dojem, že výkon je příliš nízký, můžete jej zlepšit změnou rozdělení paměti tak, aby procesor ARM získal její větší část. Podrobné pokyny, jak to provést, naleznete v kapitole 6, „Konfigurace počítače Raspberry Pi“.

Použití sady OpenOffice.org

Jestliže nechcete přistupovat ke cloudové službě, další variantou je instalace sady OpenOffice.org. Sada OpenOffice.org, která byla navržena jako multiplatformní alternativa typu open source k oblíbené sadě Microsoft Office, je velmi výkonná a nabízí přibližně tolik funkcí jako její komerční konkurent.

Tyto funkce ovšem mají určitou cenu. Balíček sady OpenOffice.org je značně velký a se zahrnutím všech závislostí zabírá na kartě SD počítače Pi téměř 400 MB. To může být problém: ve své výchozí konfiguraci má doporučená distribuce Debian méně volného místa na kartě SD, než kolik je potřeba k instalaci sady OpenOffice.org. (Další informace o struktuře oddílů distribuce Debian naleznete v části „Struktura systému souborů“ v kapitole 2, „Správa systému Linux“.)

Chcete-li sadu OpenOffice.org nainstalovat, potřebujete kartu SD s kapacitou 4 GB nebo více. Musíte také změnit velikost kořenového systému souborů, abyste mohli využít volné místo na větší kartě SD. Pokyny, jak to provést, naleznete v kapitole 5, „Správa oddílů“. Nezapomeňte uvedenou operaci provést dříve, než budete pokračovat ve čtení této kapitoly, protože jinak na kartě SD v počítači Pi dojde volné místo dříve, než bude možné se sadou OpenOffice.org pracovat.

Jestliže je na kartě SD dostatek volného místa, není instalace sady OpenOffice.org o nic složitější než instalace libovolného jiného balíčku. Sada se sice dodává společně s mnoha doplňkovými balíčky, ale všechno je součástí jediného *metabalíčku*, díky kterému lze software nainstalovat jediným příkazem. Otevřete okno terminálu a zadejte příkaz:

```
sudo apt-get install openoffice.org
```



Tip: Pokud se při instalaci softwaru typu OpenOffice.org zobrazují chybové zprávy o chybějících souborech, pravděpodobně je to způsobeno neaktuálními mezipaměťmi balíčků. Spusťte příkaz **sudo apt-get update**, který mezipaměť aktualizuje, a opakujte akci. (Další podrobnosti naleznete v kapitole 2, „Správa systému Linux“.)

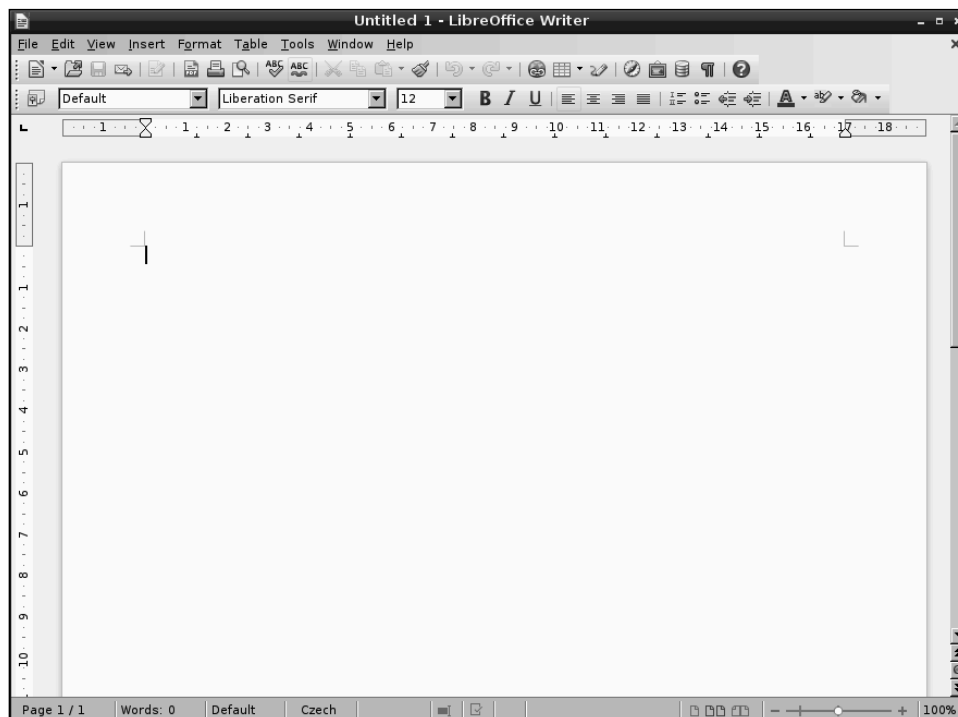


Poznámka: Přestože je ve výše uvedeném příkazu uveden balík OpenOffice.org, nainstalujete jím balík LibreOffice. Ten navazuje na původní OpenOffice.org a je s ním plně kompatibilní. Vznikl poté, co projekt OpenOffice.org přešel pod společnost Oracle kvůli obavám, že by Oracle mohl tento projekt ukončit. Proto v následujícím textu najdete místo OpenOffice.org označení LibreOffice.

Po nainstalování sady LibreOffice se v nabídce Applications (Aplikace) prostředí plochy počítače Pi zobrazí několik položek. Jedná se o následující položky:

- **LibreOffice** – hlavní aplikace, která poskytuje odkazy na jednotlivé části sady
- **LibreOffice Calc** – aplikace tabulkového procesoru, která odpovídá aplikaci Microsoft Excel
- **LibreOffice Draw** – aplikace na vektorové ilustrování, která umožňuje kreslit vysoce kvalitní škálovatelné obrázky použitelné jako kliparty v jiných programech sady OpenOffice.org
- **LibreOffice Impress** – prezentační aplikace, která je ekvivalentní aplikaci Microsoft PowerPoint
- **LibreOffice Writer** – aplikace textového procesoru, která nahrazuje aplikaci Microsoft Word (viz obrázek 8.2)

Sada LibreOffice ve výchozím nastavení ukládá a načítá soubory ve formátu, který se označuje jako *ODF (Open Document Format)*. Jedná se o formát souborů založený na standardech a nezátížený žádnými licenčními poplatky, jehož podporu nabízí většina sad kancelářských programů – včetně novějších verzí sady Microsoft Office.



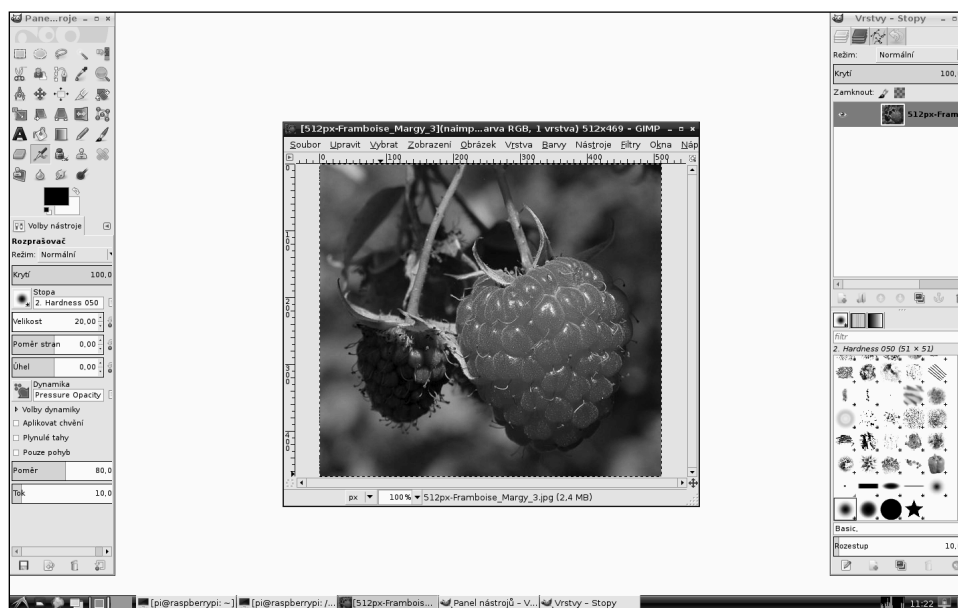
Obrázek 8.2: Aplikace LibreOffice Writer spuštěná v počítači Raspberry Pi

Při ukládání souboru v sadě LibreOffice můžete formát změnit pomocí rozevřacího seznamu v dialogovém okně Save As (Uložit jako). V poli File Type (Typ souboru) lze vybrat některý z mnoha formátů včetně několika z nich, které jsou plně kompatibilní se staršími verzemi sady Microsoft Office. Pokud pomocí sady LibreOffice v počítači Pi vytváříte soubory, které chcete sdílet s uživateli starších programů, nezapomeňte formát změnit, aby soubory dokázali otevřít všichni uživatelé. Případně se je můžete pokusit přesvědčit, aby také nainstalovali sadu LibreOffice, která je bezplatně dostupná pro systém Linux, Windows i OS X.

Úpravy obrázků programem Gimp

Sada LibreOffice sice poskytuje široké možnosti, ale zaostává v oblasti editace obrazu. I když je program LibreOffice Draw výkonný nástroj na tvorbu ilustrací, bohužel v něm nelze retušovat digitální fotografie. Takové soubory se nazývají *rastrové obrázky* (bitmapy) a značně se liší od vektorových obrázků, k jejichž úpravám program Draw slouží.

Mezi nejvýkonnější nástroje, které jsou v systému Linux dostupné na úpravy obrázků, patří GNU Image Manipulation Program, který se obvykle označuje jako Gimp. Gimp patří mezi nejoblíbenější projekty hnutí open source, protože nabízí bohaté možnosti úprav rastrových obrázků a jeho uživatelské rozhraní je podobné rozhraní komerčního balíku Adobe Photoshop (viz obrázek 8.3).



Obrázek 8.3: Program Gimp spuštěný v počítači Raspberry Pi

Ve většině distribucí pro počítač Raspberry Pi není program Gimp standardně nainstalován, takže je nutné počítač Pi připojit k Internetu a k instalaci využít systém správy balíčků (po-

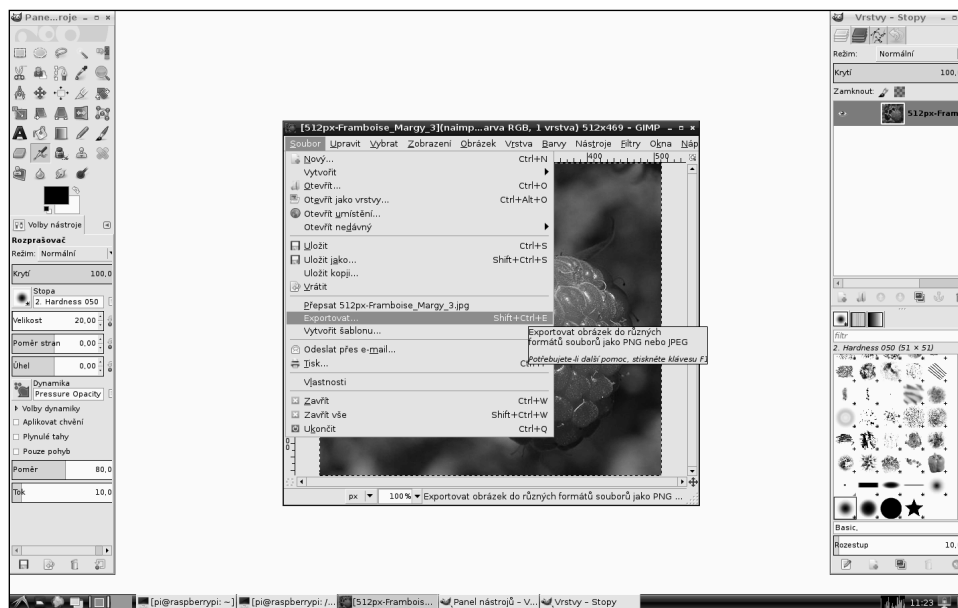
drobnosti naleznete v kapitole 2, „Správa systému Linux“). Program Gimp zabírá na kartě SD poměrně hodně místa, ačkoli poněkud méně než sada LibreOffice. Před instalací se tedy nezapomeňte přesvědčit, zda má karta dostatečnou volnou kapacitu.

Chcete-li program Gimp nainstalovat, otevřete okno terminálu a zadejte příkaz:

```
sudo apt-get install gimp
```

Seznámení s programem Gimp nějakou dobu trvá, protože jeho uživatelské rozhraní zahrnuje místo jednoho obvyklého okna hned tři. Okno na levé straně ve výchozím nastavení obsahuje Toolbox (Panel nástrojů), okno vpravo poskytuje možnosti Layers (Vrstvy), Channels (Kanály) a Gradients (Cesty) a prostřední okno znázorňuje právě upravovaný obrázek. Pokud otevřete více obrázků, dostanete více než jedno hlavní okno, ale jak okno Toolbox (Panel nástrojů), tak okna Layers (Vrstvy), Channels (Kanály) a Gradients (Cesty) jsou zobrazena pouze jednou.

Uživatelská příručka (*The Gimp User Manual*) standardně není nainstalována. Z hlediska počítače Pi je to výhodné: Gimp je značně výkonný nástroj a jeho uživatelská příručka zabírá nezanedbatelnou část karty SD. Jestliže se pokusíte o přístup k příručce stisknutím klávesy F1 nebo volbou příkazu Help (Nápověda) ze stejnojmenné nabídky, zobrazí se doporučení, abyste si přečetli webovou verzi online. Chcete-li uživatelskou příručku otevřít v prohlížeči, klepněte na tlačítko Read Online (Přečíst on-line).



Obrázek 8.4: Export souboru z programu Gimp

Program Gimp nabízí mnoho možností a spotřebuje také hodně paměti. Vzhledem k tomu funguje v počítači Raspberry Pi poměrně pomalu, i když se s ním rozhodně dá pracovat. Při používání programu buďte trpěliví, zejména otevíráte-li velké fotografie z digitálního fotoaparátu.

tu. V některých případech pomůže zvýšit rozsah dostupné systémové paměti změnou souboru `start.elf` (podrobný postup naleznete v kapitole 6, „Konfigurace počítače Raspberry Pi“).

Při ukládání souborů z programu Gimp můžete zvolit různé formáty souborů. Pokud plánujete, že se k souboru vrátíte a budete jej dále upravovat, měli byste použít výchozí formát souborů XCF programu Gimp. Tento formát uchovává beze změny důležitá *metadata*, kvůli dosažení maximální kvality obrazu používá bezeztrátovou kompresi a podporuje obrázky, které se skládají z více vrstev.

Jestliže plánujete obrázky nahrávat na Internet nebo je jinak sdílet s ostatními, lze doporučit přenosnější formát typu JPG či PNG. Chcete-li formát souboru změnit, zvolte z nabídky File (Soubor) namísto příkazu Save (Uložit) možnost Export (Exportovat) – viz obrázek 8.4. Tímto způsobem můžete zvolit formát souborů z různých dostupných typů.

Počítač Pi jako webový server

Počítač Pi sice svým výkonem značně zaostává za většinou zařízení, která se nacházejí v datovém centru, což však neznamená, že by nedokázal fungovat jako užitečný server v domácím nebo podnikovém prostředí. Navzdory nevelké kapacitě paměti a relativně málo výkonnému procesoru představuje počítač Pi díky své nízké spotřebě energie a tichému chodu ideální volbu k obsluze málo vytižených webových stránek v místní síti, nebo dokonce v Internetu.

Značná část moderních webových serverů je založena na kombinaci systému Linux, webového serveru Apache, databáze MySQL a jazyka PHP, jejichž kombinace se běžně označuje jako *sada LAMP* (LAMP stack). Systém Linux poskytuje základní operační systém, MySQL databázový back-end, jako webový server slouží Apache a jazyk PHP umožňuje skriptování dynamických stránek. Díky webovému serveru založenému na sadě LAMP můžete používat poměrně složité balíčky od systémů správy obsahu, jako je WordPress, až po interaktivní fóra typu phpBB. To vše je dostupné i v počítači Raspberry Pi za předpokladu, že nebudete očekávat podobný výkon jako u nadupaného komerčního serveru.



Tip: Webové servery fungují nejlépe, mají-li k dispozici spoustu paměti. Pokud chcete zajistit maximální výkon, upravte rozdělení paměti počítače Pi na poměr 224/32 MB (viz kapitola 6, „Konfigurace počítače Raspberry Pi“) a současně nespouštějte *grafické uživatelské rozhraní (GUI)*.

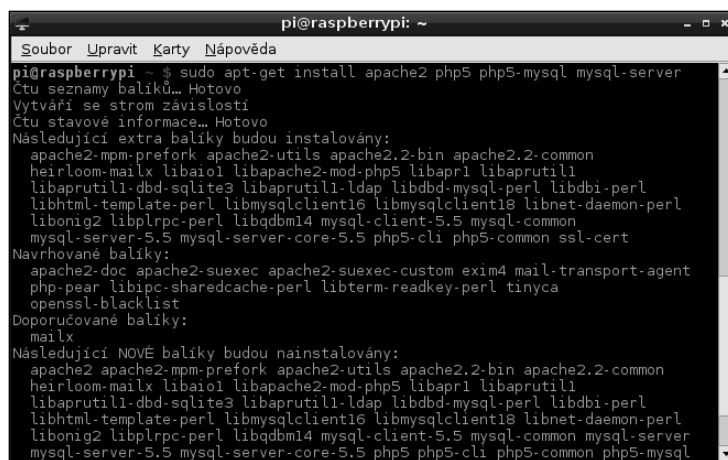
Instalace sady LAMP

Jestliže používáte doporučenou distribuci Debian pro počítač Raspberry Pi, máte za sebou již první krok zprovoznění sady LAMP – instalaci systému Linux. Dalším krokem je instalace chybějících komponent: Apache, MySQL a PHP. Chcete-li tyto balíčky nainstalovat, zadejte v okně terminálu nebo v konzole následující příkaz:

```
sudo apt-get update
sudo apt-get install apache2 php5 php5-mysql mysql-server
```

Tímto příkazem požádáte správce balíčků apt (viz kapitola 2, „Správa systému Linux“), aby vyhledal všechny závislosti, které jsou nezbytné k činnosti plně funkční sady LAMP (viz obrázek 9.1). Tyto balíčky a jejich závislosti zabírají na kartě SD v počítači Pi hodně místa – celkem asi 113 MB. Pokud jste tedy zatím nezměnili velikost svého kořenového oddílu na kartě SD, přejděte na kapitolu „Správa oddílů“, kde naleznete pokyny, jak získat více volného místa.

Instalace kompletní sady LAMP do počítače Pi může trvat poměrně dlouho. Nelekejte se, pokud se zdá, že systém na minutu či dvě „zamrzl“. Instalace by poté měla normálně pokračovat. V polovině instalačního procesu se zobrazí výzva databáze MySQL k zadání hesla (viz obrázek 9.2). Nezapomeňte zvolit bezpečné heslo, protože bude chránit databázi MySQL. Podle konfigurace webového serveru může tato databáze uchovávat uživatelská jména, údaje o platbách a jiné citlivé osobní údaje. Přitom také dávejte pozor, abyste si vybrané heslo dokázali zapamatovat! Zobrazí se výzva k potvrzení hesla, která má vyloučit případné překlipy, a poté bude instalace pokračovat.

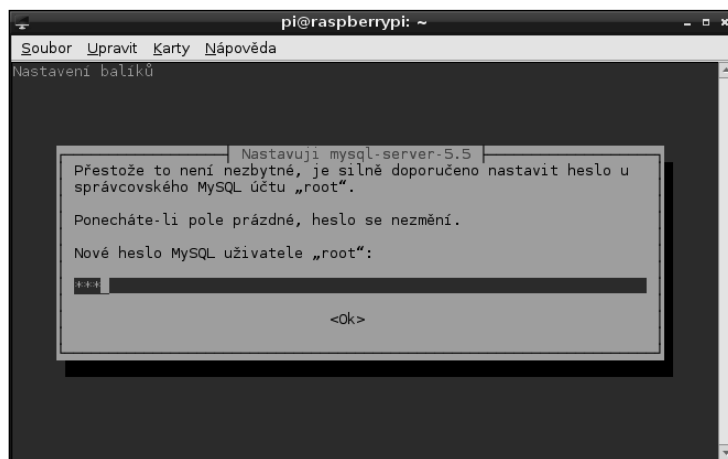


```

pi@raspberrypi: ~
Soubor Upravit Karty Nápověda
pi@raspberrypi ~ $ sudo apt-get install apache2 php5 php5-mysql mysql-server
Čtu seznamy balíků... Hotovo
Vytváří se strom závislostí
Čtu stavové informace... Hotovo
Následující extra balíky budou instalovány:
 apache2-mpm-prefork apache2-utils apache2.2-bin apache2.2-common
 heirloom-mailx libaio1 libapache2-mod-php5 libapr1 libaprutil1
 libaprutil1-dbd-sqlite3 libaprutil1-ldap libdbd-mysql-perl libdbi-perl
 libhtml-template-perl libmysqlclient16 libmysqlclient18 libnet-daemon-perl
 libonig2 libplrpc-perl libqdbm14 mysql-client-5.5 mysql-common
 mysql-server-5.5 mysql-server-core-5.5 php5-cli php5-common ssl-cert
Navrhované balíky:
 apache2-doc apache2-suexec apache2-suexec-custom exim4 mail-transport-agent
 php-pear libipc-sharedcache-perl libterm-readkey-perl tinyca
 openssl-blacklist
Doporučované balíky:
 mailx
Následující NOVÉ balíky budou nainstalovány:
 apache2 apache2-mpm-prefork apache2-utils apache2.2-bin apache2.2-common
 heirloom-mailx libaio1 libapache2-mod-php5 libapr1 libaprutil1
 libaprutil1-dbd-sqlite3 libaprutil1-ldap libdbd-mysql-perl libdbi-perl
 libhtml-template-perl libmysqlclient16 libmysqlclient18 libnet-daemon-perl
 libonig2 libplrpc-perl libqdbm14 mysql-client-5.5 mysql-common mysql-server
 mysql-server-5.5 mysql-server-core-5.5 php5 php5-cli php5-common php5-mysql

```

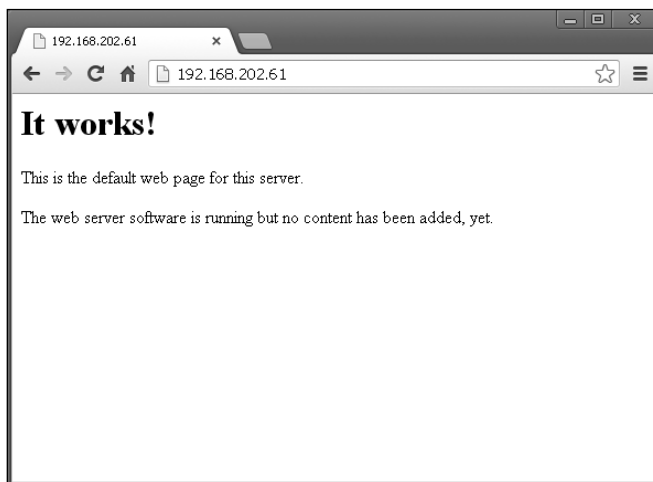
Obrázek 9.1: Instalace sady LAMP v distribuci Debian



Obrázek 9.2: Volba hesla databáze MySQL

Po dokončení instalace softwaru budou servery MySQL a Apache – v terminologii systému Linux se označují jako *démony* (daemon) – spuštěny na pozadí. Chcete-li zkontrolovat, zda

server funguje správně, pokuste se pomocí webového prohlížeče připojit k počítači Raspberry Pi z jiného počítače v síti. Na panel adresy zadejte IP adresu počítače Pi, která zobrazí výchozí stránku instalace serveru Apache (viz obrázek 9.3). Pokud nevíte, která IP adresa je počítači Pi přiřazena, zadejte v okně terminálu příkaz `ifconfig` a vyhledejte IP adresu v sekci `eth0`, případně v sekci odpovídající použitému síťovému adaptéru počítače Pi, pokud tento není připojen pomocí integrovaného ethernetového portu modelu B. Další informace o kontrole a úpravách nastavení sítě v počítači Pi naleznete v kapitole 4, „Konfigurace sítě“.



Obrázek 9.3: Připojení k serveru Apache počítače Pi pomocí webového prohlížeče

Jako poslední krok je potřeba zkontrolovat, zda je na serveru Apache správně načten skriptovací modul PHP. Tento modul je důležitý, protože webový server Apache může díky němu spouštět skripty PHP a poskytovat dynamický obsah. Bez funkčního modulu PHP – nebo alternativního modulu pro jiný skriptovací jazyk, jako je Python – dokáže server Apache zobrazovat pouze statické stránky. Chcete-li modul PHP serveru Apache vyzkoušet, vytvořte nový soubor skriptu PHP pomocí následujícího příkazu zadaného na jediný řádek:

```
sudo sh -c 'echo "<?php phpinfo(); ?>" > \n/var/www/phptest.php'
```

Tento příkaz vytvoří nový soubor s názvem `phptest.php` v adresáři `/var/www`. Tento soubor požádá modul PHP, aby generoval informační stránku pro diagnostické účely. Chcete-li tuto stránku zobrazit z prohlížeče v jiném počítači, zadejte na panel adresy `http://ip_adresa/phptest.php` (kde výraz *ip_adresa* nahradíte IP adresou počítače Raspberry Pi). V samotném počítači Pi zadejte na panel adresy `http://localhost/phptest.php` (viz obrázek 9.4).

Když modul vyzkoušíte, odstraňte soubor `phptest.php` tímto příkazem:

```
sudo rm /var/www/phptest.php
```



Obrázek 9.4: Testování modulu PHP serveru Apache v počítači Raspberry Pi



Tip: Webový server Apache je sice nejrozšířenější, ale k dispozici jsou i jiné. Pokud zjistíte, že server Apache funguje pro vaše účely příliš pomalu, vyzkoušejte odlehčený webový server `lighttpd`, který je navržen tak, aby nevyžadoval tolik paměti jako server Apache. V distribuci Debian jej můžete nainstalovat příkazem `sudo apt-get install lighttpd`.

Po instalaci a zprovoznění sady LAMP můžete nyní vytvářet vlastní weby, které bude počítač Pi zpřístupňovat. Za předpokladu, že stránky nejsou příliš složité a nepřístupuje k nim příliš mnoho uživatelů současně, měl by počítač Pi tento úkol zvládnout na jedničku. Malé rozměry a mimořádně nízké nároky na napájení tohoto zařízení pak více než vynahradí případný pokles výkonu v případě, že se zvýší popularita příslušného webu.

Soubory webového serveru jsou ve výchozím nastavení uloženy ve složce `/var/www`, do které může zapisovat jen uživatel `root`. Pokud chcete změnit umístění, kde server Apache vyhledává soubory (například při přesunu webu na externí úložiště s větší kapacitou), upravte textový soubor `000-default`, který se nachází ve složce `/etc/apache2/sites-enabled`. Další informace o konfiguraci serveru Apache, modulu PHP a databázi MySQL získáte zadáním následujících příkazů v okně terminálu nebo v konzole:

```
man apache2
man php5
man mysql
```

Instalace platformy WordPress

WordPress patří mezi nejoblíbenější blogovací platformy. Jedná se o projekt typu open source, který má uživatelům poskytnout jednoduchou, ale výkonnou základnu k vytváření atraktivních a interaktivních webů. Platforma WordPress je založena na skriptovacích jazycích PHP a JavaScript a nabízí atraktivní webové rozhraní na tvorbu pestrých webů. Na přizpůsobené platformě WordPress jsou například postaveny některé z nejnavštěvovanějších zpravodajských webů.

Chcete-li nainstalovat platformu WordPress do počítače Raspberry Pi, zadejte v okně terminálu nebo v konzole tento příkaz:

```
sudo apt-get install wordpress
```

Podobně jako sada LAMP má také platforma WordPress řadu závislostí (viz obrázek 9.5). K úplné instalaci potřebujete asi 37 MB volného místa na kartě SD počítače Pi nad rámec 113 MB, které vyžaduje sada LAMP. Máte-li dostatek volného místa, pokračujte v procesu instalace stisknutím klávesy Y.

Po dokončení instalace platformy WordPress je nutné jeho výchozí instalační adresář – /usr/share/wordpress – *propojit* a adresářem /var/www, aby byly soubory viditelné pro server Apache. V okně terminálu zadejte následující příkaz:

```
sudo ln -s /usr/share/wordpress /var/www/wordpress
```

```
pi@raspberrypi: ~
Soubor Upravit Karty Nápořěda
pi@raspberrypi ~$ sudo apt-get install wordpress
Čtu seznamy balíků... Hotovo
Vytvářím se strom závislostí
Čtu stavové informace... Hotovo
Následující extra balíky budou instalovány:
 javascript-common libjs-cropper libjs-prototype libjs-scriptaculous
 libphp-phpmailer libphp-snoopy php5-gd tinymce wordpress-l10n
 wwwconfig-common
Navrhované balíky:
 mail-transport-agent postgresql-client
Následující NOVE balíky budou nainstalovány:
 javascript-common libjs-cropper libjs-prototype libjs-scriptaculous
 libphp-phpmailer libphp-snoopy php5-gd tinymce wordpress wordpress-l10n
 wwwconfig-common
0 aktualizováno, 11 nové instalováno, 0 k odstranění a 4 neaktualizováno.
Potřebuji stáhnout 14,7 MB archivů.
Po této operaci bude na disku použito dalších 44,9 MB.
Chcete pokračovat [Y/n]?
```

Obrázek 9.5: Instalace platformy WordPress do počítače Raspberry Pi

Propojení souboru nebo adresáře se liší od přesunutí: soubory platformy WordPress nyní existují v adresářích /usr/share i /var/www současně, aniž by na kartě SD počítače Pi zabíraly více místa než předtím. Pokud jste server Apache nastavili tak, aby pro výchozí web používal jiný adresář, odpovídajícím způsobem příkaz propojení upravte. Poté pro platformu WordPress spusťte konfigurační skript databáze MySQL pomocí následujícího příkazu, který zadejte na jediný řádek:

```
sudo bash /usr/share/doc/wordpress/examples/setup-mysql <
-n wordpress localhost
```



Poznámka: Pokud příkaz po spuštění oznámí, že soubor `setup-mysql` nenašel, je tento v daném adresáři zabalen. Rozbalíte jej příkazem `sudo gunzip /usr/share/doc/wordpress/examples/setup-mysql.gz`. Poté by již mělo proběhnout spuštění příkazu výše korektně.

Tento příkaz přidá do databázového systému MySQL, který je součástí sady LAMP, novou databázi určenou pro platformu WordPress. Tato databáze uchovává uživatelské účty, příspěvky, komentáře a další podrobnosti. Po dokončení činnosti skriptu se zobrazí pokyn, abyste v počítači Raspberry Pi přešli v okně prohlížeče na adresu `http://localhost` a pokračovali tím v instalaci. Tento pokyn obsahuje malou chybu: adresa, jejíž návštěvou lze dokončit instalaci platformy WordPress, má tvar `http://localhost/wordpress` (viz obrázek 9.6).



Obrázek 9.6: Konfigurace platformy WordPress ve webovém prohlížeči NetSurf

Vyplňte formulář načtený v okně webového prohlížeče. Přitom zvolte popisný název svého webu a nastavte bezpečné (ale zapamatovatelné) uživatelské heslo. Nezapomeňte změnit obsah pole Username (Uživatelské jméno) a místo účtu `admin` zadejte jiný uživatelský účet, abyste zvýšili bezpečnost platformy. Po vyplnění všech polí klepněte na tlačítko `Install WordPress` v dolní části stránky.

Dokončení procesu instalace trvá asi jednu až dvě minuty. Poté se v prohlížeči zobrazí nová stránka, která potvrzuje úspěšnou instalaci platformy WordPress. Chcete-li platformu začít používat, klepněte na tlačítko `Log In (Přihlášení)` v dolní části této stránky. Následně se můžete k platformě přihlásit pomocí uživatelského jména a hesla, které jste zvolili na předchozí obrazovce (viz obrázek 9.7).



Obrázek 9.7: Přihlášení k platformě WordPress v počítači Raspberry Pi

Chcete-li k platformě WordPress přistupovat z jiného počítače, musíte nejdříve vytvořit dodatečný konfigurační soubor. Tento soubor vytvoříte propojením existujícího konfiguračního souboru (určeného pro místní přístup) pomocí následujícího příkazu, který zadáte na jediný řádek:

```
sudo ln -s /etc/wordpress/config-localhost.php ↔  
/etc/wordpress/config-ip_adresa.php
```

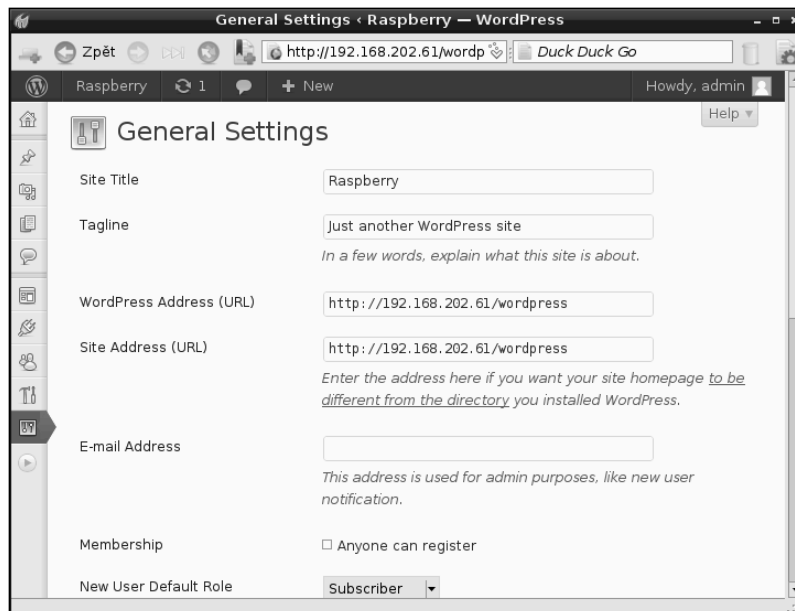
Řetězec *ip_adresa* v tomto kódu nahradíte IP adresou počítače Raspberry Pi. Jestliže jste počítači Raspberry Pi přidělili název hostitele v systému DNS, můžete také vytvořit konfigurační soubor pro daný název hostitele. Přitom zadejte stejný příkaz, ale místo hodnoty *ip_adresa* uveďte zvolený název hostitele. Pokud název hostitele nemáte, jednoduše použijte IP adresu počítače Pi. Příkaz pro počítač Pi s IP adresou 192.168.0.115 by například vypadal takto:

```
sudo ln -s /etc/wordpress/config-localhost.php ↔  
/etc/wordpress/config-192.168.0.115.php
```

Chcete-li dokončit konfiguraci pro externí přístup, zvolte z nabídky Settings (Nastavení), která je k dispozici na levé straně ovládacího panelu WordPress Dashboard, příkaz General (Obecné) a změňte adresu URL tak, aby buď odpovídala IP adrese počítače Pi, nebo vybranému názvu hostitele (viz obrázek 9.8).



Tip: Platforma WordPress má velké nároky na paměť. Chcete-li dosáhnout nejvyššího výkonu, použijte počítač Pi jako *bezobslužný server* (headless server), což je systém, který funguje bez načteného grafického uživatelského rozhraní. K ovládacímu panelu WordPress Dashboard pak můžete přistupovat na adrese **`http://ip_adresa/wordpress/wp-login.php`**. Pokud se k panelu WordPress Dashboard pokusíte připojit přímo z počítače Pi, může zobrazení panelu trvat značně dlouho!



Obrázek 9.8: Změna adresy URL v nabídce Settings-General (Nastavení-Obecné) platformy WordPress

Software WordPress zahrnuje funkci automatických aktualizací, která zajišťuje, že je v počítači vždy nainstalována nejnovější verze platformy. Vzhledem ke své oblibě je platforma WordPress častým cílem útoků malwaru. Průběžně se proto objevují aktualizace, které mají opravit bezpečnostní díry nebo doplnit nové funkce. Jestliže však platformu nainstalujete nástrojem APT, nemá dostatečná oprávnění k tomu, aby se mohla sama aktualizovat. Chcete-li tento problém vyřešit, zadejte v okně terminálu následující příkaz:

```
sudo chown -R www-data /usr/share/wordpress
```

Tím poskytnete uživateli `www-data` (což je účet používaný webovým serverem Apache) práva měnit soubory umístěné ve složce `/usr/share/wordpress`. Díky tomu budou fungovat automatické aktualizace, pokud je na panelu WordPress Dashboard zvolíte.

Další informace o použití platformy WordPress naleznete na oficiálním webu na adrese <http://www.wordpress.org>.

PROGRAMOVÁNÍ A HACKOVÁNÍ

V této části:

- **KAPITOLA 10** – Úvod do jazyka Scratch
- **KAPITOLA 11** – Úvod do jazyka Python
- **KAPITOLA 12** – Hackování hardwaru
- **KAPITOLA 13** – Rozšiřující desky

Úvod do jazyka Scratch

V předchozích kapitolách této knihy jste získali hodně informací o tom, jak používat programy, které pro počítač Raspberry Pi napsali jiní lidé. Hlavním cílem projektu Raspberry Pi je však naučit uživatele, aby psali své vlastní programy – a nevztahuje se to jen na dospělé. Nadace Raspberry Pi Foundation usiluje o to, aby se její počítač rozšířil jako vzdělávací nástroj pro všechny věkové skupiny.

K dosažení tohoto cíle je klíčové, aby se tvorbou vlastního softwaru mohly bavit i malé děti a nemusely jen konzumovat kód, který vytvořil někdo jiný. Hlavním nástrojem je přitom jazyk *Scratch*.

Představení jazyka Scratch

Jazyk Scratch vznikl roku 2006 ve skupině Lifelong Kindergarten laboratoře Media Lab Massachusettského technického institutu (MIT). Tento jazyk, odvozený od jazyků Squeak a Smalltalk, je založen na klíčových programátorských principech, které zpřístupňuje všem zájemcům. Místo psaní dlouhých řetězců, které je pro mladší děti únavné a nudné, se využívá jednoduché prostředí, v němž lze skládat stavební kameny přetažením myši. Přesto však rozvíjí programátorské myšlení a uvádí základní koncepce, které se používají ve všech programovacích jazycích.

Programovací prostředí je oficiálně určeno uživatelům od osmi let věku, ale s trochou pomoci a vedení je přístupné dokonce i mladším programátorům. Jazyk Scratch je překvapivě silný. Za barevným a myší ovládaným uživatelským prostředím se skrývá programovací jazyk, který obsahuje působivé multimediální funkce. Nemělo by vás překvapit, že většinu z více než 2,5 milionu projektů, které uživatelé jazyka Scratch sdílejí na jeho oficiálním webu, představují hry.

Když se děti motivují k tomu, aby se učily tvořit své vlastní hry, lze do jejich zábavy vpašovat i trochu učení. Díky přátelskému uživatelskému rozhraní a vynikajícímu zpracování klíčových programovacích principů by se nemělo stát, že děti odradí náhlé nepřekonatelné problémy. Navíc koncepce osvojené v jazyce Scratch poskytují vynikající základ pro přechod na pružnější jazyky, jako je Python (viz kapitola 11, „Úvod do jazyka Python“).

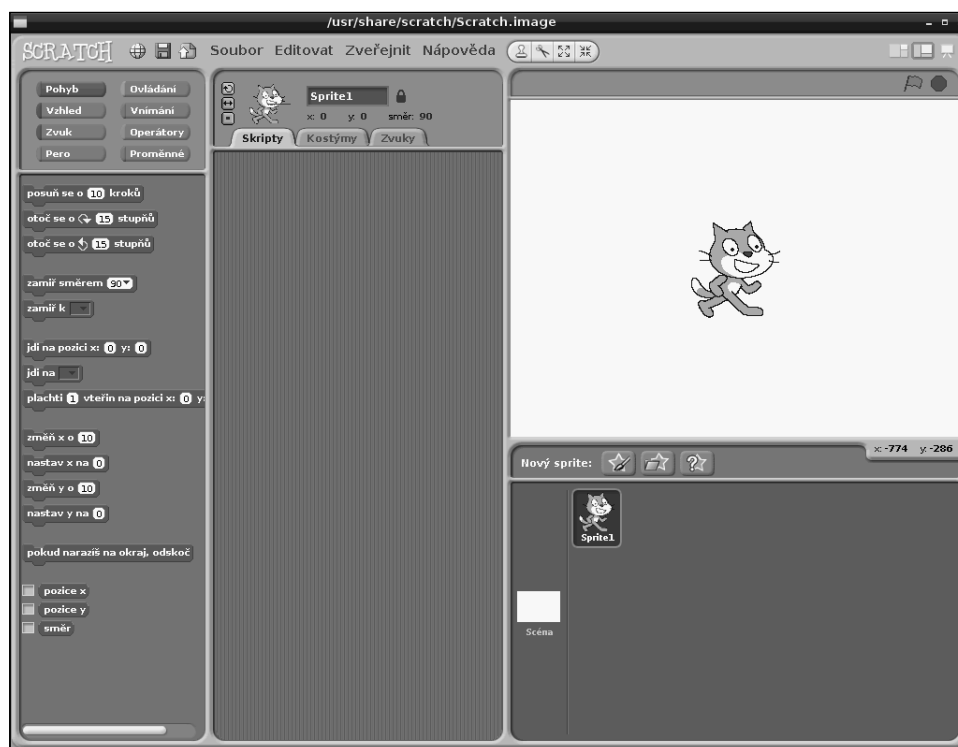
Dokonce i před přechodem na jiný jazyk nabízí jazyk Scratch více než pouhou architekturu na vývoj her. Umožňuje vytvářet interaktivní prezentace a komiksy a díky přidávanému hardwaru, jako jsou robotické sady PicoBoard a LEGO WeDo, také rozhraní s externími senzory a motory.

Doporučená distribuce Debian pro počítač Raspberry Pi standardně obsahuje nejnovější verzi vývojového prostředí Scratch, takže pokud jste postupovali podle pokynů v předchozích částech knihy, můžete hned začít. Jestliže používáte jiný operační systém, navštivte oficiální web jazyka Scratch na adrese <http://scratch.mit.edu>, kde je dostupný software ke stažení. Na stejném místě můžete najít i verze pro systémy Windows a OS X společnosti Apple.

Příklad 1: Ahoj Raspberry

Při seznamování s novým programovacím jazykem se obvykle začíná velmi jednoduchým programem, který vypisuje jeden řádek textu. Obvykle se označuje jako program „Hello World“ (u nás často „Ahoj světe“) a představuje první krok na cestě k tvorbě vlastních programů. Vy vytvoříte alternativní program Ahoj Raspberry.

Oproti tradičním programovacím jazykům neočekává jazyk Scratch, že si bude uživatel pamatovat názvy instrukcí jako `print` či `inkey$`. Místo toho lze téměř vše provést přetažením bloků kódu a jejich uspořádáním do logického schématu.

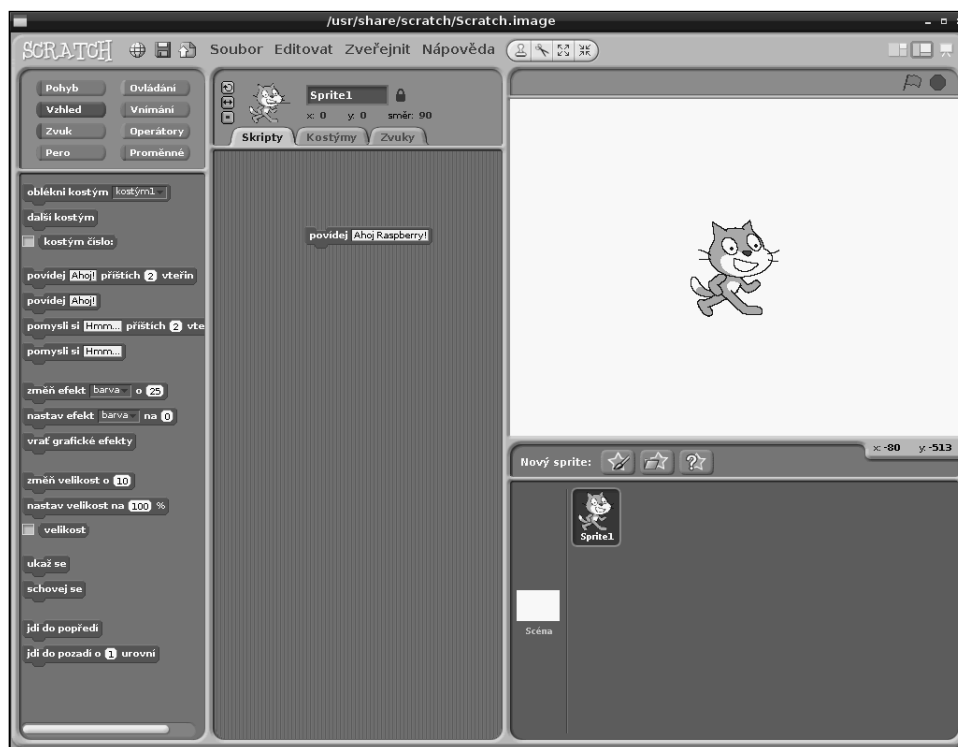


Obrázek 10.1: Hlavní rozhraní Scratch, které je spuštěno v počítači Raspberry Pi

Chcete-li začít, spusťte prostředí Scratch poklepáním na jeho ikonu na ploše nebo klepnutím na příslušnou položku v nabídce Programming (Programování). Po několika sekundách se zobrazí hlavní rozhraní Scratch (viz obrázek 10.1). Pokud okno neleží uprostřed obrazovky nebo je příliš malé, klepněte na tlačítko Maximise (prostřední ze tří tlačítek ovládání okna na pravé straně horního okna), aby okno vyplnilo celou obrazovku.

Rozhraní Scratch se skládá z více podoken. Na levé straně je *paleta dílků* (block palette), kde jsou umístěny všechny dílky kódu, z nichž lze vytvářet programy. Vpravo dole se zobrazuje seznam objektů v programu, kterým se říká *postavy* (sprite), spolu s ovládáním *scény* (stage), na které se postavy objevují. Samotná scéna leží v pravé horní části okna a právě tam se vykresluje spuštěný program. Nakonec prostřední část okna slouží ke stavbě vlastního programu.

Aby mohl uživatel rychle začít, nový projekt Scratch již obsahuje prázdnou scénu a jednu postavu. Zatím však chybí program, takže se po klepnutí na zelenou ikonu praporku v pravé horní části okna nic nestane, protože prostředí Scratch dosud neví, co chcete provést.



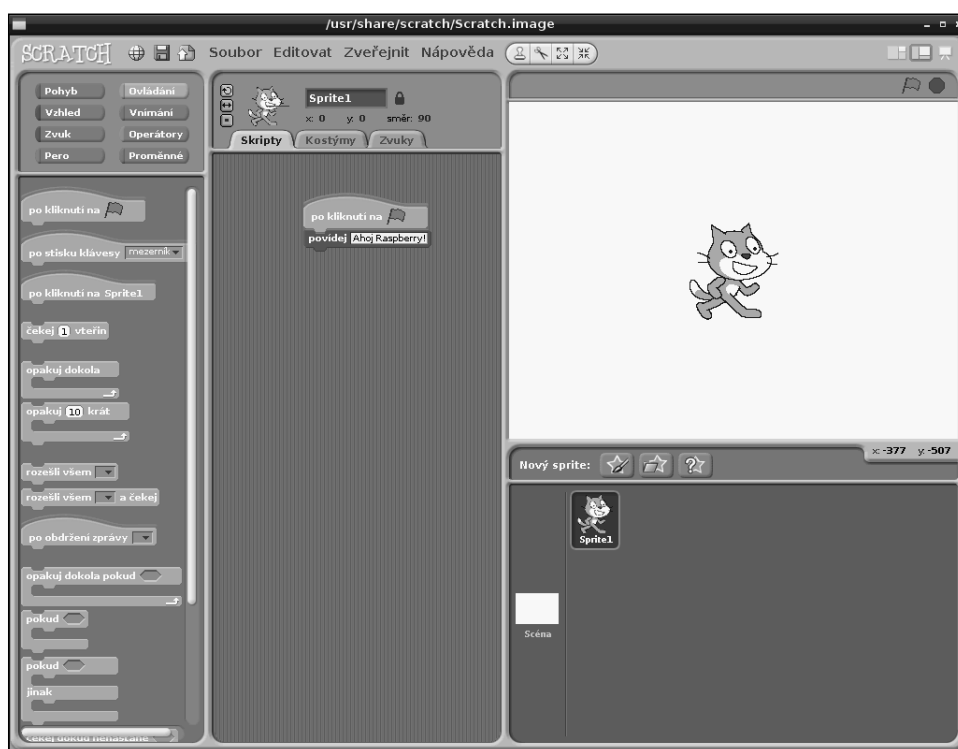
Obrázek 10.2: První dílek umístěný do programu Scratch

U programu Ahoj Raspberry je nutné přepnout paletu dílků na levé straně obrazovky do režimu Looks (Vzhled) klepnutím na jeho tlačítko. Níže v seznamu dílků Looks (Vzhled) se nachází dílek s popisem say Hello! (povídej Ahoj!). Klepněte na tento dílek a přetáhněte jej na prázdné místo uprostřed podokna označeného Skripty (Skripty). Chcete-li se držet tradičního

názvu programu, který vznikl již před desítkami let, můžete také případně klepnout na popis Hello! (Ahoj!) dílku a upravit jej tak, aby obsahoval popis Ahoj Raspberry! (viz obrázek 10.2).

Ani po dalším klepnutí na zelený praporek program nic neudělá. Prostředí Scratch sice ví, že má postava kocoura něco říci, ale zatím neví kdy. Událost vyžaduje *aktivační dílek* (trigger block), který najdete v části Control (Ovládání) palety dílků.

Přejděte do této části klepnutím na tlačítko Control (Ovládání) a poté přetáhněte myší první položku označenou when [flag icon] clicked (po kliknutí na [ikonu praporku]) a umístěte ji těsně nad fialový dílek say (povídej – viz obrázek 10.3). Pokud dílek přetáhnete dostatečně blízko, měl by se automaticky připojit ke stávajícímu dílku jako kousek skládačky.



Obrázek 10.3: Dílek Control (Ovládání) připojený k dílku Looks (Vzhled) v prostředí Scratch

Toto propojování více dílků dohromady představuje základní princip jazyka Scratch. Podíváte-li se na dílek Control (Ovládání), který jste právě umístili, můžete si všimnout, že ve své horní části nemá žádnou spojovací díru. To znamená, že nad něj nemůžete uložit další dílek, protože tento dílek Control (Ovládání) má přímo aktivovat řadu jiných dílků a musí se nacházet na začátku řetězce. Ve své dolní části je však dílek say (povídej) vybaven konektorem, který zapadá do horní části jiných dílků. Z toho lze poznat, že lze pod něj umístit více dílků.

Když jsou dva dílky na svém místě, znovu klepněte na ikonu zeleného praporku v pravém horním rohu obrazovky. Tentokrát se u kocourových úst objeví textová bublina (viz obrázek 10.4) a program Ahoj Raspberry je hotový.

Než se pustíte do dalšího příkladu, nezapomeňte svou práci uložit pomocí nabídky File (Soubor). Prostředí Scratch je navrženo tak, že může být v jednu chvíli otevřen pouze jediný projekt. Jestliže tedy vytvoříte nový prázdný soubor, bude aktuální soubor uzavřen, aby udělal místo pro nový. Nemusíte se však obávat, že na to při přechodu k novému souboru zapomenete. Pokud jsou v existujícím souboru neuložené změny, prostředí Scratch vás před uzavřením souboru vyzve k tomu, abyste jej uložili.



Obrázek 10.4: Program Ahoj Raspberry spuštěný v prostředí Scratch

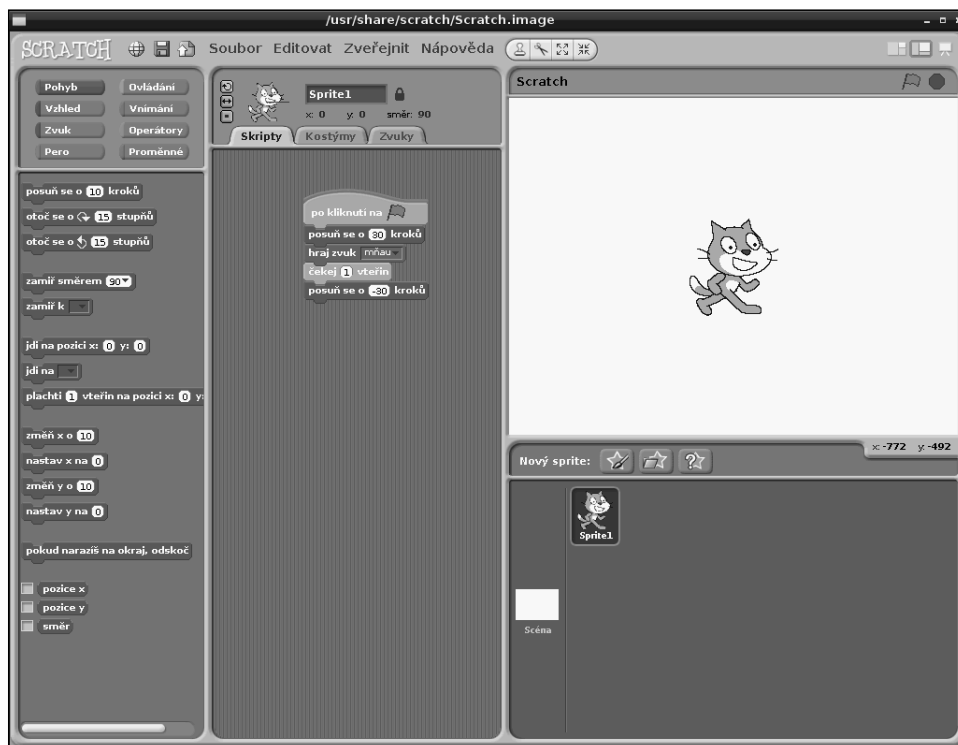
Příklad 2: Animace a zvuk

Program Ahoj Raspberry sice představuje klasický příklad, ale není příliš zajímavý. Neukazuje také skutečné možnosti jazyka Scratch, které leží v jeho působivých multimediálních možnostech a systému ovládání postav. Tento systém se hodí zejména pro jednoduché animace, které mohou tvořit základ interaktivní hry.

Nejdříve začněte nový projekt jazyka Scratch. Přitom můžete buď znovu spustit celé prostředí, nebo zvolit příkaz New (Nový) z nabídky File (Soubor). Jako u každého nového projektu poskytne prostředí Scratch výchozí postavu. Jedná se o postavu, kterou budete řídit.

Jednoduchou animaci v prostředí Scratch lze ovládat pomocí části Motion (Pohyb) palety dílků. Při spuštění nového projektu je tato paleta nastavena jako výchozí. Přetáhněte dílek označený *move 10 steps* (posuň se o 10 kroků) do oblasti Scripts (Skripty). Jak je zřejmé z jeho názvu, tento dílek požádá vybranou postavu, aby se posunula o 10 kroků ve směru, kam je aktuálně otočena. Výchozí postava jazyka Scratch je vždy orientována přímo doprava. Dílek *move 10 steps* (posuň se o 10 kroků) tedy zajistí posun postavy o 10 kroků doprava.

Deset kroků je poměrně málo. Klepněte tedy na hodnotu 10 a změňte ji na 30. Popis dílku by se měl změnit na *move 30 steps* (posuň se o 30 kroků). Animace kocoura, který se pouze pohybuje po scéně zleva doprava, však není příliš zajímavá. Přepněte tedy na paletu dílků Sound (Zvuk) a přetáhněte dílek *play sound meow* (hraj zvuk mňau) do oblasti Scripts (Skripty) a připojte jej pod existující dílek *move* (posuň se). Chcete-li kocoura chvíli ponechat v této pozici, přidejte z palety dílků Control (Ovládání) dílek *wait 1 secs* (čekej 1 vteřin). Bez tohoto dílku by postava kocoura rychle přeskočila ze své počáteční pozice do cílové.



Obrázek 10.5: Dokončený program animace v jazyce Scratch

Pokud chcete nastavit, aby bylo možné program spustit vícekrát a postava kocoura přitom nezmizela za okraj scény, doplňte další dílek `move 10 steps` (posuň se o 10 kroků) pod dílek `play sound` (hraj zvuk) a upravte jej tak, aby měl popis `move -30 steps` (posuň se o -30 kroků). Prostředí Scratch bez problémů umožňuje zadávat taková záporná čísla: jestliže hodnota 30 způsobí, že se postava posune o určitou vzdálenost doprava, pak hodnota -30 zajistí posun postavy o přesně stejnou vzdálenost, ale v opačném směru.

Nakonec z palety dílků Control (Ovládání) přidejte do horní části oblasti Scripts (Skripty) dílek `when [flag icon] clicked` (po kliknutí na [ikonu praporku]). Dokončený program je znázorněn na obrázku 10.5. Program lze spustit klepnutím na ikonu zeleného praporku v pravém horním rohu okna. Kvůli silnějšímu efektu nezapomeňte k počítači Pi připojit sluchátka nebo reproduktory!

Tento jednoduchý animační program je možné různými způsoby rozšiřovat. Pomocí možnosti New Sprite (Nový sprite) přímo pod scénou na pravé straně okna prostředí Scratch může programátor přidávat další postavy, které se dokáží pohybovat a přehrávat zvuky nezávisle na sobě. Přidáte-li dílek `say` (povídej) z prvního příkladu (nebo podobný dílek `think` (pomysli si), který místo mluvené bubliny vytvoří myšlenou bublinu), můžete vytvořit animovaný komiks.



Obrázek 10.6: Přidání nekonečného cyklu do jednoduché animace jazyka Scratch

Důležitější je, že dokonce i tento jednoduchý příklad učí důležité programátorské principy: navzdory své délce pouhých pěti dílků zahrnuje pohyb postavy v kladných i záporných souřadnicích, přehrávání zvuku a koncepci *čekání* (delay) v programu. Pokud se nebojíte, že se zblázníte z opakovaně přehrávaného zvuku, můžete vyzkoušet další koncepci a přidat z palety Control (Ovládání) dílek *forever* (opakuj dokoła – viz obrázek 10.6). Tento dílek do programu dodá *cyklus* (loop), takže program bude seznamem dílků procházet stále znovu a znovu – alespoň do té doby, než zvuk začne být nesnesitelný a klepnete na červené tlačítko stop v horní části scény!

Příklad 3: Jednoduchá hra

V jazyce Scratch lze sice vytvářet jednoduché animace, ale software umožňuje i tvorbu interaktivních programů, které reagují na vstup z klávesnice. Když výše popsaný program doplníte o několik jednoduchých animačních ovladačů, můžete získat jednoduchou hru – a zároveň si osvojit princip *kolize postav* (sprite collision), *příkazů if a vstupu*.

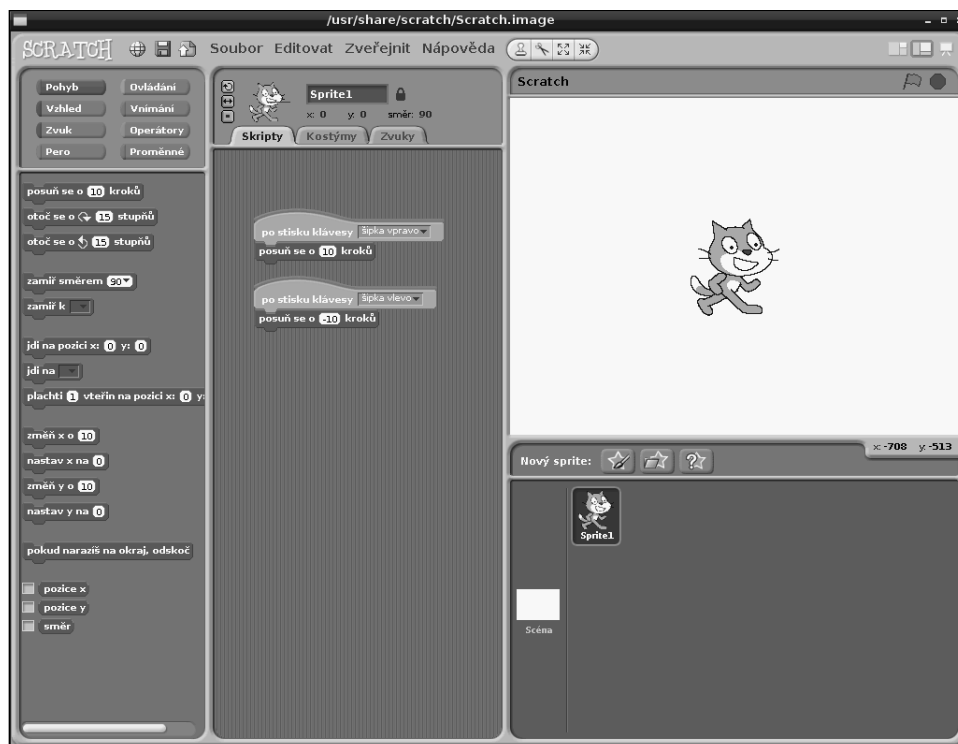
V tomto příkladu začnete nový projekt jazyka Scratch (pokud jste to zatím neudělali, nezapomeňte uložit předchozí příklad). Nejdříve do oblasti Scripts (Skripty) přetáhněte dílek *move 10 steps* (posuň se o 10 kroků). Místo toho, abyste nastavili spuštění dílků kódu při klepnutí na ikonu praporku, přejděte tentokrát do palety dílků Control (Ovládání) a přetáhněte dílek *when space key pressed* (po stisku klávesy mezerník) nad dílek *move* (posuň se).

Jak je zřejmé z jeho názvu, dílek *when space key pressed* (po stisku klávesy mezerník) sleduje vstup od uživatele – v tomto případě stisknutý mezerník – a používá jej jako aktivátor spuštění seznamu dílků. Dílek funguje kdykoli – jestliže stisknete mezerník ihned, postava splní pokyn a posune se o 10 kroků doprava.

Dílek *when space key pressed* (po stisku klávesy mezerník) se také od dílku *when [flag icon] clicked* (po kliknutí na [ikonu praporku]) liší dalším důležitým způsobem: lze jej upravit. Po klepnutí na šipku dolů vedle slova *space* (mezerník) se zobrazí seznam všech klíčů, které může dílek monitorovat. Ze seznamu poté vyberte položku *right arrow* (šipka vpravo). Dílek se tím změní na *when right arrow key pressed* (po stisku klávesy šipka vpravo).

Hra, ve které se hráč může pohybovat pouze v jednom směru, není příliš zábavná, takže do oblasti Scripts (Skripty) přetáhněte nový dílek *when space key pressed* (po stisku klávesy mezerník). Tento dílek nelze propojit s existujícím seznamem dílků, protože aktivací dílek může být pouze jeden. Začněte tedy nový seznam o něco níže. Stejně jako předtím použijte tlačítko se šipkou dolů vedle slova *space* (mezerník), abyste dílek přizpůsobili. Změňte jej na dílek *when left arrow key pressed* (po stisknutí klávesy šipka vlevo). Nakonec přepněte paletu dílků zpět do režimu Motion (Pohyb) a připojte dílek *move 10 steps* (posuň se o 10 kroků) pod nový dílek *when left arrow key pressed* (po stisknutí klávesy mezerník). Potom změňte popis dílku na *move -10 steps* (posuň se o -10 kroků).

Stisknete-li nyní levou a pravou šipku, zjistíte, že se kocour pohybuje podle vstupu z klávesnice (viz obrázek 10.7): při stisknutí šipky doleva se kocour posune o 10 kroků vlevo (ačkoli z hlediska jazyka Scratch se jedná o posun o minus 10 kroků vpravo) a při stisknutí šipky vpravo se kočka přemístí o 10 kroků vpravo.



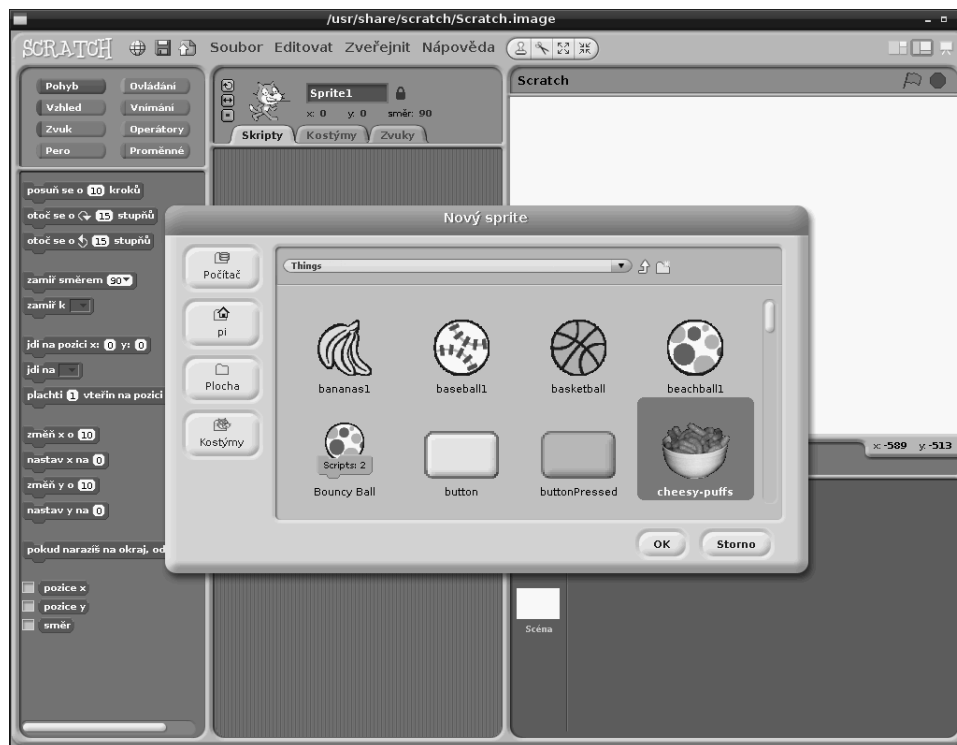
Obrázek 10.7: Ovládání pohybu pohyblivého prvku jazyka Scratch pomocí vstupních dílků

Když nyní víte, jak může hráč ovládat postavu, je potřeba přiřadit jí nějakou funkci. Protože se jedná o velmi jednoduchou hru, bude cíl také prostý: najít nějaké jídlo. Klepněte na tlačítko Choose New Sprite From File (Výběr nového sprite ze souboru), které je prostřední ze tří tlačítek nad seznamem postav v pravé dolní části okna Scratch.

Zobrazí se dialogové okno s pokynem, abyste vybrali postavu: poklepejte na složku Things (Věci) a poté poklepejte na pohyblivý postavu Cheesy-Puffs (kukuřičné křupky se sýrem – viz obrázek 10.8). Postava se objeví jako nová položka seznamu postav a získáte tak nový objekt, který můžete ve hře kontrolovat.



Tip: Jazyk Scratch přirozeně podporuje *multithreading* a částečně je *objektově orientován*. To znamená, že ke každému objektu v programu (včetně postav) lze připojit vlastní kód a všechny dílky kódu mohou fungovat souběžně a nezávisle na jiných dílcích. Když tyto možnosti vhodně uplatníte, můžete díky nim vytvářet poměrně složité programy.



Obrázek 10.8: Přidání postavy Cheesy-Puffs do projektu hry jazyka Scratch

Každá postava přidaná do projektu Scratch se standardně zobrazí přesně uprostřed scény. Z tohoto důvodu zakryje stávající postavu kocoura. Chcete-li tento problém vyřešit, klepněte na novou postavu a přetáhněte ji doprava od kocoura.

Přesunutá mísa kukuřičných křupek je v poměru k antropomorfnímu dvounohému kocourovi stále příliš velká. Klepněte tedy na tlačítko Shrink Sprite (Zmenšit sprite), které je umístěno v pravé horní části oblasti scény a zobrazuje čtyři šipky, které míří dovnitř. Jestliže si nejste jisti, o které tlačítko se jedná, můžete umístit ukazatel myši nad jednotlivé ikony a zobrazí se stručný popis jejich účelu.

Klepnete-li na tlačítko Shrink Sprite (Zmenšit sprite) nebo na tlačítko Grow Sprite (Zvětšit sprite), které má přesně opačnou funkci, ukazatel myši se změní na kopii ikony tlačítka. Pomocí tohoto nového ukazatele klepněte na postavu Cheesy-Puffs, abyste ji zmenšili. Při každém následujícím klepnutí se bude postava dále zmenšovat. Jakmile její velikost bude vypadat přiměřeně, klepněte na kterékoli místo mimo oblast scény, abyste obnovili normální podobu ukazatele myši. Chcete-li, můžete pak mísu přesunout blíže k pravému okraji scény.

Nyní se pokuste přesunout postavu kocoura směrem k postavě Cheesy-Puffs. Jak se sami přesvědčíte, při setkání obou postav se nic nestane. To proto, že program neobsahuje žádné poky-

ny, co má dělat při překrytí dvou postav, které se označuje jako *kolize postav*. Nprovede tedy žádnou akci. To lze napravit pomocí nového typu dílku: dílku *Sensing* (Vnímání).

Pokud je postava Cheesy-Puffs aktivní, měl by se její obrázek objevit v horní části podokna Scripts (Skripty). V opačném případě stačí poklepat na prvek na scéně. Při aktivní postavě Cheesy-Puffs přepněte paletu dílků do režimu Sensing (Vnímání) klepnutím na stejnojmenné tlačítko. Z palety Sensing (Vnímání) přetáhněte do podokna Scripts (Skripty) dílek *touching ?* (dotýká se ?).

Stejně jako dílek *when space key pressed* (po stisku klávesy mezerník), kterým jste kontrolovali pohyb postavy kocoura, lze i dílek *touching ?* (dotýká se ?) upravit. Klepněte na tlačítko se šipkou dolů vedle otazníku a ze seznamu zvolte položku *Sprite1* (postava kocoura). Tento dílek se nyní bude aktivovat při kolizi dvou postav.



Tip: Chcete-li postavu pojmenovat, klepněte v podokně Scripts (Skripty) na rámeček vedle obrázku postavy a zadejte název. Když postavám dáte srozumitelná jména (například **Kocour**, **Kukuřičné-Křupky** atd.) bude později mnohem snazší sledovat, co se v programu děje.

Podívejte se na tvar dílku *touching Sprite1?* (dotýká se Sprite1?). Je zřejmé, že nahoře ani dole nemá žádné propojovací výstupky a svým tvarem připomíná diamant. Stejný tvar slouží ve vývojových diagramech jako rozhodovací bod. To není náhoda: většina dílků Sensing (Vnímání) funguje teprve po svém vložení do dílku Control (Ovládání).

Přepněte paletu dílků do režimu Control (Ovládání) a vyhledejte dílek *if (pokud)* – vypadá jako rozmáčklé a hrbolaté písmeno C. Všimněte si, že dílek *if (pokud)* má výřez ve tvaru diamantu, což je tvar dílku *touching Sprite1?* (dotýká se Sprite1?). Přetáhněte dílek *if (pokud)* do podokna Scripts (Skripty) a poté přetáhněte dílek *touching Sprite1?* (dotýká se Sprite1?) na jeho výřez ve tvaru diamantu. Získáte dvoubarevný dílek, který má aktuálně popis *if touching Sprite1?* (dotýká se Sprite1?).

Tento kombinovaný dílek zastupuje v programu *podmínku*: po dosažení tohoto bodu programu je veškerý kód umístěný v rámci podmínky proveden tehdy a pouze tehdy, je-li *podmínka* splněna. V tomto případě se jedná o podmínku, že se postava *Sprite2* dotkla postava *Sprite1*. Pomocí logických dílků *and* (a), *or* (nebo) a *not* (není) z palety dílků Operators (Operátory) lze realizovat poměrně složité scénáře.

Z palety dílků Looks (Vzhled) přetáhněte dílek *say Hello! For 2 secs* (povídej Ahoj! příštích 2 vteřin) doprostřed podmínky *if touching Sprite1?* (pokud se dotýká Sprite1?). Změňte text na *Nejez!* a poté přidejte dílek Control (Ovládání) s názvem *wait 1 secs* (čekej 1 vteřin) a změňte hodnotu na 2. Nahoru doplňte dílek *when space key pressed* (po stisku klávesy mezerník) a upravte jeho popis na *when right arrow key pressed* (po stisku klávesy šipka vpravo). Nakonec na konec cyklu přetáhněte dílek *hide* (schovej se) z palety Looks (Vzhled). Dostanete seznam dílků, který je znázorněn na obrázku 10.9.



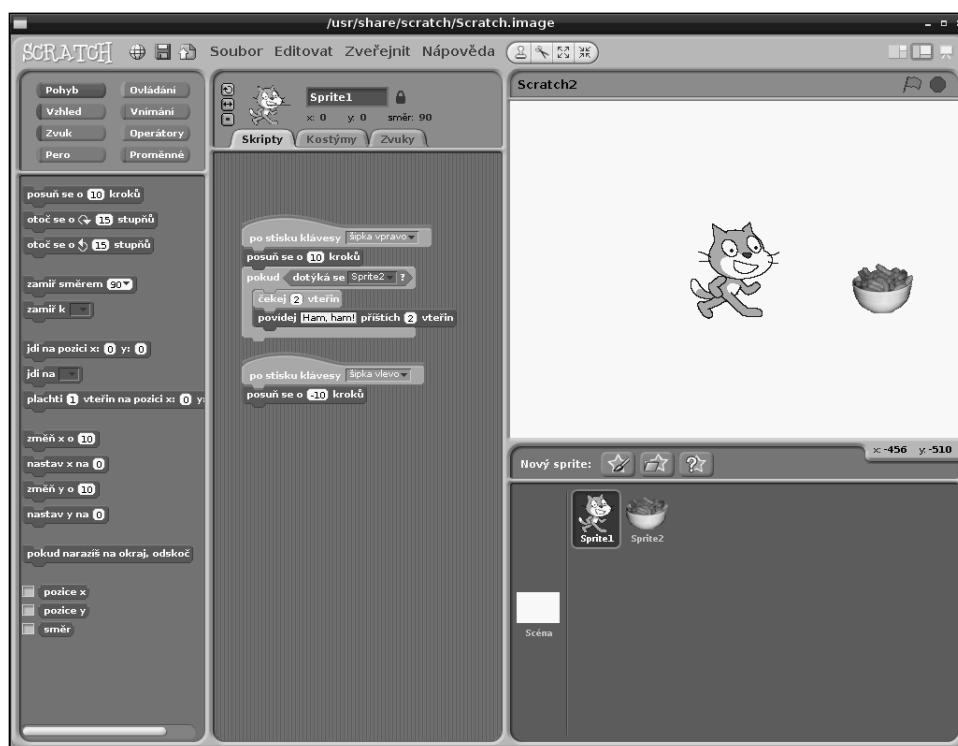
Booleovská logika či booleovská algebra, která svůj název získala podle George Booleho, představuje klíčový princip fungování počítačů. V jazyku Scratch je booleovská logika implementována ve třech dílcích Operators (Operátory): **and** (a), **or** (nebo) a **not** (není).

Operátor **or** (**nebo**) vyžaduje, aby byl pravdivý některý z jeho dvou vstupů. Je-li pravdivý alespoň jeden vstup, bude pravdivý i výstup operátoru. Nabízí se tím praktický způsob opakovaného použití kódu: máte-li více postav, kterých se postava hráče nesmí dotknout, stačí použít jediný dílek kódu s operátorem **or** (**nebo**), který se aktivuje při kontaktu s libovolnou nepřátelskou postavou.

Poslední operátor **not** (**není**) se označuje jako *invertor*: ať už má jakýkoli vstup, jeho výstup je vždy opačný. Jestliže je jeho vstup nepravdivý, výstup je pravdivý. Naopak platí, že pokud je vstup pravdivý, výstup je nepravdivý.

Poklepáním na postavu kocoura na scéně přejdete zpět k úpravám jeho skriptů. Skript vytvořený pro postavu Cheesy-Puffs zmizí, ale nelekejte se. Zůstává zachován, ale znovu se objeví teprve tehdy, začnete-li upravovat příslušnou postavu.

Přetáhněte další dílek `if` (pokud) z palety Control (Ovládání) spolu s dalším detekčním dílkem `touching ?` (dotýká se ?). Tentokrát změňte dílek Sensing (Vnímání) tak, aby oba dílky společně měly popis `if touching Sprite2?` (pokud dotýká se Sprite2?). Do tohoto dílku vložte dílek Control (Ovládání) `wait 1 secs` (čekej 1 vteřin), jehož hodnotu změňte na 2, a dílek Looks (Vzhled) `say Hello! for 2 secs` (povídej Ahoj! příštích 2 vteřin), kde původní text nahraďte textem `Ham, ham!`. Nakonec přetáhněte celou sadu dílků nahoru, abyste ji propojili s dolní částí existujícího dílku `when right arrow key pressed` (po stisku klávesy šípka vpravo) pod dílkem `move 10 steps` (posuň se o 10 kroků). Hotový kód pro postavu kocoura by měl odpovídat dílkům na obrázku 10.10.



Obrázek 10.10: Dokončený kód jednoduché hry jazyka Scratch s pohyblivým prvkem kočky

Když pomocí klávesy se šipkou doprava posunete kocoura směrem k postavě Cheesy-Puffs, hra začíná. Když se kocour dostane k postavě Cheesy-Puffs, proběhne rozhovor a mísa křupek by měla zmizet.

Tento příklad sice efektivně představuje několik důležitých programátorských přístupů, ale rozhodně nejde o optimální způsob, jak by takovou hru bylo možné naprogramovat. Jazyk

Scratch obsahuje systém *vysílání zpráv* (message broadcast), který kódu připojenému k jednomu objektu umožňuje komunikovat s kódem připojeným k jinému objektu. Díky tomu lze dosahovat mnohem elegantnějších výsledků kolizí, které nemusí spoléhat na pečlivě načasované pauzy, aby dávaly smysl.

Chcete-li experimentovat s vysíláním zpráv, vyzkoušejte dílky broadcast (rozešli všem) a when I receive (po obdržení zprávy) z palety Control (Ovládání). Zpráva vytvořená pro dílek vysílání v libovolném objektu může aktivovat kód v kterémkoli jiném objektu pomocí příznaku when I receive (po obdržení zprávy). To znamená, že tímto způsobem lze vzájemně propojit více objektů a jejich kód.

Robotika a senzory

Kromě čistě softwarových vstupů a výstupů je také možné navázat programy jazyka Scratch na vnější hardware pomocí senzorkých desek a robotických systémů. Díky své multithreadingové povaze a výkonnému systému zasílání zpráv mezi procesy lze pomocí jazyka Scratch vytvořit překvapivě pokročilý robotický modul. Jedná se o zdaleka nejjednodušší způsob, jak z počítače Raspberry Pi udělat centrum jednoduchého robota.

Na rozdíl od programovacího jazyka Python, který je sice výkonnější, ale značně složitější, nedokáže jazyk Scratch komunikovat s externím hardwarem pomocí portu GPIO počítače Raspberry Pi. Aby mohl interagovat s fyzickým světem, je tedy nejdříve nutné jej doplnit o dodatečné zařízení.

Detekce pomocí sady PicoBoard

Sada PicoBoard, kterou navrhl výrovec elektroniky SparkFun, je rozšiřující modul, který se připojuje k počítači s prostředím Scratch. Poskytuje proměnlivý vstup ovládaný jezdcem, světelný senzor, tlačítko, zvukový senzor a čtyři páry krokosvorek, které umožňují monitorovat externí elektrická zařízení.

V současnosti jsou k dispozici dva typy sad PicoBoard: původní model, který používá sériové připojení, a jeho nástupce, který pracuje s připojením USB. Druhý model je kompatibilní s počítačem Raspberry Pi. Stačí jej připojit do volného portu USB nebo do portu rozbočovače USB připojeného k počítači Pi a restartovat prostředí Scratch. Poté lze využívat různé jeho funkce přímo z rozhraní Scratch.

Podrobnosti o sadě PicoBoard naleznete na adrese <http://www.sparkfun.com/products/10311>.

Robotika se sadou LEGO

Kromě doplňkových modulů se senzory je možné pomocí jazyka Scratch řídit i externí robotický systém. Robotická sada LEGO Education WeDo umožňuje připojit k počítači motory,

senzory vzdálenosti a senzory rychlosti. Všechny tyto prvky lze přitom řídit pomocí specializovaných dílků, které jsou dostupné v prostředí Scratch.

Dětem sada WeDo nabízí vynikající úvod do robotiky: součásti se kombinují stejně snadno jako běžné dílky LEGO a ani při nesprávném zapojení nehrozí riziko elektrického šoku. Sada WeDo je kompatibilní s běžnými sadami LEGO a LEGO Technic, takže je možné rychle a snadno budovat velké projekty.

Další informace o použití sady LEGO Education WeDo s jazykem Scratch naleznete na stránce <http://info.scratch.mit.edu/WeDo>.

Další zdroje informací

Tato kapitola sice slouží jako stručný úvod do jazyka Scratch, ale ani zdaleka toto téma nevyčerpává. Pro mladší čtenáře, kteří se obvykle učí snáze, když jejich lekce doprovází hodně barevných obrázků, je také poněkud popisná.

Sekce Support (Podpora) oficiálního webu jazyka Scratch, které provozuje univerzita MIT na adrese <http://info.scratch.mit.edu/support>, obsahuje odkaz na příručku pro začátečníky (Getting Started Guide) ve formátu PDF. Pestrá příručka vysvětluje práci s jazykem Scratch způsobem, který odpovídá dětskému čtenáři. Jedná se o skvělý výukový nástroj. Učení lze dále oživit díky kombinaci s kartami Scratch Cards, což jsou kartičky ke stažení, které obsahují definice jednotlivých typů dílků jazyka Scratch.

Univerzita MIT také provozuje fórum uživatelů jazyka Scratch, kde se mohou nadšenci učit od zkušenějších kolegů a sdílet řešení běžných problémů. Registrace je bezplatná a web dobře funguje i v prohlížeči Midori, který je dostupný v distribuci Debian pro počítač Raspberry Pi. Fóra jsou k dispozici na adrese <http://scratch.mit.edu/forums/>.

Chcete-li se však zdokonalit v jazyku Scratch, je nejjednodušší prostě si začít hrát. Název Scratch odkazuje na používání gramofonu: když diskžokej otáčí desku, jehla přejede přes drážky a ozve se zvuk škrábnutí. Tak jako diskžokejové mixují z hotových písniček nové skladby, nadšení uživatelé jazyka Scratch mohou posílat své výtvořky na oficiální web, kde si je jiní uživatelé mohou stáhnout, prozkoumat, upravit a kombinovat. Na oficiálním webu jazyka Scratch je v současnosti uloženo více než 2,5 milionu programů. Jedná se proto o dokonalý výukový zdroj, který umožňuje poznat, jak lze v jazyce Scratch vytvářet projekty, a sdílet vlastní nápady s ostatními. Seznam nejnovějších sdílených projektů je k dispozici na adrese <http://scratch.mit.edu/latest/shared>.

Úvod do jazyka Python

První část názvu počítače Raspberry Pi vychází z dlouhodobé tradice pojmenovávání počítačových systémů podle ovoce – od klasických mikropočítačů jako Acorn (žalud), Apricot (meruňka) a Tangerine (mandarinka) po v současnosti známější značky včetně Apple (jablko) a BlackBerry (ostružina). Druhá polovina názvu však odkazuje na programovací jazyk *Python*.

Představení jazyka Python

Pružný a silný jazyk Python vznikl koncem 80. let v institutu National Research Institute for Mathematics and Computer Science a jeho autor Guido van Rossum vycházel z jazyka ABC. Od svého zveřejnění jazyk Python neustále získává na oblibě díky jasné a expresivní syntaxi, při jejímž vývoji byl kladen důraz na čitelnost kódu.

Python je *vysokoúrovňový jazyk*. To znamená, že kód jazyka Python se skládá ze srozumitelných anglických slov a umožňuje počítač Pi programovat způsobem, který se lze rychle naučit a kterému je možno snadno porozumět. To představuje zásadní rozdíl oproti *nízkoúrovňovým jazykům* typu assembleru (jazyka symbolických adres), které se sice blíží stylu „uvažování“ počítače, ale pro člověka bez dostatečných zkušeností jsou téměř nesrozumitelné. Díky své vysokoúrovňové povaze a jasné syntaxi je jazyk Python cenným nástrojem pro každého, kdo se chce naučit programovat. Tento jazyk rovněž doporučuje nadace Raspberry Pi Foundation těm, kdo chtějí od jednoduchého jazyka Scratch (popsaného v kapitole 10, „Úvod do jazyka Scratch“) postoupit k „praktičtějšímu“ programování.

Jazyk Python je publikován pod licencí open source a je volně dostupný pro operační systémy Linux, OS X i Windows. Díky této multiplatformní podpoře lze software napsaný v jazyku Python používat také v počítačích s jinými operačními systémy – s výjimkou případů, kdy program pracuje s hardwarem specifickým pro počítač Pi, jako je port GPIO. Postup adresování tohoto portu pomocí jazyka Python naleznete v kapitole 12, „Hackování hardwaru“.

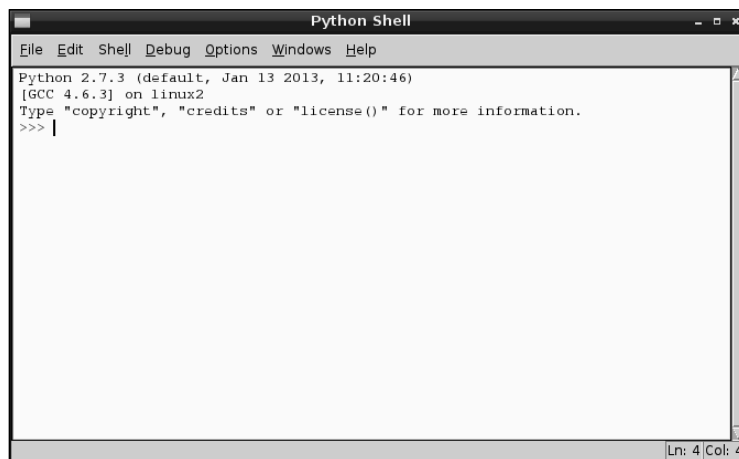
Příklad 1: Ahoj Raspberry

Jak jste se již dozvěděli v kapitole 10, „Úvod do jazyka Scratch“, při učení programovacího jazyka se obvykle začíná projektem, který na obrazovku vypíše zprávu „Hello World!“. V jazyce

Scratch stačilo pouze přetáhnout bloky předem napsaného kódu, ale v jazyce Python je nutné celý program napsat ručně.

Projekt jazyka Python v podstatě není nic jiného než textový soubor, který obsahuje písemné instrukce, podle nichž se má počítač řídit. Takový soubor lze vytvořit v libovolném textovém editoru. Pokud například dáváte přednost práci v konzole nebo okně terminálu, můžete pracovat v programu nano. Jestliže máte raději grafické uživatelské rozhraní (GUI), můžete spustit editor Leafpad. Další variantu představuje integrované vývojové prostředí (IDE), jako je IDLE. Tato prostředí poskytují funkce specifické pro jazyk Python, které ve standardním textovém editoru nejsou k dispozici. Jedná se například o kontrolu syntaxe, funkce ladění a možnost spouštět program bez nutnosti přecházet z editoru do jiného okna. V této kapitole se naučíte, jak vytvářet soubory jazyka Python v prostředí IDLE, ale můžete si samozřejmě zvolit integrované prostředí, jaké vám nejvíce vyhovuje. V této kapitole naleznete také pokyny, jak vytvořené soubory spouštět přímo z okna terminálu. Tyto postupy se týkají všech textových editorů a prostředí IDE.

Chcete-li začít s projektem Hello World, spusťte prostředí IDLE z nabídky Programming (Programování) na ploše distribuce Debian. Jestliže prostředí IDLE nepoužíváte, vytvořte prázdný dokument ve svém oblíbeném textovém editoru a přeskočte zbytek tohoto odstavce. Prostředí IDLE se standardně otevírá v režimu *příkazového interpretu Python* (Python shell – viz obrázek 11.1). Všechny příkazy zadané do výchozího okna budou tedy okamžitě spuštěny. Chcete-li otevřít nový projekt jazyka Python pro pozdější spuštění, klepněte v nabídce File na příkaz New Window. Otevře se prázdný soubor.



Obrázek 11.1: Okno příkazového interpretu Python v prostředí IDLE



Tip: Když z nabídky Programming (Programování) zvolíte místo položky IDLE položku IDLE 3, spustíte jazyk Python ve verzi 3.0. Tato verze jazyka Python obsahuje funkce, které ve starší verzi jazyka v prostředí IDLE nejsou dostupné. V této kapitole se však z žádnými takovými funkcemi nesetkáte. Příklady z této kapitoly budou fungovat nezávisle na tom, kterou verzi spustíte.

Doporučuje se začínat všechny programy jazyka Python řádkem označovaným jako *shebang* (vysl. „šíbeng“), který dostal svůj název podle znaků `#` a `!` na začátku řádku. Z tohoto řádku operační systém zjistí, kde má hledat soubory jazyka Python. Řádek sice není nezbytný u programů, které budou spouštěny v rámci prostředí IDLE nebo budou explicitně volat jazyk Python v okně terminálu, ale *povinný* je v případě programů, které se spouštějí přímo voláním názvu příslušného souboru.

Chcete-li zajistit, že budou programy fungovat bez ohledu na umístění spustitelného souboru jazyka Python, měl by první řádek programů vypadat takto:

```
#!/usr/bin/env python
```

Tento řádek sdělí operačnímu systému, že má zjistit umístění jazyka Python v proměnné prostředí `$PATH` (kam operační systém Linux ukládá umístění souborů, které lze spouštět jako programy). Tento zápis by měl být platný v každé linuxové distribuci pro počítač Pi. Proměnná `$PATH` obsahuje seznam adresářů, kde jsou uloženy spustitelné soubory, a umožňuje vyhledat programy, jejichž název zadáte v konzole nebo okně terminálu.

Chcete-li vypsát zprávu, použijte příkaz `print` jazyka Python. Jak vyplývá z jeho názvu, tento příkaz vypíše text na výstupní zařízení, což je ve výchozím nastavení konzola nebo okno terminálu, odkud je program spuštěn. Příkaz se používá jednoduše: na standardní výstupní zařízení bude vypsán libovolný text, který následuje za slovem `print` a je uzavřen do uvozovek. Do svého nového projektu zadejte následující řádek:

```
print "Ahoj Raspberry!"
```

Hotový program by měl vypadat takto:

```
#!/usr/bin/env python
print "Ahoj Raspberry!"
```

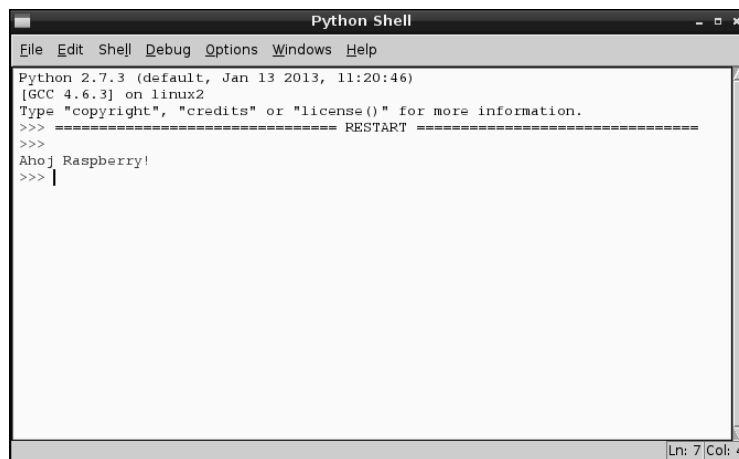
Vytváříte-li ukázkový program místo textového editoru v prostředí IDLE, můžete si všimnout, že text je zvýrazněn různými barvami (viz obrázek 11.2, kde jsou při černobílém tisku barvy patrné jako různé odstíny šedi). Tato vlastnost se označuje jako *zvýrazňování syntaxe* (syntax highlighting) a jedná se o funkci prostředí IDE a pokročilejších nástrojů na úpravy textu. Při zvýrazňování syntaxe se mění barva částí textu podle jejich funkce, aby byl program na první pohled lépe srozumitelný. Uspodňuje také odhalení takzvaných *syntaktických chyb*, které spočívají například v zapomenutých ukončovacích uvozovkách příkazu `print` nebo chybějícím znaku komentáře. V tomto krátkém příkladu není zvýrazňování syntaxe nutné, ale u větších programů se může jednat o neocenitelný nástroj na hledání chyb.

Chcete-li program spustit, uložte jej nejdříve pomocí nabídky File jako soubor `ahoj.py`. V prostředí IDLE dostane soubor příponu `.py` automaticky. Pokud používáte textový editor, nezapomeňte při ukládání zadat na konec názvu souboru příponu `.py` (nikoli `.txt`). Tato přípona určuje, že soubor obsahuje kód jazyka Python – ačkoli prostředí Python je natolik propracované, že program spustí i v případě, že je uložen s jinou příponou názvu souboru.



Obrázek 11.2: Zvýrazňování syntaxe v prostředí IDLE

Postup spouštění souborů závisí na tom, zda pracujete v prostředí IDLE nebo v textovém editoru. V prostředí IDLE stačí z nabídky Run zvolit příkaz Run Module nebo stisknout klávesu F5 na klávesnici. Prostředí IDLE tím přejde zpět do okna příkazového interpretu Python a spustí program. Na obrazovce by se měla modrými písmeny vypsát zpráva Ahoj Raspberry! (viz obrázek 11.3). Jestliže se nezobrazí, zkontrolujte syntaxi – zejména ověřte, zda jste na řádku s příkazem print zadali uvozovky na začátku i na konci zprávy.



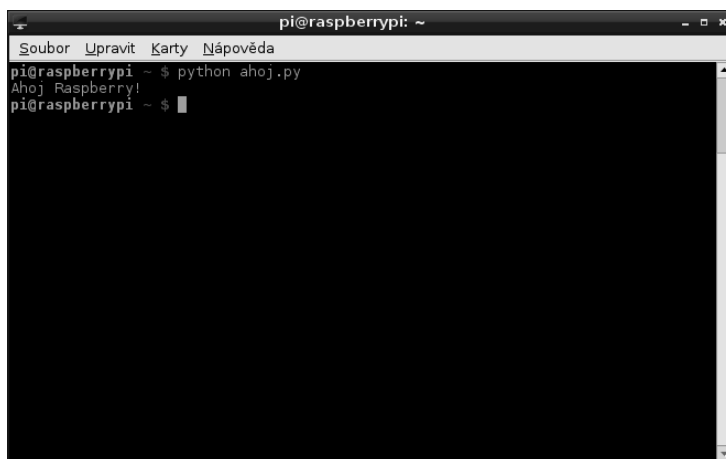
Obrázek 11.3: Spuštění souboru ahoj.py v prostředí IDLE

Pokud jste program ahoj.py vytvořili v textovém editoru, musíte otevřít okno terminálu z nabídky Accessories (Příslušenství) na ploše. V případě, že jste soubor uložili kamkoli jinam než do svého domovského adresáře, musíte také přejít do příslušného adresáře příkazem cd (viz

kapitola 2, „Správa systému Linux“). Jakmile se nacházíte ve správném adresáři, můžete program spustit následujícím příkazem:

```
python ahoj.py
```

Příkaz požádá operační systém, aby spustil prostředí Python a poté načtl soubor `ahoj.py` a provedl jej. Na rozdíl od příkazového interpretu Python v prostředí IDLE se prostředí Python po dosažení konce souboru ukončí a vrátí řízení zpět oknu terminálu. Výsledek je však stejný: na standardní výstup je vypsaná zpráva Ahoj Raspberry! (viz obrázek 11.4).



Obrázek 11.4: Spuštění souboru `ahoj.py` v okně terminálu

Nastavení programů jazyka Python jako spustitelných souborů

Obvykle musíte programy jazyka Python spouštět tak, že o otevření souboru požádáte software Python. Díky řádku shebang na začátku souboru lze však soubor spouštět přímo a není nutné nejdříve volat prostředí Python. To může být užitečné, vytváříte-li vlastní nástroje, které chcete spouštět z okna terminálu: poté, co program jazyka Python zkopírujete do umístění uvedeného v systémové proměnné prostředí `$PATH`, můžete jej volat přímo zadáním názvu jeho souboru.

Nejdříve musíte operačnímu systému Linux sdělit, že soubor jazyka Python by měl být označen jako spustitelný. Tento atribut znamená, že se jedná o soubor programu. Tento atribut není nastaven automaticky, aby byl systém chráněn před malwarem staženým z Internetu. Spouštět lze totiž pouze ty soubory, které jsou jako spustitelné označeny. Chcete-li soubor `ahoj.py` nastavit jako spustitelný, použijte příkaz `chmod` (podrobný popis naleznete v kapitole 2, „Správa systému Linux“) takto:

```
chmod +x ahoj.py
```

Nyní se pokuste program spustit přímo zadáním následujícího příkazu:

```
./ahoj.py
```

Navzdory tomu, že jste nevolali prostředí Python, měl by se program `ahoj.py` spustit stejně, jako byste napsali `python ahoj.py`. Program je možné spustit pouze voláním celého umístění – `/home/pi/ahoj.py` – nebo z aktuálního adresáře, kdy je umístění symbolizováno znaky `./`. Aby

byly soubory přístupné stejným způsobem jako kterýkoli jiný příkaz terminálu, musíte je následujícím příkazem zkopírovat do adresáře `/usr/local/bin`:

```
sudo cp ahoj.py /usr/local/bin/
```

Úvodní příkaz `sudo` je nezbytný, protože neprivilegovaní uživatelé z bezpečnostních důvodů nemohou do adresáře `/usr/local/bin` zapisovat. Pokud je soubor `ahoj.py` uložen v adresáři `/usr/local/bin`, který je součástí proměnné `$PATH`, lze jej spustit z libovolného adresáře pouhým zadáním jeho názvu. Vyzkoušejte přejít do jiného adresáře a poté program spustit tímto příkazem:

```
ahoj.py
```

Chcete-li své programy více přiblížit nativním nástrojům, můžete je přejmenovat tak, aby neobsahovaly příponu souboru `.py`. Jestliže chcete program `ahoj.py` přejmenovat tímto způsobem, zadejte v okně terminálu následující příkaz na jediný řádek:

```
sudo mv /usr/local/bin/ahoj.py ↵  
/usr/local/bin/ahoj
```

Po přejmenování stačí ke spuštění programu zadat v okně terminálu či konzole slovo `ahoj`.

Příklad 2: Komentáře, vstupy, proměnné a cykly

Program Hello World (respektive Ahoj Raspberry) sice představuje funkční a jednoduchý úvod do jazyka, ale nikoho zrovna neohromí. Ze své podstaty naznačuje pouze základní možnosti jazyka a nepoužívá žádné principy, které slouží k tvorbě užitečných či zajímavých programů. Další příklad však již pracuje s některými základními nástroji, které jsou potřebné k psaní interaktivních programů jazyka Python.

Stejně jako v prvním příkladu začněte tím, že v prostředí IDLE nebo ve svém textovém editoru otevřete nový prázdný dokument. Program začněte následujícím řádkem shebang:

```
#!/usr/bin/env python
```

Jak jsme již vysvětlili, tento řádek není vyloženě nezbytný, pokud program nebudete nastavovat jako spustitelný, ale jeho vložení ničemu neškodí a jedná se o užitečný zvyk.

Poté do programu vložte komentář, abyste měli k dispozici základní informace o programu, až se k němu později vrátíte. Tento komentář je potřeba zadat na jediný řádek programu stejně jako všechny řádky kódu, které ve výpisech v knize končí symbolem `↵`:

```
# Příklad 2: Program jazyka Python z knihy ↵  
Raspberry Pi Uživatelská příručka
```

Vše, co následuje za symbolem mřížky (s výjimkou řádku shebang), se v jazyce Python považuje za komentář. Když jazyk Python nalezne symbol komentáře, ignoruje jej a přejde na další řádek. Komentování kódu lze doporučit: v současnosti sice nejspíš víte, k čemu jistá část kódu slouží, ale když soubor otevřete za nějakých šest měsíců, už vám to tak jasné být nemusí. Komentáře také usnadňují údržbu kódu, a pokud se rozhodnete svůj kód sdílet s jinými

programátory, mohou jim vaše komentáře pomoci porozumět, jaký mají jednotlivé části kódu účel. U jednoduchých programů není zahrnutí komentářů vyloženě nezbytné, ale stejně jako u vkládání řádků shebang se jedná o dobrý zvyk. Komentáře se mohou nacházet na vlastním řádku (stejně jako v předchozím případě) nebo na řádku s příkazy. V tomto případě jazyk Python provede kód na řádku až po symbol mřížky.

Nyní pomocí následujícího řádku požádejte uživatele, aby zadal své jméno:

```
uzivJmeno = raw_input("Zadejte sve jmeno. ")
```

Tento krátký řádek ve skutečnosti plní mnoho funkcí. První část `uzivJmeno` = požádá jazyk Python, aby vytvořil novou *proměnnou* (variable) – umístění, kam lze uložit jistou informaci – s názvem `uzivJmeno`. Symbol „rovná se“ sděluje jazyku Python, že má proměnnou nastavit na hodnotu, která následuje. V tomto případě však za symbolem není uvedena konkrétní hodnota, ale jiný příkaz: `raw_input`. Jedná se o nástroj, který dokáže přijímat vstupy typu *string* (text) z klávesnice. Zároveň umí zprávu vypisovat na výchozí výstup, aby uživatel věděl, co píše. Program se díky tomu zjednodušuje. Pokud by příkaz nevypisoval znaky zadávané uživatelem, bylo by nutné do programu vložit druhý řádek s příkazem `print`. Nezapomeňte na konec výzvy vložit mezeru, jinak se uživatelský vstup objeví bezprostředně za tečkou.



Upozornění: Když budete ve svých programech požadovat, aby uživatel zadal nějaký text, vždy používejte příkaz `raw_input`. Poskytuje zabezpečení, které samotný příkaz `input` nenabízí. Kdybyste zvolili pouze příkaz `input`, mohl by uživatel do programu *injikovat* svůj vlastní kód, aby program selhal nebo fungoval nežádoucím způsobem.

Když je jméno uživatele nyní bezpečně uloženo v proměnné `uzivJmeno`, může program projevít svou „inteligenci“. Přivítejte uživatele pomocí následujícího řádku:

```
print "Program vita uzivatele", uzivJmeno
```

Z tohoto řádku je patrné, že příkaz `print` z programu Hello World má také druhou funkci: dokáže vypsát obsah proměnných. Tento příkaz `print` se dělí na dvě části: první část vypíše vše mezi uvozovkami a z čárky příkaz `print` pozná, že má na stejném řádku vypsát další data. Stačí uvést název proměnné `uzivJmeno` a jazyk Python pozná, že má zobrazit obsah této proměnné. Výsledkem je zpráva, která obsahuje jméno konkrétního uživatele.



Tip: Chcete-li vypsání výstup hezky naformátovat, můžete na konci příkazu `print` použít instrukci `.format`. Pokud uvedete instrukci `.format`, bude řádek s příkazem `print` vypadat takto:

```
print "Program vita uzivatele {0}.".format(uzivJmeno)
```

Dalším ukázkovým programem bude jednoduchá, ale použitelná kalkulačka. Na rozdíl od prvního příkladu (programu Ahoj Raspberry) bude pokračovat v činnosti, dokud ji uživatel neukončí. Stejně jako v jazyce Scratch to lze zajistit pomocí cyklu. Začněte cyklus zadáním následujících dvou řádků:

```
opakuj = 1
while opakuj == 1:
```

Proč ==?

Již jste nastavovali hodnotu proměnné pomocí jednoho rovnítko. Cyklus `while` však používá dvě za sebou. Když uvedete dva symboly „rovná se“ za sebou, program provede *vyhodnocení* (evaluation), kdy se hodnota proměnné porovnává s libovolnou následující položkou. Jednoduchý symbol „rovná se“ místo toho nastaví proměnnou podle následující hodnoty.

Kromě operátoru dvojitého rovnítka, který poskytne pravdivou hodnotu pouze tehdy, jestliže se proměnná přesně shoduje s danou hodnotou, existují také další porovnání: `>` znamená větší než, `<` menší než, `>=` větší než nebo rovná se, `<=` menší než nebo rovná se a `!=` nerovná se.

Pomocí těchto symbolů vyhodnocení lze řídit tok programu podle pravidel booleovské logiky. Další informace o booleovské logice naleznete v kapitole 10, „Úvod do jazyka Scratch“.

První řádek vytvoří novou proměnnou s názvem `opakuj` a nastaví ji na hodnotu 1. Druhý řádek zahájí cyklus a informuje jazyk Python, že zatímco (`while`) se proměnná `opakuj` rovná 1, měl by opakovaně procházet následující kód. Několik následujících řádků kódu je nutné při psaní *odsadit* pomocí čtyř mezer na začátku každého řádku. Z těchto mezer jazyk Python pozná, které řádky jsou částí cyklu a které řádky leží mimo cyklus. Používáte-li prostředí IDLE, budou mezery vloženy automaticky. Jestliže program píšete v textovém editoru, nezapomeňte mezery vložit ručně.

Ve své nejjednodušší formě přijímá kalkulačka na vstupu dvě čísla a provádí s nimi matematické operace. Aby mohla kalkulačka fungovat, musí nejdříve od uživatele přijmout dvě čísla. K tomu slouží následující řádky:

```
prvniCislo = int(raw_input("Zadejte prvni cislo: "))
druheCislo = int(raw_input("Zadejte druhe cislo: "))
```

Tyto řádky požadují vložení dvou čísel pomocí instrukce `raw_input`. Kromě toho ještě obsahují klíčové slovo `int`. Instrukce `int`, která je zkratkou slova *integer* (celé číslo), požádá jazyk Python, aby vstup nepovažoval za řetězec, ale číslo. V programu kalkulačky je to samozřejmě důležité, protože se slovy počítat nelze.

Když jsou dvě čísla uložena jako proměnné, může program provést příslušné výpočty. Zadejte následující řádky, které sečítají, odečítají a násobí dvě čísla a odesílají výstup uživateli:

```
print prvniCislo, "plus", druheCislo, "se rovna", ←
prvniCislo + druheCislo
print prvniCislo, "minus", druheCislo, "se rovna", ←
prvniCislo - druheCislo
print prvniCislo, "krat", druheCislo, "se rovna", ←
prvniCislo * druheCislo
```

Všimněte si, že zatímco operace sčítání a odčítání používají očekávané symboly plus a minus, násobení je označeno symbolem `*`. Nepřehlédněte také, že mezi uvozovkami nejsou žádné formátovací mezery. Když jazyk Python vypisuje vedle sebe celá čísla a řetězce, přidává mezery podle potřeby automaticky. Nakonec poznamenejme, že program neobsahuje operaci dělení (kterou by znázorňoval symbol `/`). Ukázkový program kalkulačky totiž pracuje s celými čísly, která neumožňují počítání s desetinnými místy ani zlomky.

Výpočetní část programu je sice hotová, avšak program poběží neomezeně dlouho, protože jazyk Python aktuálně nemá možnost cyklus ukončit. Chcete-li uživateli poskytnout možnost ukončení programu, přidejte následující řádek:

```
opakuj = int(raw_input("Chcete-li vkladat dalsi cisla, zadejte cislo 1. ↵
Pokud chcete program ukoncit, zadejte libovolne jine cislo: "))
```

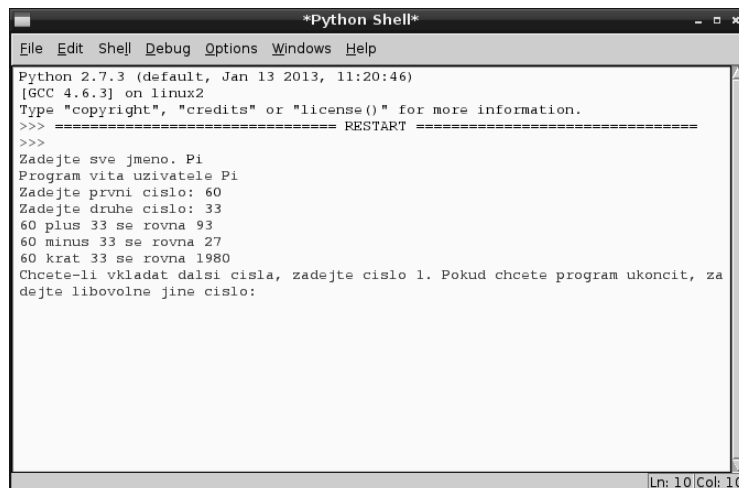
Uživatel tak může změnit hodnotu proměnné `opakuj`, která řídí cyklus `while`. Pokud uživatel zadá číslo 1, proměnná `opakuj` se i nadále rovná 1 a cyklus proběhne znovu. Jestliže však uživatel uvede libovolné jiné číslo, vyhodnocení již není pravdivé (proměnná `opakuj` se již nerovná 1) a cyklus bude ukončen.

Dokončený program by měl odpovídat následujícímu výpisu. Pamatujte, že u všech řádků ukončených symbolem `↵` je nutné pokračovat v zadávání následujícím řádkem:

```
#!/usr/bin/env python
# Příklad 2: Program jazyka Python z knihy ↵
Raspberry Pi Uzivatel'ska prirucka
uzivJmeno = raw_input("Zadejte sve jmeno. ")
print "Program vita uzivatele", uzivJmeno
opakuj = 1
while opakuj == 1:
    prvniCislo = int(raw_input("Zadejte prvni cislo: "))
    druheCislo = int(raw_input("Zadejte druhe cislo: "))
    print prvniCislo, "plus", druheCislo, "se rovna", ↵
    prvniCislo + druheCislo
    print prvniCislo, "minus", druheCislo, "se rovna", ↵
    prvniCislo - druheCislo
    print prvniCislo, "krat", druheCislo, "se rovna", ↵
    prvniCislo * druheCislo
    opakuj = int(raw_input("Chcete-li vkladat dalsi cisla, zadejte cislo 1. ↵
    Pokud chcete program ukoncit, zadejte libovolne jine cislo: "))
```

Uložte program pod názvem `kalkulacka.py` a spusťte jej buď pomocí příkazu Run Module z nabídky Run v prostředí IDLE, nebo zadáním příkazu `python kalkulacka.py` v okně terminálu. Po zobrazení výzvy zadejte své jméno a potom vkládejte čísla, s nimiž chcete počítat (viz obrázek 11.5). Až vás program omrzí, ukončete jej zadáním libovolného čísla různého od 1.

Další krátké programy, které seznamují s důležitými principy jazyka Python, naleznete na oficiální stránce wiki „Python Simple Programs“ na adrese <http://wiki.python.org/moin/SimplePrograms>.



Obrázek 11.5: Spuštění souboru kalkulacka.py v prostředí IDLE

Příklad 3: Tvorba her pomocí knihovny pygame

Na ukázkou možností jazyka Python v tomto příkladu vytvoříme plně funkční arkádovou hru, která je založena na klasické hře Snake neboli Nibbles. Použijeme k tomu externí knihovnu jazyka Python s názvem *pygame*.

Původním autorem knihovny *pygame* je Pete Shinnars. Tato knihovna shrnuje *moduly* jazyka Python, které jazyk rozšiřují o nové možnosti – konkrétně o možnosti, jejichž účelem je usnadnit psaní her v tomto jazyce. Každý modul *pygame* poskytuje funkci požadovanou v moderní hře, včetně zvuku, grafiky, a dokonce podpory síťového připojení. Ačkoli lze hry v jazyce Python psát i bez knihovny *pygame*, je mnohem jednodušší využít kód, který je v této knihovně již k dispozici.

Chcete-li začít psát program založený na knihovně *pygame*, musíte ji nejdříve nainstalovat. Pracujete-li v doporučené distribuci Debian, stačí v konzole nebo okně terminálu zadat následující příkaz:

```
sudo apt-get install python-pygame
```

V případě jiných distribucí si můžete zdrojové soubory knihovny *pygame* stáhnout z jejího oficiálního webu na adrese <http://www.pygame.org/download.shtml>. Na stejné stránce jsou uvedeny i pokyny k instalaci.

Programy *pygame* lze spouštět stejným způsobem jako kterékoli jiné projekty jazyka Python. Otevřete v prostředí IDLE nebo v textovém editoru nový prázdný dokument a na jeho začátek zadejte následující řádek shebang:

```
#!/usr/bin/env python
```

Poté je potřeba jazyku Python oznámit, že tento program pracuje s modulem `pygame`. K tomu slouží instrukce `import`, která jazyk Python požádá o načtení externího modulu (jiného souboru Python) a jeho zpřístupnění z aktuálního programu. Chcete-li do svého nového projektu importovat potřebné moduly, zadejte následující dva řádky:

```
import pygame, sys, time, random
from pygame.locals import *
```

První řádek importuje hlavní modul `pygame` spolu s modulem jazyka Python `sys`, `time` a `random`, na které bude tento program také odkazovat. Při volání modulu je obvykle nutné zadat jeho název následovaný tečkou a názvem instrukce, která je součástí modulu. Druhý řádek předchozího kódu však zajistí, že jazyk Python načte všechny instrukce z modulu `pygame.locals`, jako by se jednalo o nativní instrukce. Díky tomu si při volání těchto instrukcí ušetříte psaní. Jiné názvy modulů – např. `pygame.clock`, což je modul nezávislý na modulu `pygame.locals` – bude i nadále nutné uvádět v nezkráceném tvaru.

Zadejte následující dva řádky, které nastaví knihovnu `pygame`, aby k ní mohl ukázkový program přistupovat:

```
pygame.init()
taktovaciFrekvence = pygame.time.Clock()
```

První řádek požádá knihovnu `pygame`, aby se inicializovala. Druhý řádek nastaví novou proměnnou s názvem `taktovaciFrekvence`, která umožní kontrolovat rychlost hry. Následně pomocí následujících dvou řádků nastavte novou zobrazovací plochu knihovny `pygame`, což je plocha, na níž se budou vykreslovat objekty hry:

```
herniPlocha = pygame.display.set_mode((640, 480))
pygame.display.set_caption('Had s malinou')
```

Dále definujte barvy, se kterými bude program pracovat. Tento krok sice není úplně nezbytný, ale opět vám ušetří psaní: pokud chcete pro určitý objekt nastavit červenou barvu, můžete jednoduše použít proměnnou `cervenaBarva` a nemusíte volat instrukci `pygame.Color` a pamatovat si tři barevné hodnoty pro červenou, zelenou a modrou složku. Chcete-li definovat barvy pro tento ukázkový program, zadejte následující řádky:

```
cervenaBarva = pygame.Color(255, 0, 0)
cernaBarva = pygame.Color(0, 0, 0)
bilaBarva = pygame.Color(255, 255, 255)
sedaBarva = pygame.Color(150, 150, 150)
```

Několik dalších řádků inicializuje některé proměnné hry, aby byly připraveny k použití. Jedná se o důležitý krok, protože pokud by tyto proměnné při spuštění hry zůstaly prázdné, jazyk Python by nemohl pokračovat v činnosti. Zatím se nemusíte starat o to, k čemu jednotlivé proměnné slouží. Prostě zadejte následující řádky:

```
polohaHada = [100,100]
castiHada = [[100,100],[80,100],[60,100]]
polohaMaliny = [300,300]
malinaVyrostla = 1
```

```
smer = 'doprava'
zmenaSmeru = smer
```

Všimněte si, že tři z proměnných (`polohaHada`, `castiHada` a `polohaMaliny`) jsou definovány jako seznam hodnot oddělených čárkami. Jazyk Python tedy proměnné vytvoří jako *seznamy*, kdy je pod jedním názvem proměnné uloženo více různých hodnot. Později se dozvíte, jak lze přistupovat k jednotlivým hodnotám uloženým v seznamu.

Dále je potřeba definovat novou funkci, což je fragment kódu jazyka Python, který lze později volat z jiné části programu. Funkce jsou výhodné, protože zabraňují opakování kódu a zvyšují srozumitelnost programu. Máte-li určitou sadu instrukcí, které jsou ve stejném programu potřebné vícekrát, můžete vytvořit funkci pomocí klíčového slova `def`. Díky tomu stačí kód zadat jen jednou, a pokud je nutné jej později změnit, lze to provést na jediném místě. Definujte funkci `konecHry` pomocí následujících řádků:

```
def konecHry():
    konecHryPismo = pygame.font.Font ←
        ('freesansbold.ttf', 72)
    konecHryPlocha = konecHryPismo.render ←
        ('Konec hry', True, sedaBarva)
    konecHryObd = konecHryPlocha.get_rect()
    konecHryObd.midtop = (320, 10)
    herniPlocha.blit(konecHryPlocha, konecHryObd)
    pygame.display.flip()
    time.sleep(5)
    pygame.quit()
    sys.exit()
```

Stejně jako u cyklů musí být kód funkcí odsazený. Na začátku všech řádků za instrukcí `def` by měly být čtyři mezery. Používáte-li prostředí IDLE, budou mezery vloženy automaticky, ale jestliže program píšete v textovém editoru, nezapomeňte mezery vložit ručně. Odsazení můžete ukončit za posledním řádkem funkce – `sys.exit()`.

Funkce `konecHry` používá několik příkazů knihovny `pygame` a plní jednoduchý úkol: velkými písmeny vypíše na obrazovku text `Konec hry`, pozastaví činnost programu na 5 sekund a poté ukončí knihovnu `pygame` i samotné prostředí jazyka Python. Možná vám připadá zvláštní definovat pokyny na ukončení hry, která zatím ani nezačala, ale funkce je vždy potřeba definovat dříve, než jsou volány. Jazyk Python tyto instrukce neprovede, dokud k tomu nedostane pokyn prostřednictvím nově zadané instrukce `konecHry`.

Když máte hotový začátek programu, můžete přejít k hlavní části. Tato část je uzavřena do nekonečného cyklu – cyklu `while`, který se neustále opakuje. Díky tomu může hra pokračovat, dokud `had` nezemře, když narazí do zdi nebo se zakousne do svého ocasu. Hlavní cyklus uveďte tímto řádkem:

```
while True:
```

Aniž by cokoli vyhodnocoval, jazyk Python zkontroluje, zda je hodnota `True` pravdivá. Vzhledem k tomu, že tato podmínka platí vždy, cyklus bude pokračovat neomezeně dlouho – tedy alespoň do chvíle, než jazyk Python požádáte o opuštění cyklu voláním funkce `konecHry`.

Pokračujte v psaní programu následujícími řádky a dbejte přitom na správné odsazení:

```
for event in pygame.event.get():
    if event.type == QUIT:
        pygame.quit()
        sys.exit()
    elif event.type == KEYDOWN:
```

První řádek, který následuje bezprostředně za začátkem cyklu `while`, by měl být odsazen o čtyři mezery. Jedná se však také o cyklus, který pomocí instrukce `for` kontroluje události `pygame`, jako je stisknutí kláves. Z tohoto důvodu musí být další řádek odsazen o další čtyři mezery, tj. celkem o osm mezer. Tento řádek je však rovněž cyklem, který pomocí instrukce `if` ověřuje, zda uživatel stiskl klávesu. Řádek za ním – `pygame.quit()` – je proto odsazen o další čtyři mezery, a celkem jich tedy na začátku obsahuje 12. Z této logické posloupnosti odsazení jazyk Python zjistí, kde jednotlivé cykly začínají a končí, což je důležité: pokud byste zadali mezer více či méně, program by nefungoval správně. Programování ve vývojovém prostředí typu IDLE, které se pokouší automaticky odsazovat kód podle potřeby, je proto zpravidla snazší než v editorech neformátovaného textu.

V cyklu `if` jazyk Python kontroluje, zda je vyhodnocení určité podmínky pravdivé. První kontrola `if event.type == QUIT` zajistí, že jazyk Python spustí následující odsazený kód v případě, že knihovna `pygame` oznámí zprávu `QUIT` (k tomu dochází, když uživatel stiskne klávesu `Escape`). Další dva řádky by vám již měly být povědomé z funkce `konecHry`: zajistí ukončení knihovny `pygame` a prostředí jazyka Python.

Řádek začínající klíčovým slovem `elif` rozšiřuje cyklus `if`. Instrukce `elif`, která je zkratkou slov *else if* (nebo pokud), se vyhodnotí, jestliže byla předchozí instrukce `if` vyhodnocena jako nepravdivá. V tomto případě instrukce `elif` kontroluje, zda knihovna `pygame` oznamuje událost `KEYDOWN`, která je vrácena, když uživatel stiskne klávesu na klávesnici. Kód, který má být proveden, pokud je podmínka `elif` pravdivá, by měl být stejně jako v případě instrukce `if` odsazen o další čtyři mezery oproti odsazení samotné instrukce `elif`. Chcete-li, aby instrukce `elif` při stisknutí klávesy uživatelem provedla nějakou akci, zadejte následující řádky:

```
if event.key == K_RIGHT or event.key == ord('d'):
    zmenaSmeru = 'doprava'
if event.key == K_LEFT or event.key == ord('a'):
    zmenaSmeru = 'doleva'
if event.key == K_UP or event.key == ord('w'):
    zmenaSmeru = 'nahoru'
if event.key == K_DOWN or event.key == ord('s'):
    zmenaSmeru = 'dolů'
if event.key == K_ESCAPE:
    pygame.event.post(pygame.event.Event(QUIT))
```

Tyto instrukce mění hodnotu proměnné `zmenaSmeru`, která řídí směr hráčova hada. Pomocí klíčového slova `or` s příkazem `if` lze vyhodnocovat více podmínek. V tomto případě lze řídit hada dvěma způsoby: hráč může ovládat hada směrem nahoru, doprava, dolů či doleva pomocí kurzorových kláves nebo kláves W, A, S a D. Dokud hráč nestiskne klávesu, had postupuje doprava podle hodnoty, kterou má proměnná `zmenaSmeru` nastavenou při spuštění programu.

Když se vrátíte k proměnným, které jste inicializovali na začátku programu, můžete si všimnout další proměnné s názvem `smer`. Tato proměnná v kombinaci s proměnnou `zmenaSmeru` kontroluje, zda uživatel při ovládání hada zvolil platný směr. Program by neměl povolit, aby se had mohl obrátit „čelem vzad“ – když had kousne sám sebe, hyne a hra končí. Aby k tomu nedocházelo, je směr požadovaný hráčem (který je uložen v proměnné `zmenaSmeru`) porovnáván s aktuálním směrem pohybu hada – ten uchovává proměnná `smer`. Jestliže jsou oba směry protilehlé, je instrukce ignorována a had pokračuje stejným směrem jako dosud. Kvůli porovnání zadejte tyto řádky:

```
if zmenaSmeru == 'doprava' and not ←
smer == 'doleva':
    smer = zmenaSmeru
if zmenaSmeru == 'doleva' and not ←
smer == 'doprava':
    smer = zmenaSmeru
if zmenaSmeru == 'nahoru' and not ←
smer == 'dolů':
    smer = zmenaSmeru
if zmenaSmeru == 'dolů' and not ←
smer == 'nahoru':
    smer = zmenaSmeru
```

Když program zkontroluje, zda uživatelský vstup dává smysl, může překreslit hada, který je na obrazovce znázorněn jako řada bloků. V každém kroku se had posune o vzdálenost, která se rovná velikosti jedné z jeho čtvercových částí. Vzhledem k tomu, že každý segment měří 20 pixelů, můžete knihovnu `pygame` požádat, aby hada posunula o jeden segment v příslušném směru. Zadejte následující kód:

```
if smer == 'doprava':
    polohaHada[0] += 20
if smer == 'doleva':
    polohaHada[0] -= 20
if smer == 'nahoru':
    polohaHada[1] -= 20
if smer == 'dolů':
    polohaHada[1] += 20
```

Operátory `+=` a `-=` umožňují změnit hodnotu proměnné o určitý krok: operátor `+=` zvýší předchozí hodnotu proměnné o daný krok, zatímco operátor `-=` nastaví hodnotu proměnné jako rozdíl předchozí hodnoty a kroku. Kratší zápis `polohaHada[0] += 20` znamená totéž jako `polohaHada[0] = polohaHada[0] + 20`. Číslo v hranatých závorkách za proměnnou `poloha`

Hada je pozice v seznamu, kterou příkaz mění: první hodnota v seznamu `polohaHada` uchovává polohu hada na ose X a druhá hodnota udává jeho polohu na ose Y. Jazyk Python začíná počítat od nuly, takže osu X řídí položka `polohaHada[0]` a osu Y položka `polohaHada[1]`. Pokud by byl seznam delší, bylo by možné upravovat další položky příslušným zvýšením indexu: `[2]`, `[3]` atd.

Seznam `polohaHada` má sice vždy délku dvou prvků, což ale neplatí pro jiný seznam vytvořený na začátku programu: `castiHada`. Tento seznam obsahuje polohu těla hada za jeho hlavou. Jak had pojídá maliny a prodlužuje se, délka tohoto seznamu se zvětšuje a tím zároveň vzrůstá obtížnost hry: po chvíli hry je stále těžší nenarazit hlavou hada do jeho těla. Když se hlava dotkne těla, had hyne a hra končí. Chcete-li zajistit, aby tělo hada rostlo, zadejte tento řádek:

```
castiHada.insert(0,list(polohaHada))
```

Tento příkaz pomocí instrukce `insert` vloží do seznamu `castiHada` novou hodnotu: aktuální pozici hada. Když interpret jazyka Python dosáhne tohoto řádku, zvýší pokaždé délku hadova těla o jeden segment a umístí tento segment na aktuální polohu hadovy hlavy. Z hlediska hráče to vypadá, jako by had rostl. To by se však mělo stávat jen tehdy, když had sní malinu – jinak bude neustále růst. Zadejte následující řádky:

```
if polohaHada[0] == polohaMaliny[0] <-
and polohaHada[1] == polohaMaliny[1]:
    malinaVyrostla = 0
else:
    castiHada.pop()
```

První instrukce zkontroluje, zda souřadnice X a Y hadovy hlavy odpovídají souřadnicím X a Y maliny, která představuje cíl hráče. Jestliže se hodnoty shodují, program předpokládá, že had malinu snědl, a nastaví proměnnou `malinaVyrostla` na hodnotu 0. Instrukce `else` informuje jazyk Python, jak postupovat, když had malinu nezkonzumoval: vyjmout (`pop`) nejstarší hodnotu ze seznamu `castiHada`.

Instrukce `pop` je jednoduchá, ale chytrá: vrátí ze seznamu nejstarší hodnotu, ale zároveň ji odebere, takže se seznam o jednu položku zkrátí. V případě seznamu `castiHada` požádá interpret jazyka Python, aby odstranil část hadova těla, která leží nejdále od jeho hlavy. Z hlediska hráče to vypadá, jako by se celý had posunul a přitom neprodloužil. Ve skutečnosti se však had zvětší na jednom konci a zmenší na druhém. Díky příkazu `else` bude instrukce `pop` provedena pouze tehdy, když malina zůstane nesnědená. Pokud had malinu spolkl, poslední položka v seznamu nebude odstraněna, takže se délka hada prodlouží o jeden segment.

Po chvíli hry je stále pravděpodobnější, že hráč malinu získal. Hra, ve které by bylo možné sníst jen jednu malinu, by byla docela nudná. Zadejte tedy následující řádky, které přidají na hrací plochu novou malinu, když had zkonsumuje tu stávající:

```
if malinaVyrostla == 0:
    x = random.randrange(1,32)
    y = random.randrange(1,24)
    polohaMaliny = [int(x*20),int(y*20)]
    malinaVyrostla = 1
```

Tato část kódu kontroluje, zda byla malina snědena. Přitom testuje, zda je proměnná `malinaVyrostla` nastavena na hodnotu 0. V tomto případě kód vybere náhodné umístění na hrací ploše pomocí modulu `random`, který jste importovali na začátku programu. Toto umístění je pak vynásobeno velikostí části hada (s šířkou a výškou 20 pixelů), aby jazyk Python mohl umístit novou malinu na hrací ploše. Je důležité volit umístění maliny náhodně: hráč se tak nemůže naučit, kde se malina objeví příště. Nakonec je proměnná `malinaVyrostla` nastavena zpět na hodnotu 1. Tím je zajištěno, že na hrací ploše nikdy nebude zobrazena více než jedna malina.

Nyní tedy máte kód, který je potřeba k pohybu a růstu hada a konzumaci i růstu malin – tento proces se v herní terminologii označuje jako *respawning* (generování). Na obrazovce se však nic nevymaluje. Zadejte následující řádky:

```
herniPlocha.fill(cernaBarva)
for poloha in castiHada:
    pygame.draw.rect(herniPlocha, bilaBarva, Rect ←
        (poloha[0], poloha[1], 20, 20)) ←
    pygame.draw.rect(herniPlocha, cervenaBarva, Rect ←
        (polohaMaliny[0], polohaMaliny[1], 20, 20))
pygame.display.flip()
```

Tyto příkazy požádají knihovnu `pygame`, aby vyplnila pozadí herní plochy černou barvou, vykreslila části hada (hlavu a tělo) bílou barvou a nakonec zobrazila červenou malinu. Poslední řádek `pygame.display.flip()` zajistí, aby knihovna `pygame` aktualizovala obrazovku – bez této instrukce nebudou položky pro hráče viditelné. Pokaždé, když dokončíte vykreslování objektů na obrazovce, nezapomeňte použít instrukci `pygame.display.flip()`, aby se změny pro hráče zobrazily.

Aktuálně je had nesmrtelný. Hra, kde hráč nikdy nemůže přijít o život, by se rychle stala nudnou. Zadejte tedy následující řádky, které definují scénáře hadova skonu:

```
if polohaHada[0] > 620 or polohaHada[0] < 0:
    konecHry()
if polohaHada[1] > 460 or polohaHada[1] < 0:
    konecHry()
```

První příkaz `if` kontroluje, zda had neopustil hrací plochu vodorovně, zatímco druhý příkaz `if` ověřuje, zda se had nedostal mimo hrací plochu ve svislém směru. V každém případě to pro hada nevěští nic dobrého: dojde k volání funkce `konecHry`, kterou jste definovali v předchozí části programu. Tato funkce vypíše zprávu na obrazovku a ukončí hru. Had by měl také zemřít, když jeho hlava narazí na libovolnou část jeho těla. Doplňte tedy tyto řádky:

```
for teloHada in castiHada[1:]:
    if polohaHada[0] == teloHada[0] and ←
        polohaHada[1] == teloHada[1]:
        konecHry()
```

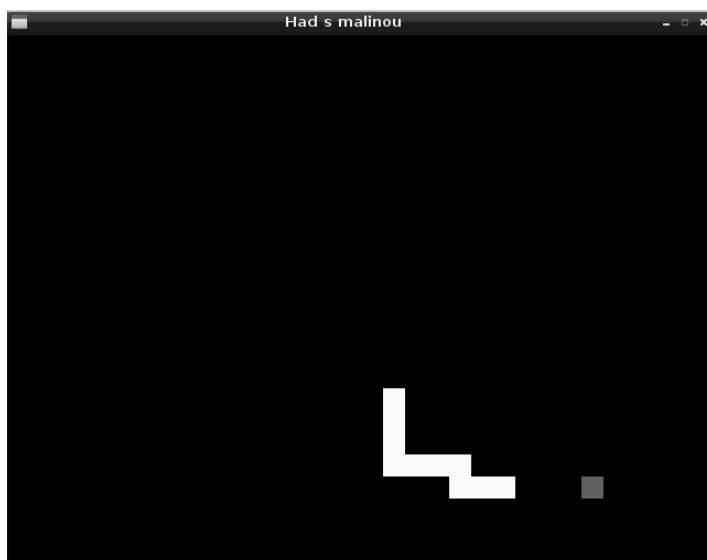
Příkaz `for` prochází všechny polohy částí hada od druhé položky seznamu až po jeho konec. Přitom je porovnává s aktuální polohou hlavy hada. Je důležité, abyste porovnávání nezačali od první položky, ale pomocí proměnné `castiHada[1:]` až od druhé. První položka je vždy

nastavena na polohu hlavy. Pokud byste porovnávání začali zde, had by okamžitě zemřel hned při spuštění hry.

K dokončení hry již stačí jen ovládání její rychlosti pomocí proměnné `taktovaciFrekvence`. Bez této proměnné, kterou jste iniciovali na začátku programu, by hra běžela příliš rychle a nedala by se ovládat. Chcete-li program dokončit, zadejte následující řádek:

```
taktovaciFrekvence.tick(20)
```

Pokud máte dojem, že je hra příliš snadná či pomalá, můžete toto číslo zvýšit. Případně je-li hra příliš těžká či rychlá, nastavte menší číslo. Uložte program pod názvem `hadsma1inou.py` a spusťte jej buď pomocí příkazu Run Module z nabídky Run v prostředí IDLE, nebo zadáním příkazu `python hadsma1inou.py` v okně terminálu. Hra začne hned po spuštění (viz obrázek 11.6), takže se nezapomeňte připravit!



Obrázek 11.6: Hraní hry „Had s malinou“ v počítači Raspberry Pi

Úplný výpis programu hry Had s malinou je uveden v Příloze A, „Recepty jazyka Python“, a na stránce knihy na adrese <http://knihy.cpress.cz/K2123> v sekci Soubory ke stažení. Když si zdrojový kód stáhnete z webu, ušetříte si sice trochu psaní, ale při ručním zadávání kódu si můžete ověřit, zda jednotlivým částem programu rozumíte. Kromě funkcí použitých ve hře Had s malinou poskytuje knihovna pygame mnoho funkcí, které se v tomto programu neuplatňují, včetně přehrávání zvuku, obsluhy pohyblivých prvků, aby bylo možné zobrazovat dokonalejší animace, a ovládání myši. Chcete-li se o pokročilejších funkcích knihovny pygame dozvědět více, nejlepším zdrojem informací je oficiální web této knihovny. Na adrese <http://www.pygame.org/wiki/tutorials> si můžete stáhnout kurzy a ukázkové programy, které vám ukáží, jak vše funguje.

Příklad 4: Jazyk Python a síťová konektivita

Zatím jste se naučili, jak lze pomocí jazyka Python psát samostatné programy. Tento jazyk však umožňuje vytvářet i programy, které komunikují s vnějším světem prostřednictvím síťového připojení počítače. Další příklad, který napsal Tom Hudson, nabízí stručný nástin těchto možností. Jedná se o nástroj na sledování uživatelů, kteří jsou připojeni ke kanálu IRC (Internet Relay Chat).

Jako obvykle vytvořte nový projekt v prostředí IDLE nebo textovém editoru a zadejte řádek shebang spolu s komentářem, který bude popisovat účel programu:

```
#!/usr/bin/env python
# Kontrola kanálu IRC, pro knihu ↵
Raspberry Pi Uživatelská příručka, napsal Tom Hudson
```

Dále pomocí následujícího řádku importujte moduly, které program potřebuje – `sys`, `socket` a `time`:

```
import sys, socket, time
```

Moduly `sys` a `time` jste již použili v předchozím programu *Had s malinou*, ale s modulem `socket` jste zatím nepracovali. Modul `socket` umožňuje v jazyku Python otevírat, zavírat, číst a zapisovat síťové sokety. Programy jazyka Python tak získávají základní síťové možnosti. Právě modul `socket` dovoluje, aby se tento ukázkový program připojil ke vzdálenému serveru IRC.

Tento program ke své činnosti potřebuje určité *konstanty*. Konstanty se podobají proměnným v tom, že jim lze přiřadit hodnoty. Na rozdíl od proměnných by se však hodnoty konstant neměly měnit. Kvůli rozlišení konstant a proměnných se doporučuje celé názvy konstant zadávat velkými písmeny. Díky tomu je možné na první pohled poznat, zda určitá část kódu pracuje s konstantou nebo proměnnou. Zadejte do programu tyto dva řádky:

```
RPL_NAMREPLY    = '353'
RPL_ENDOFNAMES = '366'
```

Jedná se o *stavové kódy* IRC, kterými server informuje o dokončení určité operace. Na základě uvedených kódů může program zjistit, zda ze serveru IRC přijal požadovaný seznam jmen. V dalším kroku zadejte následující řádky, abyste nastavili proměnné pro připojení serveru:

```
irc = {
    'hostitel' :    'chat.freenode.net',
    'port' :       6667,
    'kanal' :      '#raspiuserguide',
    'intervaljmen' : 5
}
```

První řádek požádá jazyk Python, aby vytvořil datový typ *dict*. Tento typ (zkratka slova dictionary, tj. slovník) umožňuje do jedné hlavní proměnné – v tomto případě s názvem `irc` – uložit více proměnných. Na jednotlivé proměnné lze pak odkazovat v dalších částech programu. Program by sice bylo možné napsat i bez ukládání proměnných do datových typů *dict*,

ale v takové podobě by byl mnohem hůře čitelný. Proměnná `dict` začíná otevírací složenou závorkou a končí ukončovací složenou závorkou na posledním řádku.

Proměnnou `hostitel` nastavte podle plně kvalifikovaného doménového názvu (FQDN) serveru IRC, ke kterému se program bude připojovat. V této ukázce se používá server `chat.freenode.net`, ale chcete-li program přizpůsobit a zvolit jiný server, nastavte jeho doménový název na tomto místě. Proměnná `port` informuje program, na kterém síťovém portu je služba IRC spuštěna. Obvykle se jedná o port 6667. Z proměnné `kanal` jazyk Python pozná, ke kterému kanálu se má při sledování uživatelů připojit, zatímco proměnná `interval` jmen určuje, jak dlouho bude program čekat do další aktualizace seznamu uživatelů. Tato hodnota se udává v sekundách.

Nastavte druhou proměnnou typu `dict`, která bude uchovávat uživatelsky specifické proměnné. Přitom zadejte tyto řádky:

```
uzivatel = {
    'prezdivka' :      'botnick',
    'uzivatelskejmeno' : 'botuser',
    'nazevhostitele' :  'localhost',
    'navezserveru' :    'localhost',
    'skutecnejmeno' :   'Raspberry Pi Names Bot'
}
```

Stejně jako u proměnné `irc` jsou všechny tyto proměnné uloženy v proměnné datového typu `dict` s názvem `uzivatel`, aby bylo zřejmé, do které části jednotlivé hodnoty patří. Proměnná `prezdivka` by měla být nastavena podle přezdívky IRC, kterou bude program používat. Pokud se k serveru IRC chcete zároveň připojit sami, nepoužívejte svou obvyklou přezdívku. Místo toho je vhodné připojit na konec jména řetězec `-bot`, aby bylo jasné, že se jedná o program, a nikoli o živou osobu. Stejně postupujte s proměnnou `uzivatelskejmeno` a do proměnné `skutecnejmeno` zadejte popisný text, aby bylo zřejmé, komu robot patří. U proměnných `nazevhostitele` a `navezserveru` můžete ponechat výchozí hodnotu `localhost` nebo je můžete upravit tak, aby odpovídaly vaší internetové adrese.

Modul `socket` vyžaduje, aby uživatel vytvořil *objekt* soketu. Tento objekt poskytuje zbývající části programu síťovou konektivitu. Objekt soketu vytvoříte pomocí tohoto řádku:

```
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
```

Dále musíte program požádat, aby se pokusil připojit k serveru IRC, který je uveden v proměnných na začátku programu. Zadejte následující řádky:

```
print 'Probiha pripojeni k serveru %(hostitel)s:%(port)s...' % irc
try:
    s.connect((irc['hostitel'], irc['port']))
except socket.error:
    print 'Doslo k chybe pripojeni k serveru IRC ↵
    %(hostitel)s:%(port)s' % irc
    sys.exit(1)
```

Příkazy `try` a `except` doplněné do tohoto kódu slouží ke *zpracování chyb*. Jestliže se systém nemůže připojit k serveru (například proto, že počítač Pi nemá internetovou konektivitu nebo

je server vypnutý kvůli údržbě), program vypíše chybovou zprávu a korektně se ukončí. Řádek `s.connect` požádá modul `socket`, aby se pokusil připojit k serveru IRC pomocí proměnných `hostitel` a `port`, které jsou součástí proměnné `irc` typu `dict`.

Pokud program není ukončen kvůli výjimce, znamená to, že se úspěšně připojil k serveru IRC. Chcete-li získat seznam jmen v kanálu, musíte se však nejdříve serveru identifikovat a zadat několik příkazů pomocí funkce `send` modulu `socket`. Zadejte do programu tyto řádky:

```
s.send('NICK %(prezdivka)s\r\n' % uzivatel)
s.send('USER %(uzivatelskejmeno)s %(nazevhostitele)s \r\n' % (nazevserveru, skutecejmeno))
s.send('JOIN %(kanal)s\r\n' % irc)
s.send('NAMES %(kanal)s\r\n' % irc)
```

Funkce `send` pracuje téměř stejně jako funkce `print`, ale místo vypisování na standardní výstup (obvykle okno terminálu nebo konzola) odesílá výstup pomocí síťového připojení. V tomto případě program odesílá textové řetězce serveru IRC a požádá server, aby jej zaregistroval s přezdívkou, která je uložena v proměnné `prezdivka`, a sdělí mu obsah proměnných `uzivatelskejmeno`, `nazevhostitele`, `nazevserveru` a `skutecejmeno`. Poté program odešle příkaz na připojení ke kanálu, který je nastaven v proměnné `kanal`, a nakonec odešle příkaz, kterým si od serveru vyžádá seznam uživatelů v daném kanálu. Tento příklad je sice přizpůsoben službě IRC, ale stejný základní princip lze využít k zadávání příkazů libovolné síťové službě. Po úpravách by tento program mohl vypisovat seznam souborů na serveru FTP nebo nepřečtených e-mailů na serveru POP3.

Přijímání dat ze `socketu` je trochu složitější. Nejdříve musíte vytvořit prázdnou textovou proměnnou, která bude sloužit jako *příjímací vyrovnávací paměť* (receive buffer), která bude uchovávat data přijatá ze serveru, než je bude možné zpracovat. Inicializujte vyrovnávací paměť pomocí tohoto řádku:

```
prijimaci_pamet = ''
```

Všimněte si, že za symbolem rovnítka následují dvě jednoduché uvozovky, nikoli jedna dvojité.

Poté zadejte následující řádek, abyste vytvořili prázdný seznam, který bude uchovávat jména uživatelů:

```
jmena = []
```

Datový typ list seznamu je stejný typ, do jakého jste ukládali polohy ve hře *Had s malinou*. Oproti normální proměnné může obsahovat více hodnot – v tomto případě se jedná o jména uživatelů, kteří jsou připojeni ke kanálu IRC.

V dalším kroku vytvořte nekonečný cyklus, ve kterém se bude program neustále dotazovat serveru na uživatelská jména a vypisovat je na obrazovku. Cyklus začněte příkazem:

```
while True:
    prijimaci_pamet += s.recv(1024)
```

První řádek cyklu za řádkem `while True`: požádá modul `socket`, aby přijal 1 024 bajtů (1 kB) dat ze serveru IRC a umístil je do proměnné `prijimaci_pamet`. Vzhledem k tomu, že se místo

pouhého operátoru = používá operátor +=, budou přijatá data připojena ke stávajícímu obsahu vyrovnávací paměti. Hodnota 1 024 bajtů je zvolena víceméně náhodně.

V dalším kroku je potřeba rozdělit vyrovnávací paměť na jednotlivé řádky textu. Slouží k tomu následující řádky programu:

```
radky = prijimaci_pamet.split('\r\n')
prijimaci_pamet = radky.pop();
```

První řádek nastaví proměnnou `radky` tak, aby obsahovala celý řádek textu z přijímací vyrovnávací paměti. To zajišťuje funkce `split`, která vyhledává znaky *EOL* (end of line – konec řádku) symbolizované řetězcem `\r\n`. Tyto znaky se vyskytují pouze na konci řádku. Jestliže je tedy vyrovnávací paměť rozdělena tímto způsobem, máte jistotu, že proměnná `radky` obsahuje pouze kompletní řádky odpovědi ze serveru. Instrukce `pop` na druhém řádku zajišťuje, že jsou z proměnné `prijimaci_pamet` odebírány pouze úplné řádky: vzhledem k tomu, že odpovědi ze serveru jsou načítány v dávkách po 1 kB, je pravděpodobné, že v libovolném okamžiku bude vyrovnávací paměť obsahovat pouze část řádku. V tomto případě zůstane neúplný řádek ve vyrovnávací paměti a čeká na načtení zbytku řádku při následujícím průběhu cyklu a přijetí dalšího 1 kB dat ze serveru.

V této fázi proměnná `radky` obsahuje seznam úplných odpovědí – celých řádků – přijatých ze serveru. Zadejte následující kód, který tyto řádky zpracuje a zjistí jména účastníků kanálu:

```
for radek in radky:
    odpoved = radek.rstrip().split(' ', 3)
    kod_odpovedi = odpoved[1]
    if kod_odpovedi == RPL_NAMREPLY:
        seznam_jmen = odpoved[3].split(':')[1]
        jmena += seznam_jmen.split(' ')
```

Tyto příkazy jsou provedeny pro každý řádek nalezený v proměnné `radky` a zjišťují číselný kód odpovědi, který poskytl server IRC. Existuje sice mnoho různých kódů odpovědi, ale tento program se zajímá pouze o dva z nich, které byly definovány jako konstanty na začátku programu: 353, který znamená, že následuje seznam jmen, a 366, který označuje konec seznamu. Příkaz `if` vyhledá první z těchto odpovědí a poté pomocí funkce `split` načte jména a přidá je do seznamu `jmena`.

V této fázi seznam `jmena` obsahuje všechna jména, která byla přijata ze serveru v rámci odpovědi na dotaz programu. Jména některých uživatelů však mohou chybět: do přijetí odpovědi 366, která signalizuje konec názvů členů, není seznam úplný. Proto poslední řádek – `jmena += seznam_jmen.split(' ')` – nově přijatá jména k existujícímu seznamu připojuje, místo aby seznam úplně přepsal: při každém spuštění této části kódu program pravděpodobně přijme pouze část celého seznamu členů. Chcete-li jazyku Python sdělit, jak má postupovat po získání kompletního seznamu, zadejte tyto řádky:

```
if kod_odpovedi == RPL_ENDOFNAMES:
    # Zobrazí jmena
    print '\r\nUzivatele pro %(kanal)s:' % irc
    for jmeno in jmena:
```

```
print jmeno
jmena = []
```

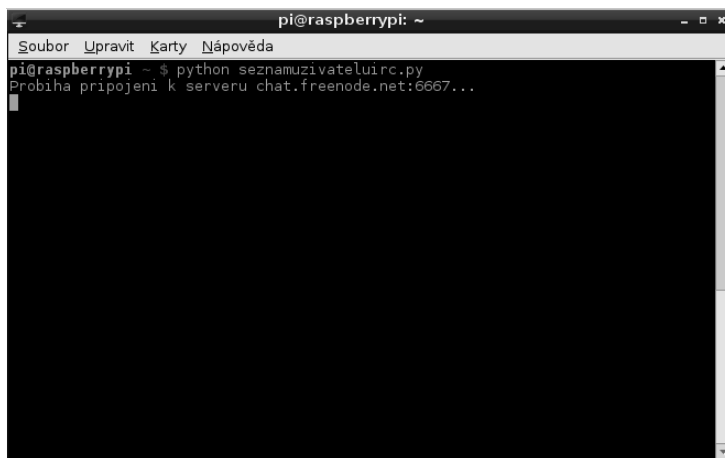
Z toho jazyk Python pozná, že po přijetí odpovědi 366 má vypsat nyní úplný seznam jmen na standardní výstup a poté seznam `jmena` znovu vyprázdnit. Tento poslední řádek – `jmena = []` – je důležitý, protože bez něj by při každém průchodu cyklem program přidal uživatelská jména do seznamu, i když by se tam již nacházela z předchozího cyklu.

Program ukončete zadáním následujících řádků:

```
time.sleep(irc['intervaljmen'])
s.send('NAMES %(kanal)s\r\n' % irc)
```

Tím požádáte interpret jazyka Python, aby před odesláním dalšího požadavku na uživatelská jména a dalším průchodem cyklu vyčkal tolik sekund, kolik je nastaveno v proměnné `intervaljmen`. Dbejte na to, abyste proměnnou `intervaljmen` nastavili na rozumnou hodnotu. Pokud server IRC v krátkém časovém intervalu přijme příliš mnoho požadavků, může program nuceně odpojit kvůli *přetěžování* (flooding).

Uložte program pod názvem `seznamuzivateluiirc.py` a spusťte jej buď pomocí příkazu Run Module (Spustit modul) z nabídky Run v prostředí IDLE, nebo zadáním příkazu `python seznamuzivateluiirc.py` v okně terminálu. Při prvním spuštění programu může připojení k serveru chvíli trvat. Jakmile je však program připojen, měl by se seznam jmen (viz obrázek 11.7) aktualizovat rychle. Chcete-li program ukončit, stiskněte klávesy Ctrl+C.



Obrázek 11.7: Výpis seznamu uživatelů kanálu IRC pomocí programu jazyka Python

Úplný výpis programu, který vypisuje seznam uživatelů IRC, je uveden v Příloze A, „Recepty jazyka Python“, a na stránce knihy na adrese <http://knihy.cpress.cz/K2123> v sekci Soubory ke stažení. Když si zdrojový kód stáhnete z webu, ušetříte si sice trochu psaní, ale při ručním zadávání kódu si můžete ověřit, zda jednotlivým částem programu rozumíte.

Další zdroje informací

V této kapitole jste snad získali představu o možnostech jazyka Python. Obrázek však zdaleka není úplný – abychom seznámili s jazykem neošidili, musela by být kniha mnohem tlustší. K dispozici je však mnoho informačních zdrojů, které slouží k výuce jazyka Python. Patří k nim:

- Oficiální příručka *Beginner's Guide to Python*, která je k dispozici na adrese <http://wiki.python.org/moin/BeginnersGuide>.
- Bezplatný interaktivní kurz, který kompletně funguje v prohlížeči a je dostupný ke stažení na adrese <http://www.learnpython.org/>.
- Kniha *Learn Python The Hard Way*, kterou napsal Zed A. Shaw, (Shavian Publishing, 2012), poskytuje vynikající přehled optimálních programátorských postupů v jazyce Python. Navzdory svému názvu se hodí i pro začátečníky. Tuto knihu si můžete koupit nebo si ji přečíst zdarma na adrese <http://learnpythonthehardway.org/>.
- Kniha *Dive Into Python*, kterou napsal Mark Pilgrim (APRESS, 2004), je sice již poněkud zastaralá a od jejího vydání již vyšla nová verze *Dive Into Python 3* (APRESS, 2009), ale i nadále slouží jako kvalitní příručka základů programování v jazyce Python. Bezplatně si ji můžete stáhnout na adrese <http://www.diveintopython.net/>, případně si ji můžete zakoupit v knihkupectvích.
- Pokud dáváte přednost praktickému učení v komunitě jiných zájemců, naleznete seznam místních uživatelských skupin jazyka Python (Python User Group – někdy se označují jako PIGgies, tj. „prasátka“) na adrese <http://wiki.python.org/moin/LocalUserGroups>.
- Chcete-li se seznámit s knihovnou pygame, skvělý úvod s praktickými příklady představuje kniha *Making Games with Python & Pygame*, kterou napsal Al Sweigart, (CreateSpace, 2012). Můžete si pořídit vázané vydání nebo si knihu zdarma stáhnout na adrese <http://inventwithpython.com/>.
- Pokud hledáte zdroje v českém jazyce, mohla by vás zaujmout kniha *Python 3 Výukový kurz* (Computer Press 2010). Zakoupit ji můžete na adrese <http://knihy.cpress.cz/K1747>.

Hackování hardwaru

V předchozích kapitolách jste se dozvěděli, jak může počítač Raspberry Pi sloužit jako pružná platforma ke spouštění různých typů programů. V tom ale není výjimečný: stejný software lze spustit i ve stolních počítačích i noteboocích a v mnoha případech tam bude fungovat rychleji, než jej dokáže provozovat úsporný procesor počítače Pi.

Počítač Pi však poskytuje něco, čím převyšuje možnosti průměrného počítače PC: svůj port *GPIO* (*general-purpose input-output*) s 26 piny, který je umístěn v levém horním rohu jeho desky s tištěnými spoji.

Díky portu GPIO může počítač Pi komunikovat s jinými komponentami a obvody a může sloužit jako řadič většího elektronického obvodu. Pomocí portu GPIO může počítač Pi měřit teploty, ovládat motory a vyměňovat si data s jinými výpočetními zařízeními pomocí různých protokolů, k nimž patří *SPI* (*Serial Peripheral Interface*) a *I²C* (*Inter-Integrated Circuit*).

Chcete-li však začít budovat obvody, které budou postaveny na portu GPIO počítače Pi, budete nejdříve potřebovat jisté dodatečné vybavení.

Elektronické příslušenství

Chcete-li stavět obvody, které by bylo možné řídit pomocí portu GPIO počítače Pi, potřebujete k tomu různé komponenty a nástroje. Následující přehled může začínajícím zájemcům o elektroniku poradit, co si mají koupit:

- **Zkušební model obvodu** – elektronický zkušební model obvodu je deska s mřížkou otvorů, které mají rozteč 2,54 mm a lze do nich vkládat a vyjímat z nich komponenty. Pod každou mřížkou se nachází řada elektrických kontaktů, které umožňují propojit komponenty ve stejné řadě i bez vodičů. Zkušební model obvodu je cenný nástroj pro elektronickou práci, protože dovoluje snadno vytvářet testovací obvody, které lze měnit bez nutnosti pájení a odpájení.
- **Vodiče** – zkušební model obvodu sice umožňuje propojit některé komponenty bez vodičů, ale vodiče jsou i nadále potřeba k propojení jednotlivých řad. Označují se jako *propojovací kabely* (*jumper wire*), a pracujete-li na zkušebním modelu obvodu, je vhodné pořídit si místo kabelů s pleteným jádrem raději kabely s plným jádrem. Kabely s plným jádrem se snadněji vkládají do otvorů zkušebního modelu než kabely s pleteným jádrem.

Kabely si kupte v různých barvách, abyste mohli jednotlivé spoje barevně označovat podle jejich funkce.

- **Rezistory** – naprostá většina elektrických obvodů obsahuje komponenty označované jako *rezistory* (odpory) a výjimkou nejsou ani ukázkové projekty v této kapitole. Rezistory se měří v *ohmech*, které se zapisují symbolem Ω (řecké písmeno omega). Snažte se mít vždy k dispozici několik kusů od každé běžné hodnoty: pro začátek se hodí například hodnoty 2,2 k Ω , 10 k Ω a 68 Ω . Někteří prodejci nabízejí *sady rezistorů*, které zahrnují širokou řadu užitečných hodnot.
- **Tlačítka** – značně rozšířená vstupní komponenta zvaná *tlačítko* (push-button) po svém stisknutí uzavře elektrický obvod. Klávesnice na nejzákladnější úrovni není nic jiného než sada takových tlačítek. Pokud navrhujete obvod, který bude počítači Pi poskytovat jednoduchý vstup, zvolte tlačítko označené *dočasný přepínač* (momentary switch).
- **Diody LED** – *diody LED* (light-emitting diode) jsou nejrozšířenějším výstupním zařízením vůbec. Dioda LED se rozsvítí, když na ni přivedete napětí. Poskytuje tak vizuální zpětnou vazbu, zda má pin portu GPIO počítače Pi vysoké či nízké napětí. Při nákupu diod LED pro použití s počítačem Pi volte diody s nízkou spotřebou. Port GPIO nemá příliš vysoký výkon a diody LED pro vysoký proud (jako jsou jasné modré či jasné bílé typy) vyžadují externí zdroj napájení a dodatečnou komponentu s názvem *tranzistor*.

Jestliže máte po dokončení svého prototypu založeného na zkušebním modelu obvodu v plánu vytvořit trvalejší obvod (postup si ukážeme v další části této kapitoly), budete ještě potřebovat tyto pomůcky:

- **Stripboard** – tuto desku lze popsat jako zkušební model obvodu na jedno použití. Stejně jako u zkušebních modelů jsou otvory uspořádány v mřížce s roztečí 2,54 mm. Na rozdíl od zkušebních modelů je však komponenty nutné na jejich místa připájet. Získáte tím trvalý elektronický obvod.
- **Páječka** – potřebujete-li komponentu trvale připojit k obvodu, musíte ji připájet. Za páječku nemusíte utratit celé jmění, ale pokud si můžete dovolit model s kontrolou teploty, jedná se o rozumnou investici. Dbejte na to, aby měla zakoupená páječka malý a špičatý hrot. Páječky s dlátovitou špicí se pro manipulaci s jemnou elektronikou nehodí.
- **Pájka** – páječka ke své práci potřebuje pájku. Pájka je směs vodivých kovů a čisticí látky, která se nazývá tavidlo (flux). Zakoupená pájka by měla být určena pro práci s elektronikou. Silná instalatérská pájka je sice levná, ale může poškodit citlivé obvody, protože ke svému tavení vyžaduje příliš mnoho tepla.
- **Stojan a houbka** – je důležité, abyste měli kam položit horkou páječku, když ji nepoužíváte, a abyste mohli při práci čistit její hrot. Některé páječky se dodávají spolu se stojanem, který je vybaven čisticí houbou. Pokud jste takový doplněk se svou páječkou nedostali, kupte si podobnou sadu samostatně.
- **Kleště** – komponenty zasunované do otvorů mají dlouhé nohy, které po svém připájení vyčnívají na druhé straně. Pomocí *kleští* (side cutters) můžete tyto přebytečné nohy zkrátit, aniž byste přitom poškodili spájený spoj.

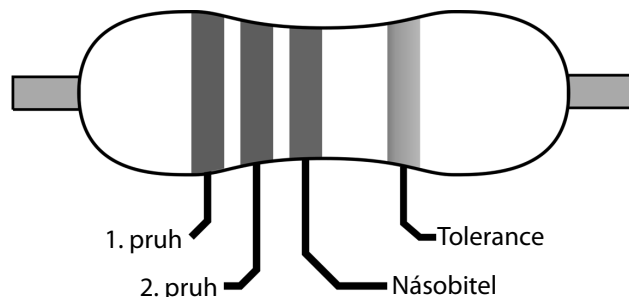
- **Pinzeta** – elektronické komponenty bývají titěrné a těžko se s nimi manipuluje. Bude se vám proto hodit dobrá pinzeta. Pokud místo snáze použitelných komponent zasunovaných do otvorů uvažujete o práci s komponentami, které se upevňují na povrchu, bude pinzeta naprosto nezbytná. Když se totiž pokusíte držet komponentu v ruce a zároveň ji pájet, bez pinzety si popálíte prsty!
- **Pracovní stojan** – tyto stojany, kterým se často říká *třetí ruka*, jsou zatížené držáky se svorkami, které stabilně drží pájený předmět. Některé pracovní stojany jsou vybaveny integrovanou lupou, která umožňuje práci s drobnými předměty. Nejdražší modely pak poskytují i osvětlení pracovní plochy.
- **Multimetr** – multimetry jsou testovací měřáky s více funkcemi včetně měření napětí, odporu a kapacity spolu s testováním souvislosti, aby bylo možné najít místa přerušení obvodu. Multimetr sice není zcela nezbytný, ale může být velmi užitečný při diagnostice potíží s obvody. Profesionální jednotky multimetrů mohou být poměrně drahé, ale jednoduché modely nestojí mnoho a jedná se o rozumný výdaj pro každého, kdo v oblasti elektroniky začíná.
- **Odsávací knot** – chyby se sice stávají, ale nemusí být trvalé. Odsávací knot (desoldering wick) je splétaná kovová páska. Když tento knot umístíte nad spájený spoj a zahřejete, odsaje pájku pryč z komponenty. S trochou praxe je možné pomocí odsávacího knotu zachránit komponenty z vyřazených elektronických obvodů. Tímto způsobem můžete levně získávat běžné součástky.

Čtení barevných kódů rezistorů

Většina elektronických komponent je jasně označena. Například přímo na *kondenzátorech* je vytištěna hodnota jejich *kapacitance* v jednotkách *faradů*, zatímco na *krystalcích* je označena jejich *frekvence*.

Důležitou výjimku představují rezistory, které obvykle na svém povrchu nemají žádný text. Jejich hodnotu rezistance v ohmech lze místo toho odvodit z barevných pruhů, které zdobí povrch součástky. Každému hardwarovému hackerovi přijde vhod, když se naučí tyto pruhy dekódovat, protože po vyjmutí rezistoru z jeho obalu lze jeho hodnotu rezistance zjistit pouze pomocí multimetru, který je pro tento úkol značně nepohodlný a pomalý.

Barevné kódy rezistorů naštěstí dodržují logickou strukturu. Obrázek 12.1 znázorňuje typický rezistor se čtyřmi pruhy. Barevnou verzi tohoto diagramu s vysokým rozlišením naleznete na stránce knihy na adrese <http://knihy.cpress.cz/K2123> v sekci Soubory ke stažení. Barva prvních dvou pruhů odpovídá hodnotě rezistance v ohmech. Třetí pruh slouží jako *násobitel*, kterým musíte vynásobit první dvě čísla, abyste dostali skutečnou hodnotu rezistance. Poslední pruh určuje *toleranci* rezistoru, tj. udává, nakolik je hodnota přesná. Rezistor s nižší tolerancí bude blíže své nominální hodnotě než rezistor s vyšší tolerancí, ale zároveň bude taková komponenta stát více.



	1. a 2. pruh	Násobitel	Tolerance
ČERNÝ	0	0	-
HNĚDÝ	1	$\times 10^1$	$\pm 1\%$
ČERVENÝ	2	$\times 10^2$	$\pm 2\%$
ORANŽOVÝ	3	$\times 10^3$	-
ŽLUTÝ	4	$\times 10^4$	-
ZELENÝ	5	$\times 10^5$	$\pm 0.5\%$
MODRÝ	6	$\times 10^6$	$\pm 0.25\%$
FIALOVÝ	7	$\times 10^7$	$\pm 0.1\%$
ŠEDÝ	8	$\times 10^8$	$\pm 0.05\%$
BÍLÝ	9	$\times 10^9$	-
ZLATÝ	-	$\times 10^{-1}$	$\pm 5\%$
STŘÍBRNÝ	-	$\times 10^{-2}$	$\pm 10\%$
ŽÁDNÝ	-	-	$\pm 20\%$

Obrázek 12.1: Rezistor se čtyřmi pruhy a dekodovací tabulka jeho barevného značení

Chcete-li zjistit hodnotu ukázkového rezistoru, nejdříve přečtete dva pruhy rezistance počínaje tím vlevo: v tomto případě mají oba červenou barvu. Podle tabulky na obrázku 12.1 odpovídá červená barva hodnotě 2, takže získáte číslo 22. Další pruh je zelený. Jedná se o násobitel a jeho barva znamená 105 neboli 100 000 (jedničku a pět nul). Když hodnotu 22 vynásobíte číslem 100 000, dostanete výsledek 2 200 000, což je hodnota rezistance v ohmech: 2 200 000 Ω .

1 000 ohmů tvoří jeden *kiloohm* a 1 000 kiloohmů dává jeden *megaohm*. Hodnotu 2 200 000 Ω bychom tedy obvykle zapsali jako 2,2 M Ω . Poslední pruh na pravé straně rezistoru je zlatý a určuje toleranci čili přesnost rezistoru, která je v případě na obrázku 12.1 plus minus 5 procent.

Existují také rezistory s pěti pruhy. Lze je číst stejným způsobem jako rezistory se čtyřmi pruhy, ale s tím rozdílem, že první tři pruhy znamenají hodnoty rezistance, čtvrtý je násobitel a pátý slouží k odečtu tolerance.

Získávání komponent

Pokud jste se zatím elektronikou nezabývali, můžete mít ze začátku potíže s hledáním komponent a nástrojů. Naštěstí existuje mnoho zdrojů online i offline, které se specializují na těžko dostupné komponenty, jaké můžete k dokončení svého projektu potřebovat.

Zdroje online

Mezi největší maloobchodní prodejce elektronických komponent a nástrojů na světě patří společnosti *RS Components* a *Farnell*. Obě mají pobočky a sklady v mnoha zemích a obě nabízejí bohatou paletu hardwarových součástek. Pravděpodobně jste již alespoň u jednoho z těchto prodejců nakoupili. V době psaní této knihy byly společnosti *RS Components* a *Farnell* jedinými dodavateli, kterým nadace Raspberry Pi Foundation poskytla licenci na výrobu hardwaru Raspberry Pi. Jestliže tedy tuto knihu nečtete v době, kdy nákup počítače Raspberry Pi teprve zvažujete, objednali jste si jej u jedné z výše uvedených firem.

Společnosti *RS Components* i *Farnell* se zaměřují na transakce typu *B2B* (business-to-business), což znamená, že hlavní zdroj jejich příjmů představují výrobci elektroniky, kteří nakupují komponenty ve velkých množstvích. Obě společnosti však ochotně umožňují spotřebitelům, aby se zaregistrovali na jejich webových stránkách a zadávali objednávky v rozsahu pouhých jednotlivých komponent.

Při zadávání malé objednávky však nezapomeňte, že společnost může účtovat další poplatky. *RS Components* zpoplatňuje službu dodání následujícího dne, a ačkoli je sazba v případě velkých objednávek poměrně příznivá, u malých objednávek může značně převyšovat cenu zakoupených komponent. Společnost *Farnell* oproti tomu neúčtuje dodání následujícího dne, ale vyžaduje, aby objednávky zadané na jejím webu dosahovaly určité minimální celkové hodnoty.

Podle místa svého pobytu si můžete položky objednané u společností *RS Components* nebo *Farnell* vyzvednout na jedné z jejich poboček. Přitom ušetříte poštovné a zboží dostanete rychleji, ale tato možnost není k dispozici všude. Webové stránky obou prodejců naleznete na těchto adresách:

RS Components: <http://www.rs-online.com>

Farnell: <http://www.farnell.com>

Zdroje offline

Může se stát, že komponentu potřebujete okamžitě a ani dodání druhého dne není dostatečně rychlé. Případně můžete požadovat jediný rezistor nebo malý kousek drátu, takže se vám nevyplatí hradit vysokou cenu za objednávku nebo objednávat v minimální hodnotě u jednoho z prodejců online. Naštěstí existují také kamenné obchody, které se specializují na elektronické komponenty. Těchto obchodů sice není k dispozici tolik jako v předchozích desetiletích, ale

ve většině velkých měst stále najdete alespoň jednu provozovnu, která má na skladě nejčastěji používané nástroje a komponenty.

Nejoblíbenější řetězec obchodů s elektronickými komponentami v Británii se nazývá Maplin Electronics. Tato společnost založená v Essexu roku 1972 se rozrostla z malého zásilkového obchodu na 187 obchodů po celé Británii. Obchody Maplin Electronics fungují ve většině velkých měst a mohou sloužit jako praktický zdroj běžných hardwarových součástek.

Společnost Maplin Electronics poskytuje také poštovní zásilkovou službu a službu rezervace na svém webu, ale mějte na paměti, že se zaměřuje na obchody typu B2C (business-to-consumer). Mnoho komponent objednaných u společnosti Maplin stojí značně více než u firem RS Components či Premier Farnell, protože společnost Maplin nedosahuje zisku pomocí velkých objemů obchodů, ale hlavně díky vyšším maržím.

V USA a jiných zemích si postavení nejoblíbenějšího řetězce obchodů s elektronikou drží společnost Radio Shack. Tato firma, která byla založena roku 1921 a může se pochlubit více než 7 150 obchody po celém světě, drží skladem bohatý výběr běžných elektronických komponent a nástrojů, které si můžete koupit osobně nebo si jejich zaslání objednat na firemním webu.

Stejně jako Maplin Electronics v Británii podniká také Radio Shack v sektoru B2C. Z tohoto důvodu za komponenty pořízené v obchodě Radio Shack obvykle zaplatíte více, než kdybyste je objednali u společnosti RS Components či Farnell. Díky velkému počtu obchodů Radio Shack se však jedná o praktickou volbu pro kupující, kteří spěchají nebo shánějí jednotlivé položky.

Výhodou obchodů Maplin a Radio Shack je také to, že v nich pracují lidé, kteří vám budou rozumět. Mnoho zaměstnanců obou společností má rozsáhlé znalosti elektroniky, a pokud si nebudete jisti, které součástky budete pro svůj projekt potřebovat, ochotně vám nabídnou svou radu a pomoc. Webové stránky obou prodejců naleznete na těchto adresách:

Maplin Electronics: <http://www.maplin.co.uk>

Radio Shack: <http://www.radioshack.com>

Firmy specializované na nadšence

Kromě velkých řetězců fungují také menší firmy, které se zaměřují na fandý do elektroniky. Rozsah jejich sortimentu se sice nemůže rovnat nabídce velkých řetězců, ale oproti nim jsou nabízené položky obvykle pečlivě vybrány a lidé, kteří firmu provozují, je osobně doporučují.

Mnoho z těchto obchodů pro nadšence se objevilo v reakci na projekt open source s názvem Arduino, který vytváří výukovou platformu prototypování mikrokontrolérů. Vzhledem k tomu, že počítač Raspberry Pi je zaměřen přibližně na stejnou cílovou skupinu jako Arduino (i když je určen k jiným úkolům), většina kromě svých běžných produktových řad zkoumá i podporu počítače Pi.

Nákup od hobby specialistů má několik výhod. Pokud u prodáváných produktů deklarují, že spolupracují s počítačem Pi, byly s tímto ohledem testovány, a získáváte tak mnohem větší jistotu. Několik společností také navrhuje svůj vlastní rozšiřující hardware pro různé platfor-

my a počítač Pi není výjimkou. Tato zařízení, která mají řešit požadavky komunity, mohou zahrnovat přídavné porty nebo dodatečný hardware, případně mohou jinak rozšiřovat funkce cílového zařízení.

V Británii je jedním z nejoblíbenějších hobby specialistů firma oomlout. Založili ji nadšenci projektů open source, kteří pracují s hardwarem Arduino, a jedná se o vynikající zdroj rozšiřujících sad i běžných komponent včetně tlačítek, displejů a tranzistorů. Na rozdíl od velkých prodejců firma oomlout vybavuje své komponenty všemi potřebnými doplňky – například pull-up rezistory k tlačítkům – a schémata obvodů kvůli snadnějšímu sestavení. Tam, kde je to možné, je k dispozici také ukázkový zdrojový kód, abyste mohli vše co nejrychleji zprovoznit.

V USA poskytuje podobnou službu firma Adafruit. Tato firma, založená s cílem zpřístupnit rozšíření open source pro základní desky Arduino, nabízí širokou škálu komponent a sad, včetně jedné z prvních rozšiřujících desek určených speciálně pro počítač Raspberry Pi (další informace naleznete v kapitole 13, „Rozšiřující desky“).

Webové stránky obou prodejců naleznete na těchto adresách:

oomlout: <http://www.oomlout.co.uk>

Adafruit: <http://www.adafruit.com>

Port GPIO

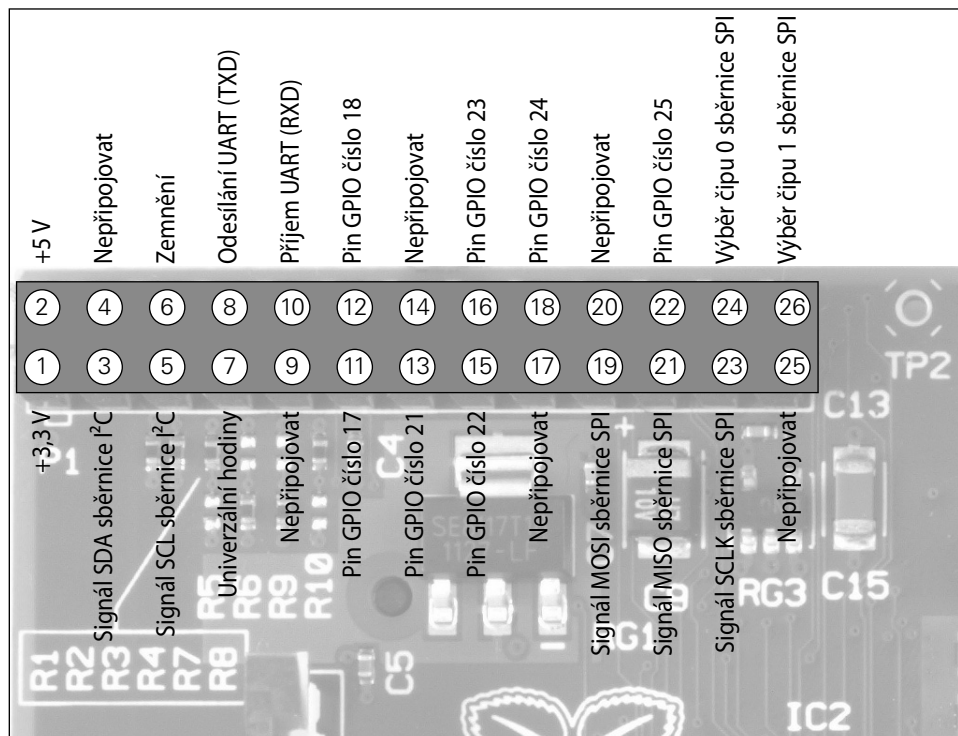
Port GPIO počítače Raspberry Pi je umístěn v levém horním rohu desky s tištěnými spoji a poznáte jej podle označení P1. Tento port má 26 pinů, které jsou z výroby umístěny do dvou řad po 13 *samčích hrotů s roztečí 2,54 mm*. Rozteč těchto řad je mimořádně důležitá: rozestupy pinů po 2,54 mm (0,1 palce v anglosaských jednotkách) se v elektronice objevují velmi často. Jedná se o standardní vzdálenost v platformách prototypování, které zahrnují desky stripboard a zkušební modely obvodů.

Každý pin portu GPIO má vlastní účel a několik pinů přitom společně vytváří určitý obvod. Rozložení portu GPIO je znázorněno na obrázku 12.2.

Piny portu GPIO v obou řadách se číslovají střídavě, přičemž piny v dolní řadě mají lichá čísla a v horní řadě sudá. Při práci s portem GPIO počítače Pi je potřeba mít na paměti, že většina jiných zařízení používá odlišné systémy číslování pinů. Vzhledem k tomu, že samotný počítač Pi piny nijak neoznačuje, můžete někdy váhat, který pin je který.



Upozornění: K pinům s názvem Do Not Connect nikdy nic nepřipojujte. Jsou rezervovány pro interní funkce hardwaru SoC BCM2835 počítače Pi. Když k těmto pinům cokoli připojíte, počítač Pi tím poškodíte.



Obrázek 12.2: Port GPIO počítače Raspberry Pi a definice jeho portů

Port GPIO počítače Pi sice na pinu 2 poskytuje zdroj napájení 5 V, který čerpá z dodávky napájení rozbočovače mikro USB, ale interně počítač Pi funguje s logikou napájení 3,3 V. To znamená, že komponenty počítače Pi vyžadují zdroj napájení 3,3 V. Chcete-li vytvořit obvod, který bude s počítačem Pi komunikovat prostřednictvím jeho portu GPIO, nezapomeňte použít komponenty, které jsou kompatibilní s logikou 3,3 V nebo které před propojením s počítačem Pi obcházejí obvod přes *regulátor napětí* (voltage regulator).



Upozornění: Pokud připojíte zdroj napájení 5 V k libovolnému pinu portu GPIO počítače Raspberry Pi nebo pokud zkratujete některý z pinů dodávky napájení (pin 1 a pin 2) s libovolným jiným pinem, poškodíte tím počítač Pi. Vzhledem k tomu, že port je přímo propojen s piny procesoru Broadcom BCM2835 typu SoC, nebude možné případné poškození tohoto procesoru opravit. Při práci kolem portu GPIO vždy dbejte mimořádné opatrnosti.

Port GPIO standardně poskytuje sedm pinů pro obecné použití: jedná se o piny číslo 11, 12, 13, 15, 16, 18 a 22. Pin 7, který standardně poskytuje hodinový signál pro obecné použití, lze alternativně také použít jako univerzální pin, takže je celkem k dispozici osm pinů. Tyto piny lze přepínat mezi dvěma stavy: *vysoký*, kdy poskytují kladné napětí 3,3 V, a *nízký*, kdy odpovídají zemnění neboli napětí 0 V. Tyto stavy odpovídají hodnotám 1 a 0 binární logiky a je

pomocí nich možné zapínat a vypínat jiné komponenty. Další informace o této problematice naleznete v následujících částech této kapitoly.



Upozornění: Interní logika počítače Pi je založena na napětí 3,3 V. Tím se počítač Pi liší od mnoha běžných zařízení s mikrokontrolérem, jako je oblíbené Arduino a jeho varianty, které obvykle pracují s 5 V. Zařízení navržená pro Arduino nemusí s počítačem Pi fungovat, pokud mezi nimi není zapojen *převaděč úrovně* (level translator) či *optický izolátor* (optical isolator). Podobně platí, že obvod s připojením pinů mikrokontroléru s napětím 5 V přímo k portu GPIO počítače Raspberry Pi nebude fungovat a může přitom nastat trvalé poškození počítače Pi.

Kromě těchto univerzálních pinů má port GPIO piny, které jsou vyhrazeny pro určité *sběrnice* (bus). Jednotlivé sběrnice popíšeme v následujících podkapitolách.

Sériová sběrnice UART

Sériová sběrnice UART (Universal Asynchronous Receiver/Transmitter) poskytuje jednoduché sériové rozhraní se dvěma vodiči. Pokud je v souboru `cmdline.txt` (viz popis v kapitole 6, „Konfigurace počítače Raspberry Pi“) nastaven sériový port, používá se jako port pro zprávy právě tato sériová sběrnice. Připojíte-li sériovou sběrnici UART počítače Pi k zařízení, které dokáže data zobrazovat, můžete sledovat zprávy jádra systému Linux. Jestliže máte potíže se spuštěním počítače Pi, může se jednat o praktický diagnostický nástroj – zejména v situacích, kdy se na displeji nic nezobrazuje.

Sériová sběrnice UART je dostupná na pinech 8 a 10, přičemž pin 8 přenáší signál *odesílání* (transmit) a pin 10 signál *příjem* (receive). Rychlost je možné nastavit v souboru `cmdline.txt` a obvykle má hodnotu 115 200 bitů za sekundu (b/s).

Sběrnice I²C

Jak je zřejmé z jejího názvu, *sběrnice I²C (Inter-Integrated Circuit)* má zajistit komunikaci mezi více *integrovanými obvody* (IC – Integrated Circuit). V případě počítače Pi patří mezi tyto integrované obvody procesor Broadcom BCM2835 typu SoC, který je jádrem systému. Tyto piny poskytují přístup k *pull-up rezistorům*, kterými je počítač Pi vybaven. Přístup k funkcím sběrnice I²C tedy nevyžaduje žádné externí rezistory.

Ke sběrnici I²C lze přistupovat na pinech 3 a 5, kde pin 3 poskytuje signál *SDA (Serial Data Line)* a pin 5 signál *SCL (Serial Clock)*. Sběrnice I²C dostupná na těchto pinech je ve skutečnosti pouze jednou ze dvou takových sběrnic, které vlastní čip BCM2835 poskytuje, a označuje se jako I²C0. Druhá sběrnice I²C1 je ukončena na rezistorech desky s plošnými spoji počítače Raspberry Pi a pro univerzální použití není k dispozici.

Sběrnice SPI

Sběrnice SPI (Serial Peripheral Interface) je synchronní sériová sběrnice, která je určena zejména pro *interní systémové programování* (ISP – in-system programming) mikrokontrolérů a jiných zařízení. Na rozdíl od sběrnic UART a I²C se jedná o sběrnici se čtyřmi vodiči s více linkami *výběru čipu* (Chip Select), které jí dovolují komunikovat s více než jedním cílovým zařízením.

Sběrnice SPI počítače Pi je dostupná na pinech 19, 21 a 23 a její dvojice linek výběru čipu je přivedena na piny 24 a 26. Pin 19 poskytuje signál *SPI MOSI (Master Output, Slave Input)*, pin 21 signál *SPI MISO (Master Input, Slave Output)*, pin 23 signál *SLCK (Serial Clock)*, který umožňuje synchronizovat komunikaci, a piny 24 a 26 vedou signály *výběru čipu* (Chip Select), aby bylo možné zvolit jedno z nejvýše dvou nezávislých vedlejších zařízení.

Procesor BCM2835 typu SoC v počítači Raspberry Pi sice obsahuje i další sběrnice, které však nejsou vyvedeny na port GPIO, a nelze s nimi tedy pracovat.

Použití portu GPIO v jazyce Python

Když jste se seznámili s teorií, je načase pustit se do praxe. V této části se dozvíte, jak nainstalovat *knihovnu*, která zajistí snadný přístup k univerzálním pinům portu GPIO počítače Raspberry Pi z programů jazyka Python. Podíváte se také na dva jednoduché elektronické obvody, které demonstrují, jak lze používat port GPIO ke vstupu a výstupu.

Jak jste zjistili v kapitole 11, „Úvod do jazyka Python“, jazyk Python je uživatelsky přátelský, ale zároveň výkonný. Nepředstavuje však ideální volbu v každé situaci. Pro jednoduché obvody, které budete zapojovat v této kapitole, sice funguje dobře, ale neposkytuje vlastnost, která se označuje jako *fungování v deterministickém reálném čase*. Většině uživatelů to nebude vadit, avšak pokud plánujete počítač Pi nasadit k ovládání jaderného reaktoru nebo komplexní robotické platformy, možná byste měli uvažovat o jazyku nižší úrovně jako C++, nebo dokonce *assembler* a programy v těchto jazycích spouštět v mikrokontroléru fungujícím v reálném čase.

Jestliže váš projekt vyžaduje skutečnou činnost v reálném čase, počítač Pi pravděpodobně nebude vhodný. Místo toho můžete zvolit platformu mikrokontroléru, jako je oblíbený projekt open source s názvem Arduino, nebo některý z mikrokontrolérů řady MSP430 od společnosti Texas Instruments. Obě tato zařízení, která lze s počítačem Pi propojit buď pomocí řady pinů portu GPIO, nebo prostřednictvím sběrnice USB, poskytují specializované prostředí fungující v reálném čase pro řízení a detekci.

Instalace knihovny jazyka Python pro port GPIO

Od uvedení počítače Pi na trh vytvořilo mnoho vývojářů softwarové moduly označované jako *knihovny*, které umožňují plně využít různých funkcí tohoto počítače. Programátoři konkrétně vyhověli požadavkům uživatelů počítače Pi, kteří chtějí přistupovat k portu GPIO, aniž by se museli zabývat nízkourovňovým programováním.

Tyto knihovny jsou navrženy tak, aby rozšiřovaly možnosti standardního jazyka Python podobně jako knihovna `pygame`, kterou jsme popsali v kapitole 11, „Úvod do jazyka Python“. Po instalaci některé z těchto knihoven získá jazyk Python schopnost snadno adresovat port GPIO počítače Pi, ačkoli to zároveň znamená, že tuto knihovnu si bude muset stáhnout a nainstalovat kdokoli, kdo bude chtít pracovat s vaším softwarem.

K dispozici je několik knihoven jazyka Python pro port GPIO, ale pro účely této části doporučujeme, abyste použili knihovnu `raspberry-gpio-python`, která byla v době psaní této knihy ve verzi 0.2.0. Knihovna je přístupná na webu Google Code na následující adrese: <http://code.google.com/p/raspberry-gpio-python/>.

Knihovnu jazyka Python si sice můžete stáhnout přímo z webového prohlížeče, ale mnohem rychlejší je provést to z okna terminálu v rámci procesu instalace. Postupujte takto:

1. V nabídce Accessories počítače Raspberry Pi otevřete okno terminálu, případně použijte konzolu, pokud jste nespustili grafické prostředí.
2. Zadejte příkaz `wget http://raspberrypi-gpio-python.googlecode.com/files/RPi.GPIO-0.2.0.tar.gz`, abyste knihovnu stáhli do svého domovského adresáře. Pokud byla vydána novější verze, nahraďte číslo verze (0.2.0) aktuální verzí.
3. Příkazem `tar xvfz RPi.GPIO-0.2.0.tar.gz` extrahujte obsah souboru. Tento příkaz rozlišuje velká a malá písmena, takže název zadejte včetně velkých písmen.
4. Zadejte příkaz `cd RPi.GPIO-0.2.0` pro přechod do nově vytvořeného adresáře. Opět platí, že pokud jste stáhli novější verzi knihovny, musíte nahradit původní číslo verze číslem stažené verze.
5. Nainstalujte knihovnu do prostředí jazyka Python příkazem `sudo python setup.py install`.

Knihovna GPIO je sice nyní nainstalována v prostředí Python, ale nebude načtena automaticky. Podobně jako knihovnu `pygame` je potřeba i tuto knihovnu do programu explicitně importovat. Chcete-li použít knihovnu, začněte svůj program řádkem `import RPi.GPIO as GPIO`. Další informace o tomto importu naleznete v následujících příkladech.



Upozornění: Port GPIO počítače Raspberry Pi neposkytuje žádnou ochranu před přepětími ani elektrickými zkraty. Než připojíte svůj obvod k počítači Pi, vždy zkontrolujte, zda je zapojen správně. Je-li to možné, použijte izolační desku, jako je Gertboard (viz popis v kapitole 13, „Rozšiřující desky“), která zajistí ochranu počítače Pi.

Výpočet mezních hodnot rezistorů

Diody LED se neobejdou bez *rezistorů omezujících napětí*, které je chrání před spálením. Bez rezistoru bude dioda LED pravděpodobně svítit jen krátce. Poté selže a budete ji muset vyměnit. Jedna věc je vědět, že potřebujete rezistor, ale důležité je také zvolit správnou hodnotu rezistoru pro dané zapojení. Když vyberete rezistor s příliš vysokou hodnotou, dioda LED bude svítit jen tlumeně nebo se nerozsvítí vůbec. V případě příliš nízké hodnoty rezistoru zase můžete diodu spálit.

Chcete-li vypočítat potřebnou hodnotu rezistoru, musíte vědět, jaké hodnoty dosahuje *propustný proud* (forward current) příslušné diody LED. Jedná se o maximální proud, který může dioda LED odebírat bez poškození, a měří se v miliampérech (mA). Potřebujete také znát *propustné napětí* (forward voltage) diody LED. Druhá hodnota měřená ve voltech by měla být 3,3 V nebo nižší. Jestliže je napětí vyšší, bude dioda LED s počítačem Pi použitelná jen s externím zdrojem napájení a prepínacím zařízením, které se nazývá *tranzistor*.

Nejjednodušší způsob, jak vypočítat potřebnou hodnotu rezistance, představuje vzorec $R = (V - F) / I$, kde R je rezistance v ohmech, V je napětí přivedené na diodu LED, F je propustné napětí diody LED a I je maximální propustný proud diody LED v ampérech (příčemž jeden ampér se rovná tisíci mA).

Uvažujeme-li typickou červenou diodu LED s propustným proudem 25 mA a propustným napětím 1,7 V, kterou budeme napájet pomocí napětí 3,3 V dodávaného portem GPIO počítače Pi, můžeme vypočítat potřebnou hodnotu rezistoru jako $(3,3 - 1,7) / 0,025 = 64$. Diodu LED tedy ochrání rezistor s hodnotou 64 Ω nebo silnější. Vypočítaná čísla zpravidla přesně neodpovídají běžně dodávaným rezistorům, takže při výběru rezistoru výsledek vždy zaokrouhlete nahoru, abyste měli jistotu, že je dioda LED chráněna dostatečně. Nejbližší běžně dodávaný rezistor má rezistanci 68 Ω, takže z hlediska ochrany této diody LED vyhovuje.

Pokud hodnoty propustného napětí a propustného proudu své diody LED neznáte (například tehdy, když k diodám nemáte dokumentaci nebo jste je získali z vyřazené elektroniky), raději buďte opatrní a připojte dostatečně velký rezistor. Jestliže dioda LED svítí příliš slabě, můžete svůj odhad korigovat směrem dolů. Spálenou diodu LED však opravit nelze.

Výstup portu GPIO: Blikání diody LED

V prvním příkladu máte za úkol zapojit jednoduchý obvod, který bude obsahovat diodu LED a rezistor. Dioda LED bude vizuálně potvrzovat, že port GPIO počítače Pi skutečně provádí to, co od něj požaduje program jazyka Python. Rezistor přitom bude omezovat proud odebíraný diodou LED, aby byla chráněna před spálením.

K sestavení obvodu budete potřebovat zkušební model obvodu, dva propojovací kabely, diodu LED a vhodný rezistor k omezení napětí (viz popis v rámečku „Výpočet mezních hodnot rezistorů“). Obvod lze sice postavit i bez zkušebního modelu pouhým zkroucením vodičů, ale zkušební model se vyplatí, protože se s ním testovací obvody sestavují a rozebírají velmi snadno.

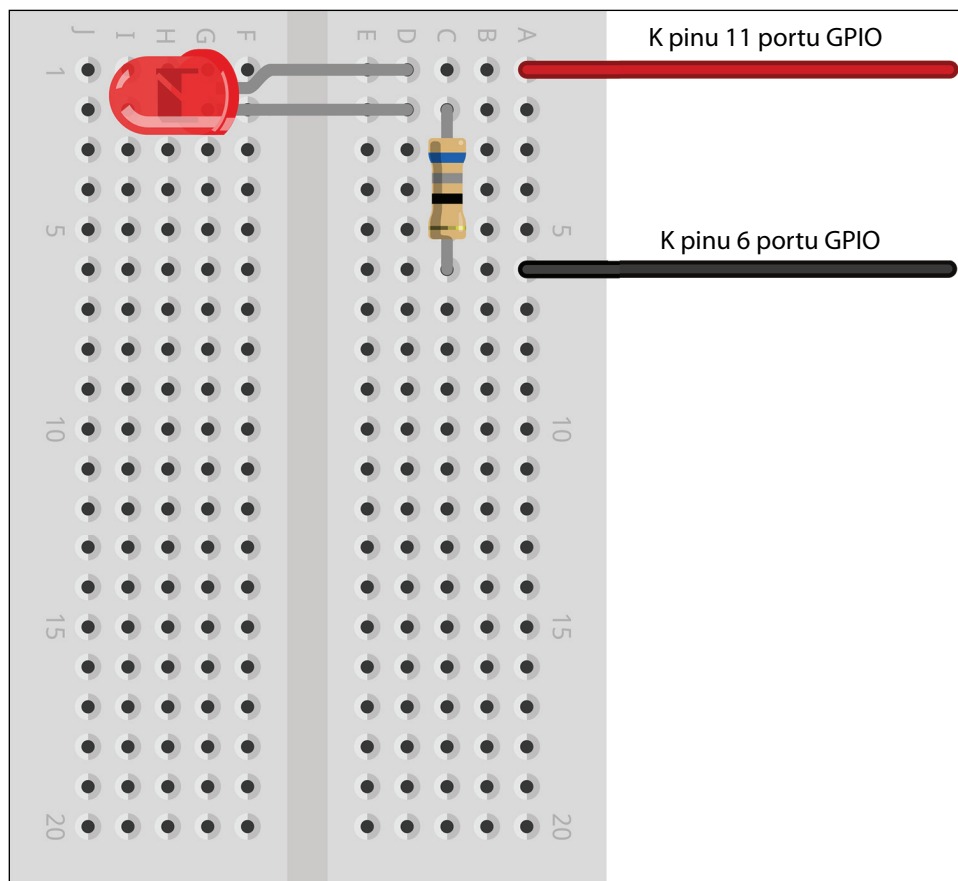
Za předpokladu, že pracujete se zkušebním modelem obvodu, můžete obvod, který bude odpovídat obvodu na obrázku 12.3, zapojit tímto způsobem:

1. Vložte do zkušebního modelu obvodu diodu LED tak, aby její delší noha (tzv. *anoda*) byla zasunuta v jednom řádku a její kratší noha (tzv. *katoda*) v druhém. Pokud zapojíte obě nohy diody LED do stejného řádku, dioda nebude fungovat.

2. Vložte jednu nohu rezistoru do stejného řádku jako kratší nohu rezistoru a druhou nohu rezistoru zapojte do prázdného řádku. Na směr umístění noh rezistoru nezáleží, protože rezistor patří mezi *nepolarizovaná* zařízení (tj. zařízení nerozlišující směr).
3. Pomocí propojovacího kabelu připojte pin 11 portu GPIO počítače Raspberry Pi (nebo odpovídající pin na kartě rozhraní připojené k portu GPIO) ke stejnému řádku, kde je zapojena delší noha diody LED.
4. Dalším propojovacím kabelem připojte pin 6 portu GPIO počítače Raspberry Pi (nebo odpovídající pin na kartě rozhraní připojené k portu GPIO) k řádku, kde je zapojena pouze jedna noha rezistoru a žádné nohy diody LED.



Upozornění: Připojujete-li vodiče k portu GPIO počítače Raspberry Pi, vždy dbejte mimořádné opatrnosti. Jak jsme již uvedli v předchozí části této kapitoly, připojením chybných pinů můžete počítač Pi vážně poškodit.



Obrázek 12.3: Zkušební model obvodu pro jednoduchý výstup diody LED

V této fázi se nic nestane. To je naprosto v pořádku: piny portu GPIO počítače Raspberry Pi jsou ve výchozím nastavení vypnuté. Chcete-li svůj obvod okamžitě vyzkoušet, přemístěte vodič z pinu 11 na pin 1, aby se dioda LED rozsvítila. Dávejte však pozor, abyste vodič nepřipojili k pinu 2: rezistor k omezení napětí, který se hodí pro zdroj napájení 3,3 V, nebude stačit k ochraně diody LED, když jej připojíte k napětí 5 V. Než budete pokračovat, nezapomeňte kabel přenést zpět na pin 11.

Aby dioda LED sloužila k něčemu užitečnému, začněte nový projekt jazyka Python. Stejně jako u projektů v kapitole 11, „Úvod do jazyka Python“, můžete při tomto projektu použít editor neformátovaného textu nebo software IDLE, který je součástí doporučené distribuce Debian.

Než budete moci pracovat s knihovnou GPIO, kterou jste nainstalovali v předchozí části této kapitoly, musíte knihovnu importovat do svého projektu Python. Začněte tedy soubor odpovídajícím řádkem:

```
import RPi.GPIO as GPIO
```

Pamatujte, že jazyk Python rozlišuje malá a velká písmena, takže řetězec `RPi.GPIO` zadejte přesně tak, jak je zde uveden. Chcete-li, aby jazyk Python podporoval i koncepci času (jinými slovy aby mohla dioda LED blikat, místo abyste ji dokázali jen zapínat a vypínat), potřebujete také importovat časový modul. Přidejte do svého projektu následující řádek:

```
import time
```

Po importu knihoven je čas na adresování portů GPIO. Knihovna GPIO umožňuje snadno adresovat univerzální porty pomocí instrukcí `GPIO.output` a `GPIO.input`, ale než je budete moci použít, musíte piny inicializovat buď jako vstupy, nebo výstupy. V tomto příkladu slouží pin 11 jako výstup, takže do svého projektu přidejte tento řádek:

```
GPIO.setup(11, GPIO.OUT)
```

Sdělíte tím knihovně GPIO, že pin 11 portu GPIO počítače Raspberry Pi by měl být nastaven jako výstup. Kdybyste řídili další zařízení, mohli byste do svého projektu doplnit další řádky `GPIO.setup`. Prozatím však postačí jen jeden.

Když je pin nakonfigurován jako výstup, můžete jeho zdroj napájení 3,3 V zapínat a vypínat v rámci jednoduché ukázky binární logiky. Instrukcí `GPIO.output(11, True)` pin zapnete, zatímco instrukce `GPIO.output(11, False)` jej znovu vypne. Pin si pamatuje svůj poslední stav, takže stačí zadat příkaz na zapnutí pinu a poté ukončit program jazyka Python a pin zůstane zapnutý, dokud jej nenastavíte jinak.

Pokud chcete pin zapnout, mohli byste sice do svého projektu Python zadat pouze `GPIO.output(11, True)`, ale zajímavější bude, když se dioda rozbliká. Nejdříve přidejte následující řádek, abyste ve svém programu vytvořili nekonečný cyklus:

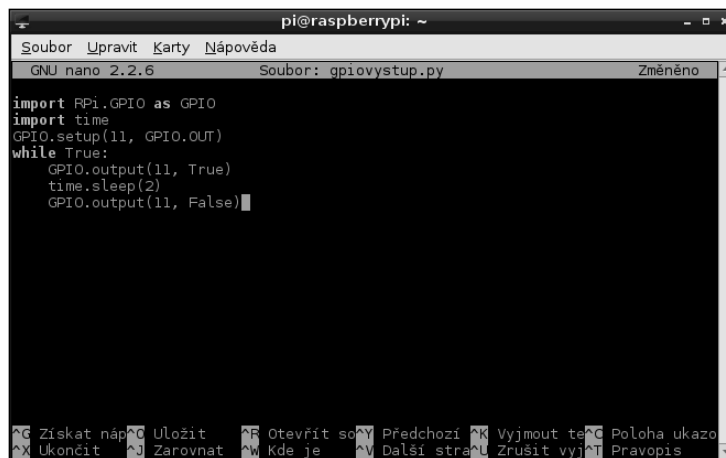
```
while True:
```

Poté přidejte následující řádky, které pin zapnou a nastaví čekání po dobu dvou sekund, načež pin znovu vypnou a opět nastaví čekání po dobu dalších dvou sekund. Zkontrolujte, zda každý řádek začíná čtyřmi mezerami. Tím zajistíte, že bude částí nekonečného cyklu `while`:

```
GPIO.output(11, True)
time.sleep(2)
GPIO.output(11, False)
time.sleep(2)
```

Dokončený program by měl vypadat takto (viz obrázek 12.4):

```
import RPi.GPIO as GPIO
import time
GPIO.setup(11, GPIO.OUT)
while True:
    GPIO.output(11, True)
    time.sleep(2)
    GPIO.output(11, False)
    time.sleep(2)
```



Obrázek 12.4: Program gpiovystup.py s chybějícím posledním řádkem při úpravách v programu nano

Uložte soubor pod názvem `gpiovystup.py`. Používáte-li vývojové prostředí jazyka Python, jako je SPE, nepokoušejte se program spustit v rámci editoru. Většina linuxových distribucí pro počítač Raspberry Pi omezuje použití portu GPIO na uživatele *root*. Program je proto nutné spustit příkazem `sudo python gpiovystup.py` v okně terminálu. Pokud nedošlo k žádné chybě, měla by dioda LED začít v pravidelných intervalech blikat – vytvořili jste tedy první domácí výstupní zařízení pro počítač Pi.

Jestliže něco nefunguje, neplašte se. Nejdříve zkontrolujte všechny spoje. Otvary ve zkušebním modelu obvodu jsou poměrně malé a snadno se může stát, že se zmýlíte a připojíte komponentu do vedlejšího řádku. Poté ověřte, zda jste obvod připojili ke správným pinům portu GPIO. Vzhledem k tomu, že samotný počítač Pi piny nijak neoznačuje, se chyb bohužel můžete dopustit velmi lehce. Nakonec znovu zkontrolujte své komponenty. Pokud je propustné napětí diody LED vyšší než 3,3 V nebo je aktuální rezistor omezující napětí příliš velký, dioda se nerozsvítí.

Tento příklad je sice jednoduchý, ale dobře osvětluje některé základní principy. Chcete-li vyzkoušet další možnosti, můžete diodu LED nahradit bzučákem, který bude vydávat slyšitelný zvuk, nebo servomechanismem či motorem v rámci robotické platformy. Kód, který aktivuje či deaktivuje pin portu GPIO, je možné integrovat do jiných programů, takže se dioda LED může rozsvítit při doručení nové pošty nebo lze třeba signalizovat, že se někdo připojil ke kanálu IRC.

Vstup portu GPIO: Detekce tlačítka

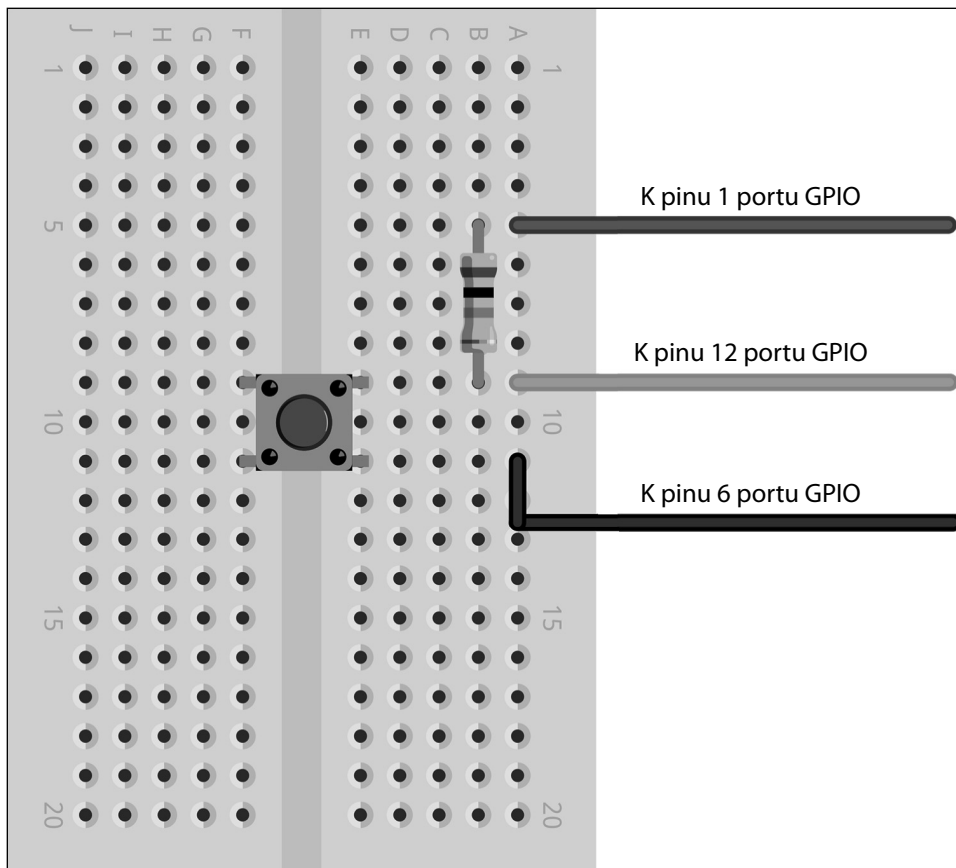
Výstup pomocí portu GPIO je rozhodně užitečný, ale ještě větší možnosti získáte, když výstup dokážete zkombinovat s jedním nebo více vstupy. V následujícím příkladu se naučíte, jak připojit tlačítkový spínač k jinému pinu portu GPIO a přečíst jeho stav pomocí programu jazyka Python.

Stejně jako u předchozího příkladu s výstupem pomocí diody LED také tento příklad využívá knihovnu GPIO jazyka Python. Za předpokladu, že máte tuto knihovnu nainstalovanu, můžete začít se zapojováním obvodu. (Jestliže jste knihovnu GPIO jazyka Python zatím nenainstalovali, přejděte o několik stránek zpátky a postupujte podle uvedených pokynů.)

V případě, že jste již vyzkoušeli příklad s výstupem portu GPIO, můžete obvod buď od počítače Pi odpojit, nebo jej nechat připojený. Tento příklad pracuje s jinými piny, takže oba obvody mohou snadno koexistovat. Ponecháte-li obvod z předchozího příkladu připojený, nezapomeňte nové komponenty připojit k jiné řadě zkušebního modelu obvodu. Jinak součástky nebudou fungovat tak, jak očekáváte.

Zapojte obvod takto:

1. Vložte tlačítkový spínač do zkušebního modelu obvodu. Většina spínačů má dvě nebo čtyři nohy. Z hlediska obvodu stačí připojit dvě nohy. Má-li tlačítko čtyři nohy, patří do dvou dvojic: informaci o tom, které nohy jsou spárovány, naleznete v dokumentaci tlačítka.
2. Připojte rezistor s hodnotou 10 kΩ ke stejnému řádku, kam jste vložili jeden pár nohou tlačítka, a k zatím nepoužitému řádku. Jedná se o pull-up rezistor, který počítači Pi poskytne referenční napětí, aby dokázal zjistit, kdy bylo tlačítko stisknuto.
3. Připojte nepoužitou nohu pull-up rezistoru k pinu 1 portu GPIO počítače Raspberry Pi. Dostanete tak referenční napětí 3,3 V.
4. Připojte nepoužitou nohu tlačítkového spínače k pinu 6 portu GPIO počítače Raspberry Pi. Tento kontakt slouží k uzemnění.
5. Nakonec připojte pin 12 portu GPIO počítače Raspberry Pi k druhé noze tlačítkového spínače na stejném řádku jako rezistor s rezistancí 10 kΩ. Zkušební model obvodu by měl nyní vypadat podobně jako na obrázku 12.5.



Obrázek 12.5: Příklad rozložení zkušebního modelu obvodu pro jednoduchý vstup z tlačítka

Právě vytvořený obvod zajišťuje, že vstupní pin (v tomto případě se jedná o pin 12 portu GPIO počítače Raspberry Pi) je díky pull-up rezistoru připojenému ke zdroji napájení 3,3 V neustále ve *vysokém* stavu. Při stisknutí tlačítka je obvod uzemněn a přejde do *nízkého* stavu, takže program jazyka Python může zjistit, že bylo tlačítko aktivováno.

Možná vás napadlo, proč je rezistor vůbec potřeba a proč spínač jednoduše nepropojí pin 12 s pinem 6 nebo 1 přímo. To je sice možné, ale vzniká přitom takzvaný *plovoucí* (floating) pin. Takový pin neposkytuje informaci o tom, zda se nachází ve vysokém či nízkém stavu. Obvod by proto fungoval tak, jako by bylo tlačítko neustále stisknuté, a nemohl by detekovat skutečné stisknutí tlačítka.

Otevřete nový soubor jazyka Python – buď v textovém editoru nebo v jednom z integrovaných vývojových prostředí (IDE) jazyka Python, které jsou v počítači Raspberry Pi k dispozici. Chcete-li začít, musíte importovat stejnou knihovnu GPIO jako v předchozím příkladu výstupu pomocí tohoto portu:

```
import RPi.GPIO as GPIO
```

Knihovnu `time` importovat nemusíte, protože v tomto příkladu se žádné časovací instrukce nepoužívají. Můžete tedy přejít přímo na nastavení vstupního pinu 12. Přitom postupujte stejně jako v případě nastavení výstupního pinu. Změní se pouze koncová část instrukce:

```
GPIO.setup(12, GPIO.IN)
```

Pokud k tomuto účelu nepoužíváte pin 12, nezapomeňte v předchozím příkazu změnit číslo pinu.

Stejně jako u předchozí ukázky spočívá další krok ve vytvoření nekonečné smyčky, která neustále kontroluje, zda vstupní pin přešel do nízkého stavu (jinými slovy, zda bylo tlačítko stisknuto). Začněte cyklus následujícím řádkem kódu:

```
while True:
```

Čtení stavu vstupního pinu je velmi podobné nastavení stavu výstupního pinu – s jednou výjimkou: než můžete výslednou hodnotu jakkoli zpracovat, musíte ji uložit do proměnné. Následující instrukce jazyka Python vytvoří novou proměnnou s názvem `vstupni_hodnota` (viz kapitola 11, „Úvod do jazyka Python“) a nastaví ji na aktuální hodnotu pinu 12:

```
    vstupni_hodnota = GPIO.input(12)
```

Program by sice bylo možné ihned úspěšně spustit, ale neplnil by svou funkci. Abyste věděli, co se děje, přidejte následující instrukci `print`, která poskytne zpětnou vazbu:

```
    if vstupni_hodnota == False:
        print("Tlačítko bylo stisknuto.")
        while vstupni_hodnota == False:
            vstupni_hodnota = GPIO.input(12)
```

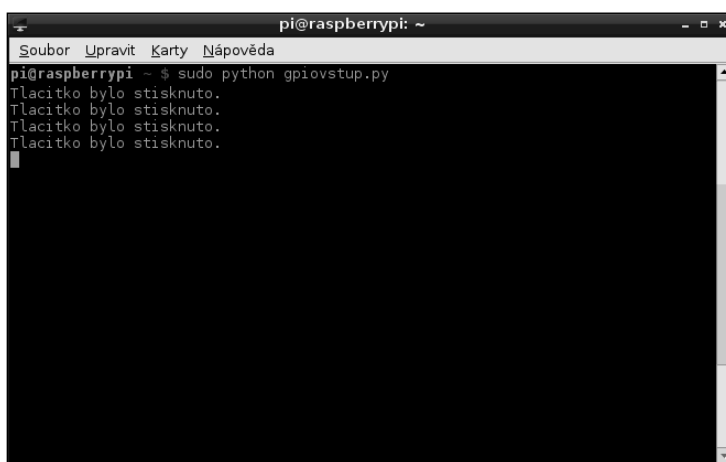
Poslední dva řádky – druhý příkaz `while` a druhé přiřazení proměnné `vstupni_hodnota`, které tvoří tzv. *vnořený cyklus* – jsou velmi důležité. Dokonce i na procesoru počítače Raspberry Pi, který je oproti výkonným procesorům stolních počítačů a notebooků poměrně pomalý, funguje interpret jazyka Python velmi rychle. Tento vnořený cyklus zajistí, že jazyk Python bude neustále kontrolovat stav pinu 12, dokud nepřejde z nízkého do vysokého stavu, což znamená, že tlačítko bylo uvolněno. Bez tohoto cyklu by program opakovaně procházel cyklus, zatímco by tlačítko zůstávalo stisknuté. I kdybyste měli rychlé reflexy a tlačítko okamžitě uvolnili, zpráva by se na obrazovku vypsala mnohokrát, což by bylo matoucí.

Hotový program by měl vypadat takto:

```
import RPi.GPIO as GPIO
GPIO.setup(12, GPIO.IN)
while True:
    vstupni_hodnota = GPIO.input(12)
    if vstupni_hodnota == False:
        print("Tlačítko bylo stisknuto.")
        while vstupni_hodnota == False:
            vstupni_hodnota = GPIO.input(12)
```

Uložte soubor pod názvem `gpiovstup.py` a poté jej v okně terminálu spusťte příkazem `sudo python gpiovstup.py`. Bezprostředně po spuštění programu se nic nestane, ale když stisknete tlačítkový spínač, program vypíše na terminál zprávu z šestého řádku (viz obrázek 12.6). Když tlačítko uvolníte a znovu jej stisknete, zpráva se objeví znovu.

Stejně jako u předchozího příkladu se vstupem jednoduchost tohoto programu klame, protože jej lze využít k mnoha účelům. Kromě toho, že lze zjišťovat stisknutí spínače, můžete pomocí stejného kódu detekovat, zda do vysokého či nízkého stavu nepřešly piny samostatného zařízení, jako je například senzor nebo externí mikrokontrolér.



```
pi@raspberrypi: ~  
Soubor Upravit Karty Nápověda  
pi@raspberrypi ~ $ sudo python gpiovstup.py  
Tlacitko bylo stisknuto.  
Tlacitko bylo stisknuto.  
Tlacitko bylo stisknuto.  
Tlacitko bylo stisknuto.  
█
```

Obrázek 12.6: Výstup programu `gpiovstup.py`

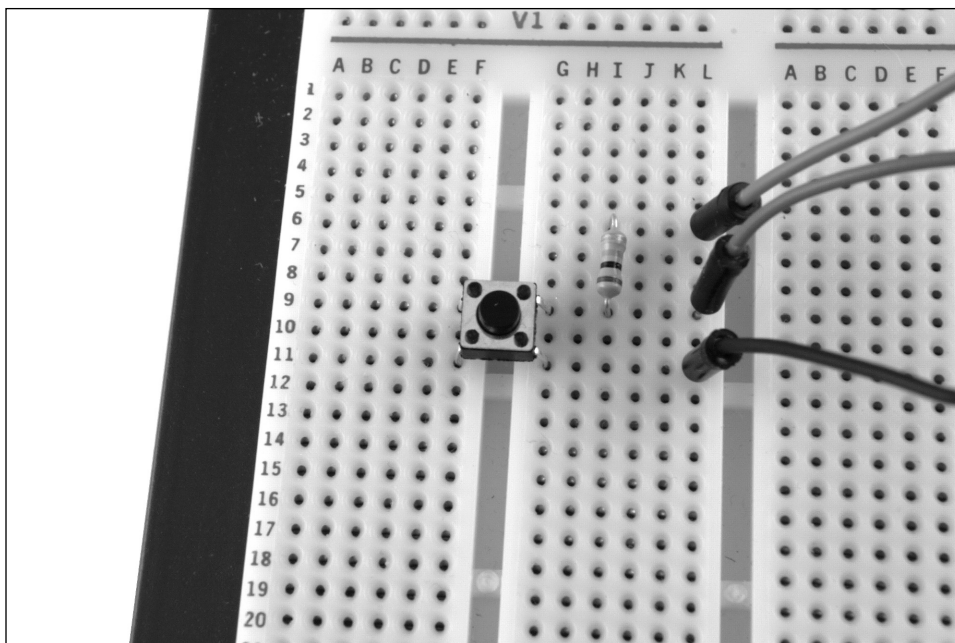
Když kód rozšíříte tak, aby kontroloval stav více tlačítek, z nichž každé je připojeno k samostatnému pinu portu GPIO, můžete dokonce vytvořit jednoduchý herní ovladač se čtyřmi tlačítky. Předchozí kód byste například mohli zkombinovat s hrou *Had s malinou* z kapitoly 11, „Úvod do jazyka Python“, a změnit tak počítač Raspberry Pi na jednoduchou herní konzolu. Do jediného programu lze také integrovat příklady na vstup i výstup, takže by program čekal na stisknutí tlačítka a poté by rozsvítil diodu LED tak, že by nastavil výstupní pin do vysokého stavu. Chcete-li si ověřit, že principům této části rozumíte, pokuste se tento kombinovaný program napsat. Když nebudete vědět, jak dál, nebo si budete chtít své řešení ověřit, naleznete ukázkové řešení v Příloze A, „Recepty jazyka Python“.

Postup od zkušebnímu modelu obvodu

Jak jste zjistili v předchozích příkladech této kapitoly, zkušební modely obvodu jsou ideální, chcete-li rychle vytvářet prototypy obvodů a experimentovat s nimi. Rychle se s nimi pracuje, lze je používat opakovaně a nepoškozuji komponenty.

Tyto zkušební modely obvodu však mají také své nevýhody. Jsou rozměrné, drahé a náchylné k uvolnění kontaktů. To může vést k tomu, že kritické komponenty vypadnou, zejména při

přenášení zkušebního modelu obvodu z místa na místo. Dokonale to dokládá obrázek 12.7: navzdory nejlepší snaze je komponenta tlačítka na zkušebním modelu obvodu upevněna jen volně, a při neopatrné manipulaci s obvodem může proto snadno vypadnout.



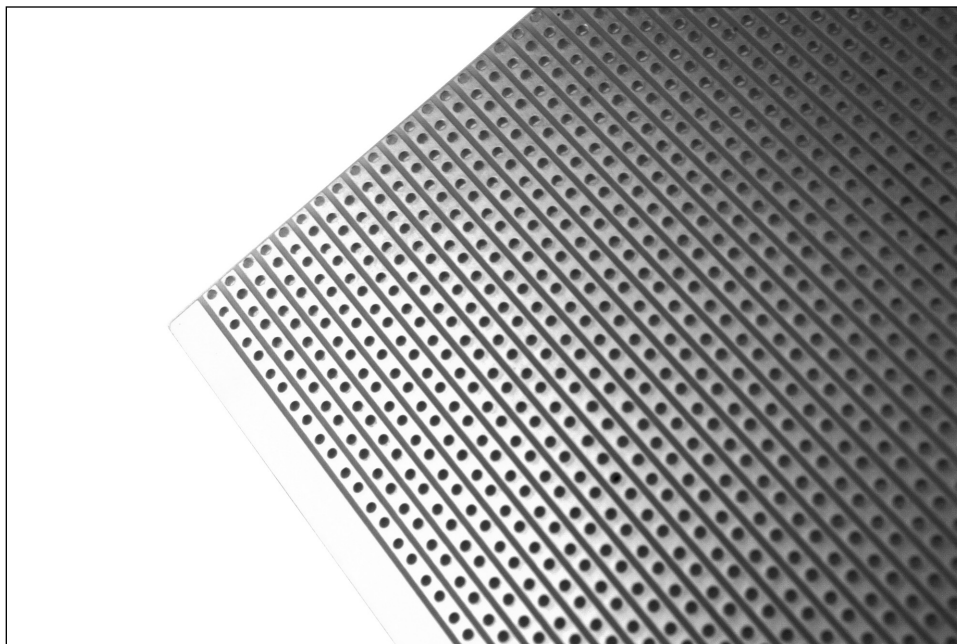
Obrázek 12.7: Ukázkový zkušební model obvodu s volně připojenou komponentou

Mimo jiné se jedná o důvod, proč samotný počítač Raspberry Pi není vytvořen na zkušebním modelu obvodu, ale na *desce s tištěnými spoji* (PCB – printed circuit board), ačkoli v počáteční fázi tvorby prototypů tohoto zařízení se metoda se zkušebním modelem obvodu určitě uplatnila. Desky s tištěnými spoji můžete tisknout a leptat i doma, ale můžete zvolit i jednodušší kompromisní cestu: vytvářet trvalé samostatné obvody pomocí desek *stripboard*.

Stripboard na první pohled připomíná zkušební model obvodu, protože jeho povrch pokrývají malé otvory s roztečí 2,54 mm. Na rozdíl od zkušebního modelu obvodu však neposkytuje žádný chytrý mechanismus, který by zajišťoval, že elektronické komponenty vložené do těchto otvorů zůstanou na svých místech. Místo toho musíte tyto komponenty na stripboard připájet. Stripboard se často označuje obchodním názvem *Veroboard*, který je ochrannou známkou společnosti Vero Technologies v Británii a společnosti Pixel Print v Kanadě.

Obvod typu stripboard nabízí oproti zkušebnímu modelu obvodu mnoho výhod. Deska typu stripboard je značně levnější než zkušební model obvodu podobné velikosti a lze ji rozdělit na části odpovídající menším obvodům. Zároveň je možné na jediném velkém kusu desky stripboard vytvořit několik menších, ale nezávislých obvodů.

Vzhledem k tomu, že komponenty jsou na desku typu stripboard připájeny, jsou tyto obvody také značně robustnější než prototypy vytvořené pomocí zkušebního modelu obvodu. Obvod na desce stripboard můžete přenášet z místa na místo a přitom se nemusíte obávat, že některá z jeho komponent odpadne a ztratí se. Na obrázku 12.8 je znázorněna spodní plocha desky stripboard, kde jsou patrné měděné stopy.



Obrázek 12.8: Měděné stopy na spodní straně kusu desky stripboard

S deskami stripboard se velmi snadno pracuje, takže představují ideální přestupní stanici k návrhu a výrobě vlastních desek s tištěnými spoji. Než si však desku stripboard zakoupíte, měli byste si přečíst následující informace:

- Existují různé typy desek stripboard. Některé z nich mají na spodní straně měděné stopy, které sledují celou délku řádku nebo výšku sloupce, zatímco jiné desky jsou rozděleny na dva samostatné řádky, které mají uprostřed mezeru jako zkušební model obvodu. Ještě jiný typ desky stripboard, který se často označuje jako projektová deska, žádné měděné stopy nemá a kvůli elektrickému propojení komponent je nutné použít vodiče.
- Desky stripboard lze vyrábět v různé tloušťce a z různých materiálů. Určitý typ se může pro konkrétní projekt hodit více než jiné typy. Například deska stripboard s tepelnou ochranou je vhodná pro obvod, který má fungovat v prostředí s vysokými teplotami, zatímco silnější desky stripboard je vhodné použít pro obvody, které mají vydržet nešetné zacházení.
- Kvůli efektivnějšímu rozložení komponent na desce stripboard je možné stopy na spodní straně přerušit a komponenty tak elektricky oddělit. Díky tomu můžete omezit plýtvání prostorem na desce a u složitějších obvodů se bez toho vůbec neobejdete. Aby však

výsledek hezky vypadal, budete potřebovat malý ruční nástroj zvaný *řezačka stop* (track cutter). Jestliže máte v plánu pracovat s deskami stripboard, nezapomeňte tuto řezačku připsat na svůj nákupní seznam, ačkoli v nouzi ke stejnému účelu poslouží i břit vrtáku.

Při manipulaci s deskami stripboard je také dobré znát určitá pravidla. Pokud je nedodržíte, můžete si práci zbytečně zkomplikovat:

- Měděné stopy na spodní straně desky stripboard obvykle nejsou ničím potaženy. Když se stop dotknete, měď ztratí svůj lesk a ztíží se její pájení. Nedotýkejte se spodní strany desky stripboard, pokud ji okamžitě nebudete používat. Je-li příliš pozdě, měděný povrch jemně vyleštíte pomocí drátěnky na nádobí, abyste před pájením odstranili korozí.
- Na rozdíl od desek s tištěnými spoji nemají desky stripboard žádnou *pájecí masku* (solder mask), což je hmota, která zabraňuje tomu, aby se pájka dostala na nežádoucí místa. Z tohoto důvodu je pájení poněkud náročnější než u desek s tištěnými spoji. Velmi snadno se může stát, že stopy náhodou propojíte příliš velkou kapkou pájky. V takovém případě odstraňte přebytečnou pájku pomocí odsávacího knotu a zkuste pájet znovu.
- Díky otvorům lze desky stripboard snadno dělit na části libovolného rozměru, ale na okrajích těchto dílů zůstávají otřepy. Než začnete zapojovat svůj obvod na rozřezanou desku stripboard, věnujte trochu času na zabroušení hran. Přitom si nezapomeňte nasadit ochrannou masku, protože vdechování prachu z desky stripboard není zrovna zdravé.

Stručný průvodce pájením

Nestačí pořídit si páječku, ale musíte také vědět, jak s ní pracovat. Stejně jako kterákoli dovednost také pájení vyžaduje ke svému zvládnutí praxi. Budete-li se řídit tipy v této části a pravidelně cvičit, zanedlouho budete vytvářet čisté a úhledné spájené spoje.



Upozornění: Tato rada může vypadat samozřejmě, ale stojí za to ji zdůraznit: pájky se při práci zahřívají na mimořádně vysokou teplotu. Dávejte pozor, abyste se nedotkli žádného odkrytého kovového povrchu, dokonce i dále od hrotu páječky, a pečlivě sledujte, kam kov odkapává. Máte-li možnost, kupte si páječku se stojanem nebo si poříďte samostatný žáruvzdorný stojan. Horkou páječku nikdy nenechávejte bez dozoru, a pokud vám pájka ukápně, nepokoušejte se ji zachytit!

Principem pájení je roztavení malého kusu kovu, který pak vytvoří spoj mezi dvěma komponentami. Když se na počítač Raspberry Pi podíváte zespoda, uvidíte mnoho příkladů takových spojů. Všechny větší komponenty se spojují pomocí takzvaného *pájení přes otvory* (through-hole soldering), kdy jsou vodiče komponent protaženy otvory v desce s tištěnými spoji a poté připájeny na své místo. Menší komponenty se upevňují pomocí *povrchového pájení* (surface-mount soldering).

Pájka není čistý kov, ale obsahuje také látku zvanou *tavidlo* (flux), která má odleptat případné nečistoty na pájených površích a zajistit tak co nejčistší spoj. Většina elektronických pájek zahrnuje tři až pět *jader* tavidla. Tavidlo lze také koupit samostatně jako pastu nebo v kapalně formě, ačkoli při kutilském pájení to zpravidla není nutné.

Na začátku pájení si připravte čistou a dobře osvětlenou pracovní plochu. Zajistěte také, aby bylo pracovní místo dobře větrané. Výpary pájky nejsou příliš zdravé, a ačkoli je téměř vyloučeno, že by se při malé intenzitě domácího pájení jejich koncentrace dostala na nebezpečnou úroveň, je rozumné expozici minimalizovat.

Kromě toho byste měli nějakým způsobem chránit svou pracovní plochu. Často se stává, že ukápnou malé kousky roztavené pájky, které mohou popálit stůl a vytvořit na něm skvrnu. Můžete si zakoupit antistatickou pracovní podložku (příklad viz obrázek 12.9), ale stejně dobře poslouží i tlustý časopis. Nemyslete si, že vystačíte s několika listy novin – pájka stačí propálit tenký papír ještě dříve, než vychladne.



Obrázek 12.9: Ukázka pracovní plochy pro pájení s ochrannou antistatickou podložkou

Pokud potřebujete pájet drobné součástky, které je nutné sledovat zblízka, měli byste přitom nosit ochranné brýle. Vřící tavidlo uvnitř pájky někdy způsobí, že pájka vyprskne směrem nahoru. Když vás přitom zasáhne do oka, budete chvíli pořádně trpět.

Těmito varováními se však nenechejte od pájení odradit. Pájka je sice mimořádně horká, ale chladne rychle. Popáleniny jsou proto málo časté a navíc prakticky bez následků. Respektujte své pracovní vybavení, ale nebojte se ho.

Když jste si připravili a zabezpečili pracovní plochu, rozložte své vybavení. Páječku byste měli položit na stranu své dominantní ruky a umístit ji tak, aby její kabel nevedl přes pracovní plochu. Před zapojením páječky do sítě zkontrolujte, zda s ní můžete volně pohybovat. Pokud se kabel o něco zachytí, můžete se kvůli tomu popálit.

Namočte svou pájecí houbu. Měla by být vlhká, ale neměla by z ní odkapávat voda. Je to důležité: vlhká houba slouží k čištění páječky, a pokud je suchá, spálí se a může poškodit citlivý pájecí hrot.

Než páječka dosáhne své provozní teploty, několik minut to trvá. Pokud jste zakoupili páječku s řízením teploty, obvykle to poznáte podle kontrolky, která se vypne nebo zapne, nebo podle zobrazeného číselného údaje. (Způsob, kterým páječka indikuje teploty, zjistíte v uživatelské příručce dodané spolu s páječkou.)

Po dosažení provozní teploty je čas připravit páječku postupem, který se označuje jako *pokovení* (tinning). Postupujte takto:

1. Přiložte konec pájky k hrotu páječky a nechejte malé množství pájky, aby se roztavilo na páječce. Dávejte pozor, abyste kov neroztavili příliš: kromě toho, že se jedná o plýtvání pájkou, může navíc přebytečná pájka ukápnout na pracovní plochu.
2. Otřete hrot páječky do houby. Pokud syčí a prská, je houba příliš vlhká. Ponechtejte ji zchladnout, poté ji vyjměte ze stojanu a vyždímejte ji.
3. Pokračujte v otírání hrotu páječky, dokud je pokryta stříbrnou vrstvou pájky (viz obrázek 12.10). V případě potřeby naneste na hrot více pájky.



Obrázek 12.10: Hrot páječky, který je pokovován pájkou

Díky pokovení je hrot pájky chráněn před poškozením a zajišťuje se efektivní přenos tepla na pájené povrchy. Nedostatečné pokovení hrotu pájky představuje jednu z nejčastějších příčin vadných spájených spojů. Pokud pájíte hodně spojů, je někdy potřeba tento postup vícekrát opakovat. Také byste jej měli znovu provést na konci pájení. Obecně platí, že je vhodné proces pokovení opakovat, když hrot páječky ztratí svůj lesklý povlak.

Když je páječka připravena, můžete začít pájet. Umístěte pájené položky – například desku s tištěnými spoji a nohy komponenty – do pracovního stojanu a zkontrolujte, zda vám nic nebrání ve výhledu. Vytáhněte část pájky ze schránky nebo z cívky a začněte komponenty pájet takto:

1. Pokud připevňujete komponenty zasunované do otvorů k desce s tištěnými spoji, desce stripboard nebo podobné desce s otvory, protáhněte nohy komponenty přes otvor a vyhněte je ven, aby součástka po překlopení desky nevypadla.
2. Když desku upevníte do pracovního stojanu, přiložte hrot páječky proti komponentě i měděnému kontaktu na desce. Záleží na tom, aby se páječka dotkla obou předmětů: pokud je páječka v kontaktu pouze s jedním z nich, výsledný spoj nebude kvalitní.
3. Zahřátí oblasti trvá jen několik sekund. Napočítejte do tří a poté zatlačte páječkou na komponentu a měděný kontakt (viz obrázek 12.11). Jestliže se pájka neroztaví, odtáhněte ji, počítejte o několik sekund déle a zkuste to znovu. Pokud nebude tát ani pak, zkuste změnit polohu páječky.



Obrázek 12.11: Pájení komponenty zasunované do otvoru na desku s tištěnými spoji

4. Když pájka začne téci, můžete si všimnout, že směřuje do otvoru v desce. To svědčí o tom, že oblast je dostatečně horká, aby vznikl dobrý spájený spoj. Pokud pájka pouze změkne, znamená to, že oblast ještě není dost zahřátá.
5. Odsuňte od spoje nejdříve pájku a poté páječku. (Pokud nejdříve odtáhnete páječku, pájka ztvdne a ke kontaktu zůstane připevněna celá cívka pájky!)

Jestliže vše proběhlo správně, dostanete pevný spájený spoj, který vydrží mnoho let. Pokud nikoli, nenechtejte se odradit. Stačí, když přitisknete páječku na spoj, až pájka začne znovu té-

ci, a rozlitý přebytek pájky případně očistíte pomocí odsávacího knotu. Ideální spájený spoj by měl svým tvarem poněkud připomínat sopku, která se zvedá z povrchu desky a obklopuje nohu komponenty.

Páječku nikdy nenechávejte v kontaktu se součástkami déle než několik sekund. To je důležité zejména při pájení komponent citlivých na teplo, jako jsou integrované obvody, které se mohou při delším styku s horkou pájkou poškodit. Používáte-li pájecí stanici s řízením teploty, ověřte, zda je nastavena vhodná teplota podle použitého druhu pájky (podrobnosti naleznete na balení pájky nebo v dokumentaci).

Po dokončení pájení nezapomeňte znovu pokovit hrot páječky. Když to neuděláte, může hrot při skladování zkorodovat a bude nutné jej vyměnit mnohem dříve, než by jinak bylo potřeba.



Tip: Při pájení nezapomeňte páječkou zahřát oba povrchy. Zahřejete-li pouze jeden povrch, dostanete takzvaný *suchý* neboli *studený spoj* (dry/cold joint), kde pájka na jeden z povrchů správně nepřiléhá. Tyto spoje se za nějaký čas rozpadnou a musíte je spájet znovu.

Stejně jako kterákoli jiná dovednost se ani pájení neobejde bez praxe. Mnoho obchodů s elektronikou prodává sady, které obsahují desku s tištěnými spoji a vybrané komponenty, pomocí nichž můžete cvičit pájení komponent zasunovaných do otvorů. Některé rozšiřující desky pro počítač Raspberry Pi se také dodávají ve stavebnicové formě a vyžadují pájení. Patří k nim Slice of Pi od společnosti Ciseco. Další informace o těchto deskách naleznete v kapitole 13, „Rozšiřující desky“.

Rozšiřující desky

S řadami pinů portu GPIO počítače Raspberry Pi sice můžete pracovat přímo, jak jste se dozvěděli v kapitole 12, „Hackování hardwaru“, ale mnohem praktičtější je pracovat se specializovanou prototypovou deskou. Prototypové desky jsou rozšiřující desky navržené tak, aby je bylo možné připojit mezi uživatelský projekt a počítač Raspberry Pi. Nabídka sahá od poměrně jednoduchých – jako je Slice of Pi společnosti Ciseco – až po složitou a výkonnou desku Gertboard.

Na nejzákladnější úrovni rozšiřující desky prostě usnadňují přístup k pinům portu GPIO počítače Raspberry Pi. To znamená, že je lze snáze připojit ke zkušebnímu modelu obvodu, případně mají větší rozteče a popisy, které zjednodušují připojení jiných zařízení. Některé desky jsou vybaveny obvody na připojení konkrétních přídatných zařízení, jako je malý bezdrátový vysílač XBee. Další nabízejí malý povrch, na který můžete připájet vlastní komponenty a vytvořit tak uživatelskou desku.

Trh hardwaru se neustále rychle mění a nadace Raspberry Pi Foundation stále motivuje vývojáře, aby vytvářeli další rozšiřující zařízení. Není možné sestavit kompletní seznam všech rozšiřujících desek, které jsou k dispozici nebo se jejich vydání v dohledné době plánuje. V této kapitole však představíme tři nejčastěji používané desky a ukážeme, jak je lze použít.

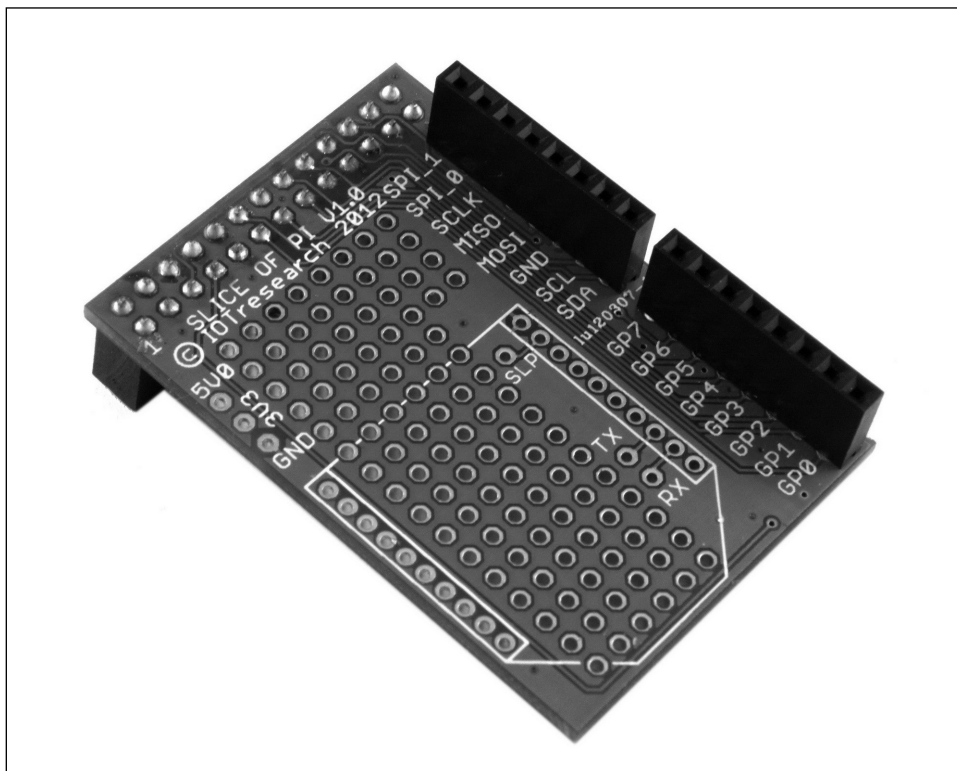
Slice of Pi společnosti Ciseco

Nejjednodušší, ale přesto užitečná je deska Slice of Pi společnosti Ciseco (viz obrázek 13.1). Deska Slice of Pi dodávaná ve formě stavebnice poskytuje malou oblast na prototypování, prostor pro bezdrátový modul XBee a plný přístup k pinům portu GPIO počítače Raspberry Pi.

Kompaktní deska Slice of Pi je navržena tak, aby ji bylo možné přímo připojit k řadám pinů portu GPIO v horní části počítače Raspberry Pi. Zbytek desky zakrývá část povrchu počítače Pi, ale nepřesahuje okraje desky ani neblokuje žádné běžně používané porty. Slice of Pi je však umístěna nad konektorem výstupu videa DSI. Když se tento konektor používá, lze plochý kabel bez problémů protáhnout pod deskou Slice of Pi.

Hlavní výhodou levné desky Slice of Pi místo přímého připojení k řadám pinů portu GPIO počítače Pi představuje značení na desce s plošnými spoji, které port GPIO postrádá, a použití samičích řad pinů. Díky tomu lze pracovat s *propojovacími vodiči* (jumper lead) typu samec-samec, nebo dokonce propojit desku se zkušebním modelem obvodu pomocí kusů drátu (jak

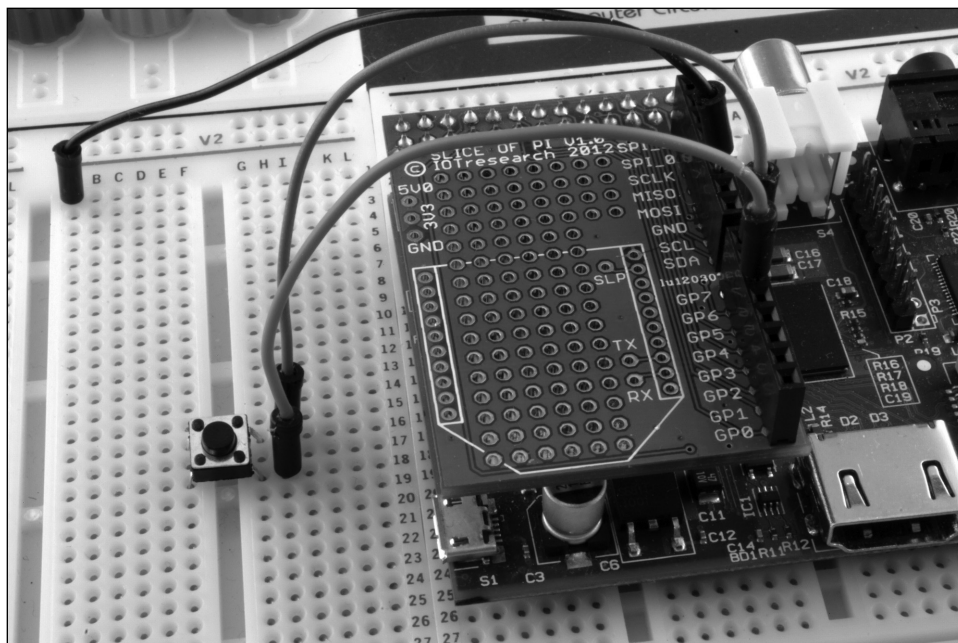
je patrné na obrázku 13.2) nebo s jinou deskou s tištěnými spoji. Případně lze komponenty přímo připojit k řadám pinů. Samčí řady pinů počítače Pi oproti tomu vyžadují použití specializovaných propojovacích vodičů typu samec-samice.



Obrázek 13.1: Slice of Pi společnosti Ciseco

Deska Slice of Pi také nabízí základní úroveň ochrany počítače Pi. Vzhledem k samičím řadám pinů je méně pravděpodobné, že náhodně zkratujete dva piny. Chybám při zapojení také pomáhají předcházet rozteče řad pinů – s osmi univerzálními piny přivedenými do jedné řady a dalšími specializovanými piny v jiné řadě. Důležitější je, že deska Slice of Pi neposkytuje přímý přístup k žádnému z pinů označených jako Do Not Connect (Neřipojovat – viz kapitola 12, „Hackování hardwaru“), takže se snižuje riziko poškození počítače Pi kvůli chybnému připojení.

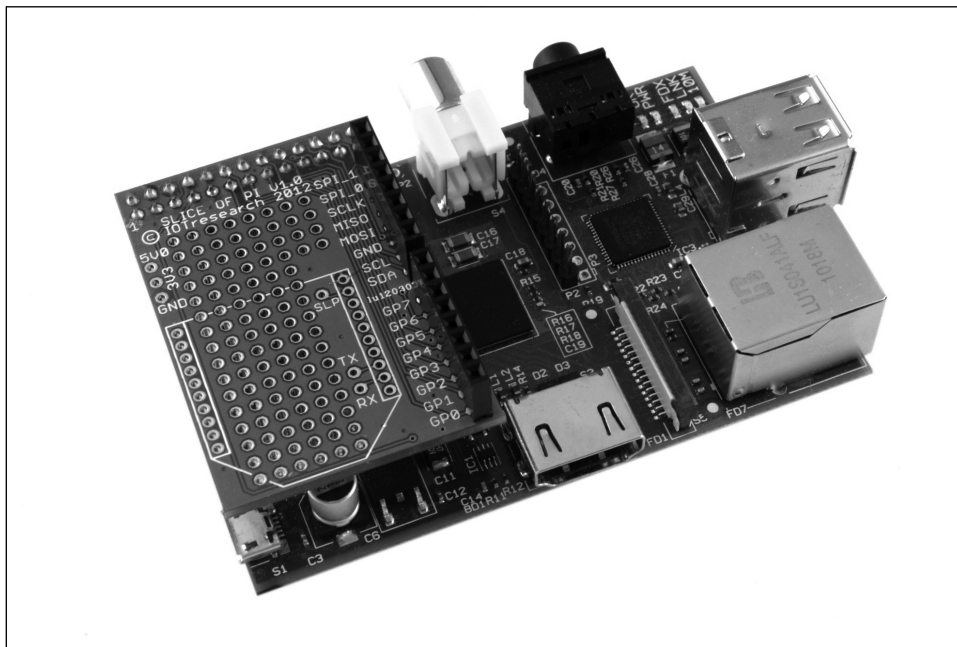
Připojení k bezdrátovému rozhraní XBee se může hodit v případech, že uvažujete o nasazení počítače Pi v projektu integrovaného počítače. V kombinaci s kompatibilním vysílačem, jako je modul XBee, RF-BEE či RN-XV, lze zajistit bezdrátový přístup k sériovému portu UART počítače Pi. Pomocí tohoto přístupu lze například z počítače Pi číst hodnoty ze senzoru, převzít řízení softwaru počítače Pi, nebo dokonce ze vzdáleného umístění ovládat robota kontrolovaného počítačem Pi.



Obrázek 13.2: Připojení desky Slice of Pi ke zkušebnímu modelu obvodu

Oblast prototypování nakonec v případě malých návrhů obvodů poskytuje alternativu ke zkušebním modelům obvodů nebo deskám stripboard. Jestliže uvažujete o vývoji malého rozšiřujícího modulu pro počítač Raspberry Pi, například desky senzorů na měření teplot nebo detekci vlhkosti, můžete potřebné komponenty připájet přímo na desku Slice of Pi. Díky nízké ceně tato deska nabízí snadný způsob, jak vytvářet zákaznické rozšiřující moduly pro počítač Pi, které jsou zcela samostatné a nevyžadují žádnou kabeláž. Stačí připojit desku Slice of Pi k portu GPIO počítače Raspberry Pi a zařízení je připraveno k použití (viz obrázek 13.3).

Nevýhodou desky Slice of Pi je její jednoduchost. Vzhledem ke své kompaktní velikosti se nehodí k zapojování větších nebo složitějších prototypů, a ačkoli rozložení řad pinů poskytuje určitou ochranu před zkratováním, neexistuje žádná ochrana před použitím komponent s napájením 5 V, které mohou poškodit logické obvody počítače Pi pod napětím 3,3 V. Také značení se poněkud liší od obvyklého popisu pinů portu GPIO počítače Raspberry Pi, což může vést k nejasnostem, není-li použita knihovna GPIO napsána z ohledem na desku Slice of Pi. Tabulka 13.1 poskytuje převod mezi oběma systémy značení.



Obrázek 13.3: Deska Slice of Pi připojená k portu GPIO počítače Pi

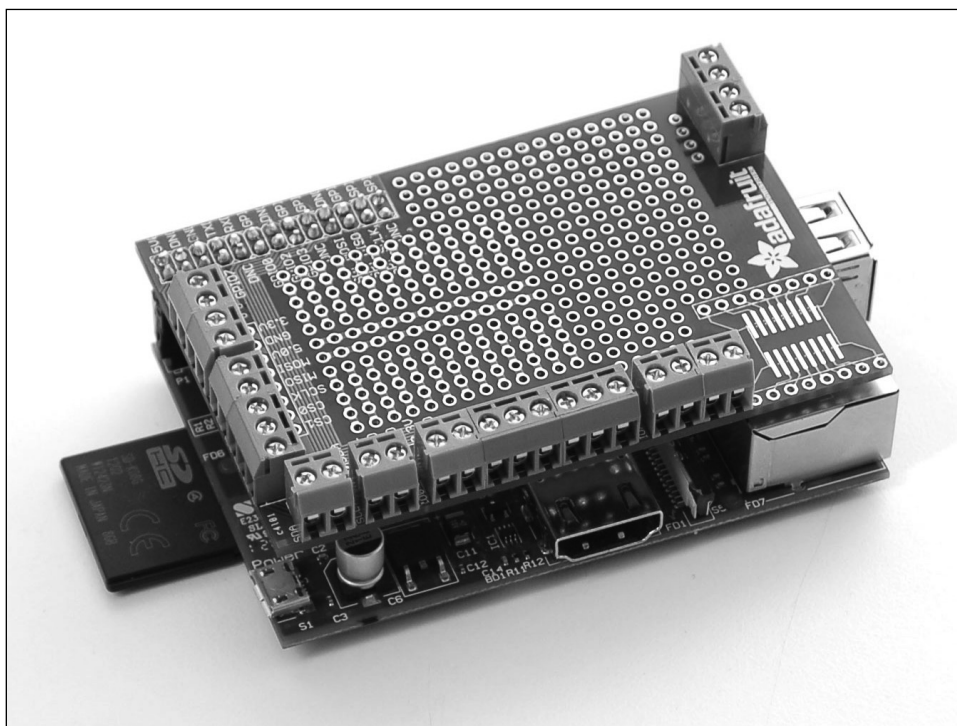
Tabulka 13.1: Převod značení pinů portu GPIO a desky Slice of Pi

Číslo fyzického pinu	Oficiální označení	Popis na desce Slice of Pi
7	General Purpose Clock (Univerzální hodiny – neboli Pin GPIO číslo 4)	GP7
11	Pin GPIO číslo 17	GP0
12	Pin GPIO číslo 18	GP1
13	Pin GPIO číslo 21	GP2
15	Pin GPIO číslo 22	GP3
16	Pin GPIO číslo 23	GP4
18	Pin GPIO číslo 24	GP5
22	Pin GPIO číslo 25	GP6

Desku Slice of Pi si můžete zakoupit ve webovém obchodě společnosti Ciseco na adrese <http://shop.ciseco.co.uk/slice-of-pi/>.

Prototyping Pi Plate společnosti Adafruit

Deska Prototyping Pi Plate společnosti Adafruit je založena na stejném obecném principu jako Slice of Pi společnosti Ciseco. Toto rozšíření je dostupné ve formě stavebnice a poskytuje plochu prototypování spolu se snadným přístupem k pinům portu GPIO počítače Pi (viz obrázek 13.4). Na rozdíl od desky Slice of Pi má však deska Prototyping Pi Plate stejný rozměr jako samotný počítač Pi. Díky tomu nabízí mnohem větší pracovní plochu než Slice of Pi a dovoluje vytvářet mnohem větší obvody. Deska Prototyping Plate je také vybavena *koncovkami šroubů* (screw terminal) s průměrem 3,5 mm pro připojení pinů, díky kterým lze vytvářet pevná dočasná připojení pomocí obyčejných vodičů – ačkoli dostupné jsou i otvory s roztečí 2,54 mm pro samčí nebo samičí řady pinů, pokud jim dáváte přednost.

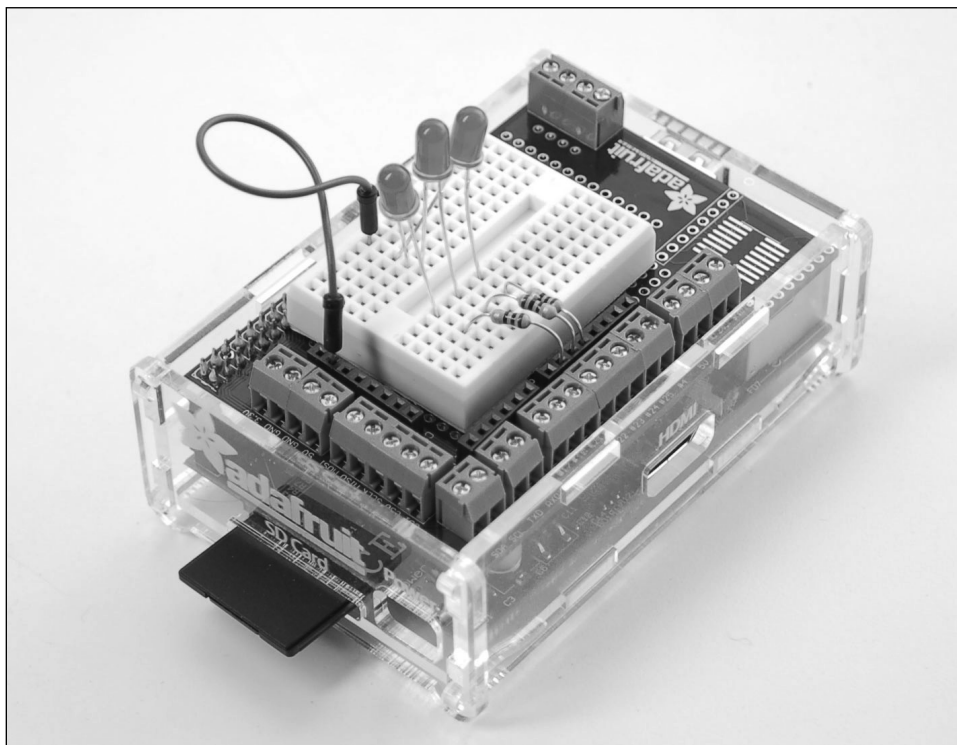


Obrázek 13.4: Prototyping Pi Plate společnosti Adafruit

Návrh a rozložení desky Prototyping Pi Plate bude povědomé každému, kdo někdy pracoval s mikrokontrolérem Arduino. Rozšiřující deska se stejným formátem jako cílové zařízení, která se připojuje k integrovaným řadám pinů a zůstává umístěna nad povrchem původní desky, se běžně vyskytuje ve sféře zařízení Arduino, kde se takovým rozšiřujícím deskám říká *shield*. Koncepce desky Prototyping Pi Plate ve skutečnosti vychází z vlastního návrhu společnosti Adafruit, která vytvořila rozšiřující desku Protoshield pro mikrokontrolér Arduino.

Rozšiřující deska plného formátu Pi Plate umožňuje připojit řadu pinů portu GPIO a pokrývá celý povrch počítače Pi. Vzhledem k jejímu umístění nad počítačem Pi se může zkomplikovat přístup k výstupu videa DSI a vstupním konektorům kamery MIPI CSI-2, ačkoli delší plochý kabel byste měli pod deskou Pi Plate protáhnout bez potíží. Jestliže plánujete použití přídavného kamerového modulu nebo displeje připojeného ke konektoru DSI, zkontrolujte délku plochého kabelu, než začnete připravovat svůj projekt na desce Pi Plate.

Plocha prototypování desky Pi Plate je rozdělena na dvě části, přičemž každá polovina umožňuje zapojování přes otvory se standardní roztečí 2,54 mm. První polovina plochy prototypování má podobný vzhled jako zkušební model obvodu: řádky jsou propojeny měděnými stopami na spodní straně a ústřední sběrnice uprostřed poskytuje společné napájení a zemnění. Druhá polovina plochy prototypování takové stopy neobsahuje, takže dovoluje sestavovat více vlastních obvodů. Dostupný povrch pro tvorbu obvodů je značně větší než u desky Slice of Pi, takže se deska Pi Plate hodí pro složitější projekty. Kromě toho nabízí *povrch SOIC*, který je určen k pájení komponent, které se umísťují na povrchu. To lze využít u součástí, které nejsou k dispozici ve formátu zasunovaném do otvorů.

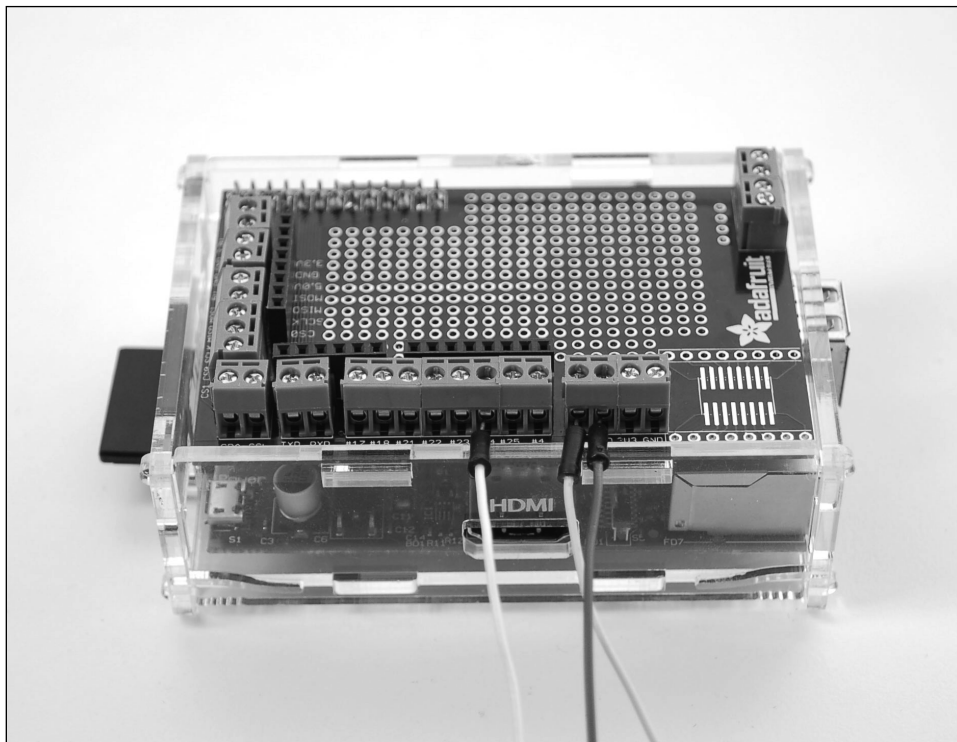


Obrázek 13.5: Deska Pi Plate se zkušebním modelem obvodu čtvrtinové velikosti

Díky jejím značným rozměrům lze také desku Pi Plate použít se zkušebním modelem obvodu čtvrtinové velikosti (v obchodě Adafruit se prodává jako „tiny breadboard“ – malý zkušební

model). Tento malý zkušební model obvodu s dvěma částmi se dodává spolu se samolepicí pěnovou podložkou a můžete jej připevnit na horní část desky Pi Plate nad plochu prototypování (viz obrázek 13.5). V takovém případě nelze na ploše prototypování vytvořit trvalý obvod, ale tato kombinace představuje samostatné rozšíření pro rychlé dočasné prototypování menších obvodů.

Stejně jako deska Slice of Pi je také deska Prototyping Pi Plate v základě jednoduché zařízení. Sada neobsahuje žádné aktivní komponenty, ale pouze řady pinů, koncovky a vlastní desku. Z tohoto důvodu nabízí jen minimální dodatečnou ochranu portu GPIO počítače Pi kromě toho, že ztěžuje náhodné zkratování dvou připojení. Oproti desce Slice of Pi poskytuje deska Pi Plate přístup ke všem 26 pinům v obou řadách portu GPIO počítače Pi. To znamená, že můžete náhodou připojit součástky k pinům, které jsou na diagramu portu GPIO označeny jako Do Not Connect (viz kapitola 12, „Hackování hardwaru“). Stejně jako v případě přímého připojování součástek k těmto pinům to není vhodné a může to způsobit trvalé poškození počítače Pi.



Obrázek 13.6: Deska Pi Plate připojená k počítači Raspberry Pi v krytu

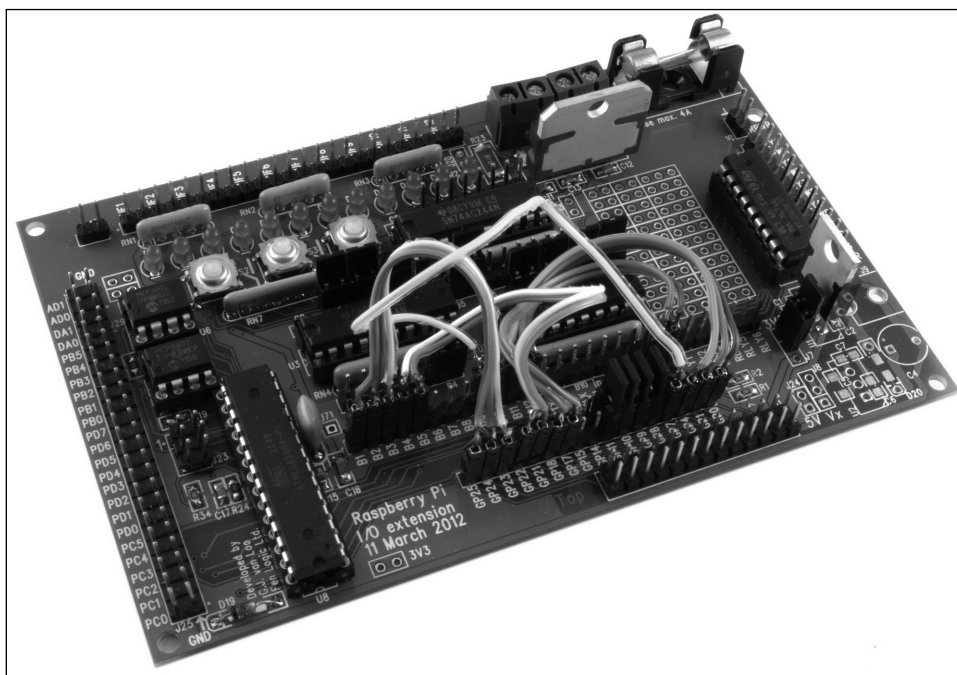
Pravděpodobně hlavním argumentem pro desku Pi Plate oproti desce Slice of Pi je její inteligentní návrh. Po připojení řady pinů portu GPIO k počítači Pi již prakticky není důvod rozšiřující desku odpojovat, pokud nepotřebujete přístup ke konektoru DSI nebo konektoru kamery MIPI CSI-2. Vzhledem k tomu, že rozšiřující deska zvětšuje pouze výšku, ale nikoli

šířku či délku počítače Pi, a vzhledem k tomu, že obsahuje koncovky šroubů pro boční přístup k pinům portu GPIO, je také kompatibilní s překvapivým počtem krytů pro počítač Raspberry Pi (viz obrázek 13.6). Plánujete-li používat počítač Pi v kombinaci s deskou Pi Plate spolu s krytem, zkontrolujte však, zda lze kryt přizpůsobit na výšku nebo zda nabízí přístup k portu GPIO přes své víko.

Sadu Prototyping Pi Plate Kit od společnosti Adafruit si můžete zakoupit na adrese <http://www.adafruit.com/products/801>.

Gertboard společnosti Fen Logic

Deska Gertboard (viz obrázek 13.7), která se správně nazývá Raspberry Pi I/O Extension, získala svůj krátký název podle svého konstruktéra Gerta van Loo. Van Loo je zaměstnancem společnosti Broadcom a členem týmu, který navrhl procesor BCM2835 typu SoC v jádru počítače Raspberry Pi. Desku Gertboard vytvořil proto, aby umožnila přístup k některým funkcím čipu, které obecný návrh počítače Pi skrývá, a poskytla výkonnou a všestrannou platformu pro experimentování s elektronikou.

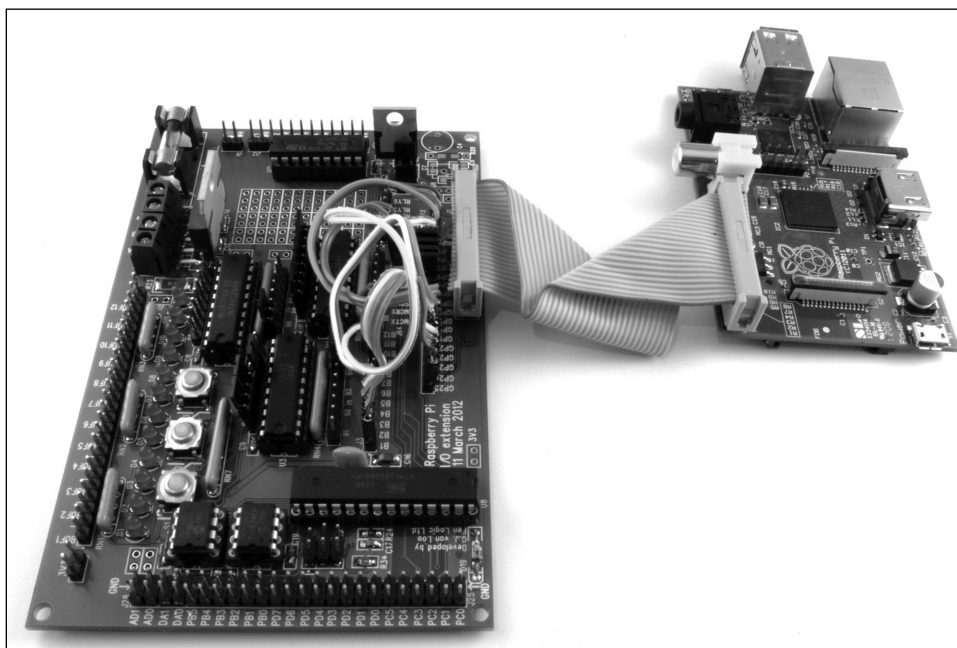


Obrázek 13.7: Deska Gertboard, kterou navrhl Gert van Loo

Na rozdíl od desek Slice of Pi a Prototyping Pi Plate je Gertboard *aktivní* rozšiřující deska s mnoha komponentami, které mají rozšířit funkčnost počítače Pi. Deska Gertboard posky-

tuje 12 vstupně-výstupních portů s vyrovnávací pamětí se stavovými diodami LED, tři tlačítkové spínače, šest budičů otevřeného kolektoru, motorový kontrolér s napětím 48 V a proudem 4 A, dvoukanálový digitálně-analogový převodník (DAC) a dvoukanálový analogově-digitální převodník (ADC). Deska Gertboard kromě toho podporuje mikrokontrolér s 28 piny, jako je Atmel ATmega328 použitý v systému prototypování Arduino, který lze vložit do patice na desce Gertboard a programovat přímo z počítače Raspberry Pi.

Jak je patrné už z přehledu funkcí, je deska Gertboard výkonné a komplexní zařízení. Díky své flexibilitě představuje ideální volbu pro náročnější projekty, ale její složitost se zároveň projevuje i na velikosti. Deska Gertboard je téměř dvaapůlkrát větší než samotný počítač Raspberry Pi, takže se jedná o nejvýkonnější a zároveň největší rozšiřující desku, která je pro počítač Pi v době psaní této knihy k dispozici. Kvůli rozdílným rozměrům se deska Gertboard k počítači Pi připojuje plochým kabelem, který je ukončen samčími koncovkami (viz obrázek 13.8). Díky tomuto kabelu lze desku Gertboard umístit dále od počítače Pi, případně obě zařízení spojit, ale s možností rychlého odpojení v případě potřeby.

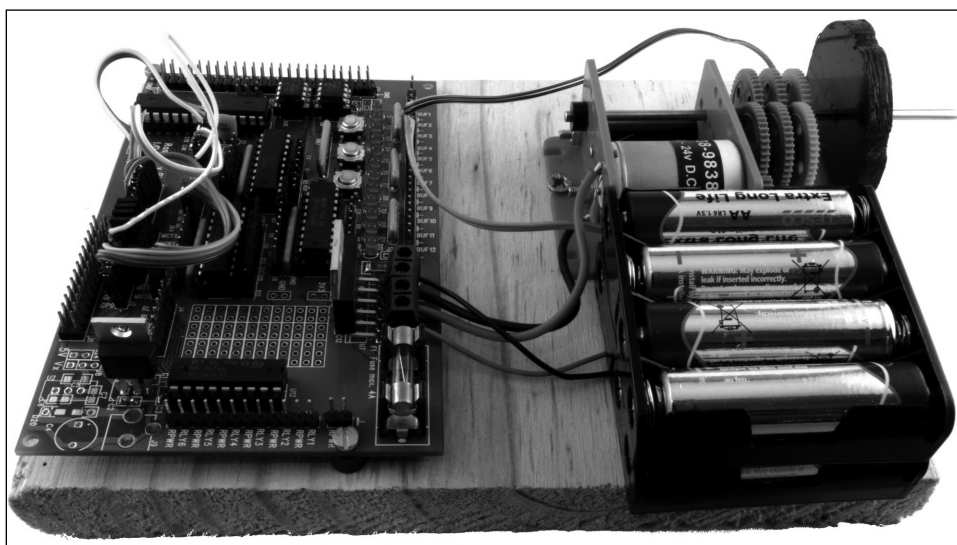


Obrázek 13.8: Deska Gertboard připojená k počítači Raspberry Pi

Pravděpodobně nejdůležitější funkcí desky Gertboard je ochrana počítače Pi, kterou poskytuje. Díky regulátoru 3,3 V zabráňuje návrh desky Gertboard tomu, aby bylo na port GPIO počítače Pi přivedeno napětí, jaké nedokáže zvládnout. Podobně jako deska Slice of Pi se také pokouší omezit přístup k šesti pinům označeným Do Not Connect a také k pinům 3,3 V a 5 V, aby nedošlo k náhodnému zkratování. Při zapojování obvodů je sice nutné dbát opatrnosti,

ale při dodržování pokynů v uživatelské příručce desky Gertboard je zaručeno, že počítač Pi nebude poškozen.

Funkce motorového kontroléru desky Gertboard umožňuje počítač Pi snadno integrovat do robotických projektů. Hodí se pro motory do napětí 48 V s proudem 4 A, které jsou dostatečně výkonné, aby mohly pohánět malé robotické vozidlo. Motor bohužel není standardní součástí dodávky desky Gertboard. Pokud váš projekt zahrnuje řízení motoru, musíte desku zkombinovat s vlastním kontrolérem (doporučuje se sada L6203). Po připojení motorového kontroléru na své místo lze pomocí desky Gertboard ovládat překvapivě silné motory buď pomocí instrukcí odesílaných ze samotného počítače Pi, nebo z volitelného modulu mikrokontroléru Atmel. Obrázek 13.9 znázorňuje ukázkový obvod, který pomocí desky Gertboard ovládá motor pro napětí 12 V napájený bateriemi.



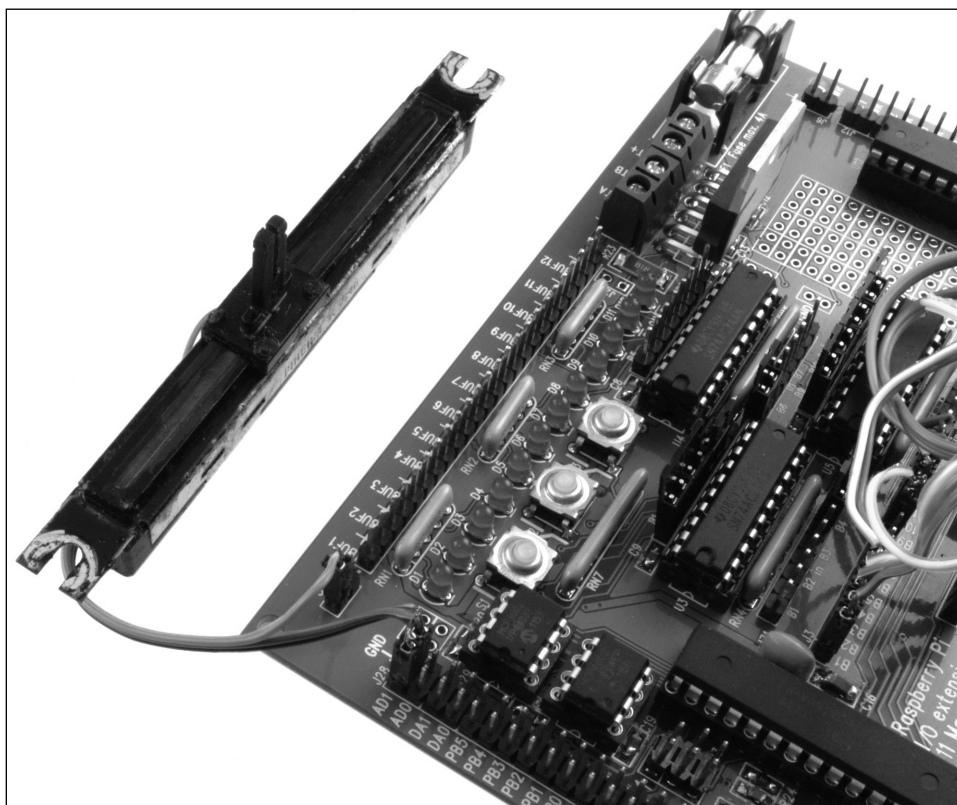
Obrázek 13.9: Použití motorového kontroléru desky Gertboard

Při práci na projektech nabízí deska Gertboard značně více možností než samotný počítač Raspberry Pi. Dvanáct vstupně-výstupních portů s vyrovnávací pamětí, které se nacházejí v horní části desky, je možné konfigurovat jako vstupy či výstupy a poskytují lepší konektivitu než sedm nebo osm univerzálních vstupně-výstupních pinů, které jsou součástí standardního portu GPIO počítače Pi. Integrované diody LED, které informují o tom, zda je pin ve vysokém či nízkém stavu, usnadňují řešení potíží s obvody a také pomáhají při vzdělávání v oblasti elektroniky, protože dovolují snadno sledovat, které z různých vstupů a výstupů jsou v danou chvíli aktivní.

U projektů detekce nebo poskytování zpětné vazby lze s výhodou uplatnit komponenty převodníků ADC a DAC desky Gertboard. Oproti pinům portu GPIO samotného počítače Pi, které mohou přijímat a odesílat pouze digitální signály, jsou na desce Gertboard k dispozici dvě připojení převodníku ADC a dvě připojení převodníku DAC. Piny ADC, které jsou umístěny

v levém horním rohu desky Gertboard, umožňují převádět signály z analogových komponent na digitální signály kompatibilní s počítačem Pi. Jako příklad je na obrázku 13.10 znázorněno, jak lze pomocí pinů převodníku ADC načítat stav *potenciometru*, což je komponenta, která mění svou rezistanci podle nastavení posuvného nebo otočného ovladače. Tento obvod by bylo možné použít k ovládání hlasitosti počítače Pi v aplikaci multimediálního centra nebo k nastavení rychlosti připojeného motoru. Piny DAC poskytují opačnou funkci: přebírají digitální signál z počítače Pi a převádějí jej na analogový. Mohou tak ovládat reproduktor, který vytváří zvuk, nebo měnit rychlost motoru či jas diody LED.

Pro složitější projekty nabízí deska Gertboard budič otevřeného kolektoru. Tato komponenta pomocí tranzistorů zapíná a vypíná zařízení, která mají odlišné požadavky na napájení než 3,3 V použité deskou Gertboard nebo která odebírají velký proud a vyžadují ke svému fungování externí zdroj napájení. Budič otevřeného kolektoru může řídit zdroj napájení až šesti zařízení současně, aniž by bylo potřebné instalovat další hardware. Deska Gertboard díky tomu poskytuje značnou flexibilitu, ačkoli stejného cíle lze dosáhnout pomocí tranzistorů nebo relé s pasivními rozšiřujícími deskami, jako je Prototyping Pi Plate.



Obrázek 13.10: Připojení potenciometru k pinům převodníku ADC desky Gertboard

Deska Gertboard nakonec poskytuje přístup k mikrokontroléru s 28 piny, který lze připojit k patici umístěné na vlastní desce. Po připojení je možné desku Gertboard propojit tak, aby adresovala buď port GPIO počítače Raspberry Pi, mikrokontrolér, nebo kombinaci obou. Deska Gertboard je navržena tak, aby spolupracovala s mikrokontroléry Atmel ATmega168 nebo ATmega328, což jsou stejné mikrokontroléry, jaké se používají v projektu prototypování Arduino. Po připojení lze psát programy pro mikrokontrolér ATmega desky Gertboards v prostředí Atmel IDE, které počítači Pi poskytuje užitečnou úroveň kompatibility s širokou škálou existujících programů pro mikrokontrolér ATmega, které jsou k dispozici ke stažení. Počítač Pi také získává schopnost adresovat externí zařízení v režimu reálného času, protože odpovědnost za hardwarové rozhraní přebírá mikrokontrolér, který počítači Pi jednoduše poskytuje zpětnou vazbu.

Deska Gertboard je velmi pokročilé zařízení a nehodí se pro každý projekt. V případě mnoha jednoduchých aplikací bude obvykle postačovat levnější pasivní rozšiřující deska typu Slice of Pi. Pokud však chcete do hloubky zkoumat, co vše procesor BC2835 počítače Raspberry Pi umožňuje nebo chcete mít jistotu, že dokážete ke svému počítači Pi připojit téměř každý hardware, jedná se o skvělou volbu a zásadní rozšíření již tak působivých možností počítače Pi. Deska Gertboard je dostupná ve webovém obchodě Farnell na adrese <http://www.farnell.com/>.

ČÁST **IV**

PŘÍLOHY

V této části:

- **PŘÍLOHA A** – Recepty jazyka Python
- **PŘÍLOHA B** – Režimy zobrazení HDMI

Recepty jazyka Python

Následující výpisy obsahují kódy programů z příkladu 3 a příkladu 4 v kapitole 11, „Úvod do jazyka Python“, a ukázkové řešení programu na kombinaci vstupu a výstupu, který byl zmíněn v kapitole 12, „Hackování hardwaru“. Všechny recepty jsou k dispozici ke stažení na stránce knihy Raspberry Pi na adrese <http://knihy.cpress.cz/K2123>. Jestliže kód zadáváte ručně, všimněte si symbolu ⇐: znamená, že řádek je zalomen kvůli nedostatku místa v knize. Pokud na konci řádku uvidíte tento symbol, stiskněte klávesu Enter teprve po dosažení konce řádku, kde tento symbol chybí.

Had s malinou (kapitola 11, příklad 3)

```
#!/usr/bin/env python
# Had s malinou (orig. Raspberry Snake)
# napsal Gareth Halfacree pro knihu Raspberry Pi Uživatelská příručka

import pygame, sys, time, random
from pygame.locals import *

pygame.init()
taktovaciFrekvence = pygame.time.Clock()

herniPlocha = pygame.display.set_mode((640, 480))
pygame.display.set_caption('Had s malinou')

cervenaBarva = pygame.Color(255, 0, 0)
cernaBarva = pygame.Color(0, 0, 0)
bilaBarva = pygame.Color(255, 255, 255)
sedaBarva = pygame.Color(150, 150, 150)
polohaHada = [100,100]
castiHada = [[100,100],[80,100],[60,100]]
polohaMaliny = [300,300]
malinaVyrostla = 1
smer = 'doprava'
zmenaSmeru = smer
```

```
def konecHry():
    konecHryPismo = pygame.font.Font('freesansbold.ttf', 72)
    konecHryPlocha = konecHryPismo.render ↵
    ('Konec hry', True, sedaBarva)
    konecHryObd = konecHryPlocha.get_rect()
    konecHryObd.midtop = (320, 10)
    herniPlocha.blit(konecHryPlocha, konecHryObd)
    pygame.display.flip()
    time.sleep(5)
    pygame.quit()
    sys.exit()

while True:
    for event in pygame.event.get():
        if event.type == QUIT:
            pygame.quit()
        elif event.type == KEYDOWN:
            if event.key == K_RIGHT or event.key == ord('d'):
                zmenaSmeru = 'doprava'
            if event.key == K_LEFT or event.key == ord('a'):
                zmenaSmeru = 'doleva'
            if event.key == K_UP or event.key == ord('w'):
                zmenaSmeru = 'nahoru'
            if event.key == K_DOWN or event.key == ord('s'):
                zmenaSmeru = 'dolů'
            if event.key == K_ESCAPE:
                pygame.event.post(pygame.event.Event(QUIT))
        if zmenaSmeru == 'doprava' and not smer == 'doleva':
            smer = zmenaSmeru
        if zmenaSmeru == 'doleva' and not smer == 'doprava':
            smer = zmenaSmeru
        if zmenaSmeru == 'nahoru' and not smer == 'dolů':
            smer = zmenaSmeru
        if zmenaSmeru == 'dolů' and not smer == 'nahoru':
            smer = zmenaSmeru
        if smer == 'doprava':
            polohaHada[0] += 20
        if smer == 'doleva':
            polohaHada[0] -= 20
        if smer == 'nahoru':
            polohaHada[1] -= 20
        if smer == 'dolů':
            polohaHada[1] += 20
        castiHada.insert(0, list(polohaHada))
        if polohaHada[0] == polohaMaliny[0] and ↵
        polohaHada[1] == polohaMaliny[1]:
```

```

        malinaVyrostla = 0
    else:
        castiHada.pop()
    if malinaVyrostla == 0:
        x = random.randrange(1,32)
        y = random.randrange(1,24)
        polohaMaliny = [int(x*20),int(y*20)]
    malinaVyrostla = 1
    herniPlocha.fill(cernaBarva)
    for poloha in castiHada:
        pygame.draw.rect(herniPlocha,bilaBarva,Rect ←
        (poloha[0], poloha[1], 20, 20))

    pygame.draw.rect(herniPlocha,cervenaBarva,Rect ←
    (polohaMaliny[0], polohaMaliny[1], 20, 20))
    pygame.display.flip()
    if polohaHada[0] > 620 or polohaHada[0] < 0:
        konecHry()
    if polohaHada[1] > 460 or polohaHada[1] < 0:
        for teloHada in castiHada[1:]:
            if polohaHada[0] == teloHada[0] and ←
            polohaHada[1] == teloHada[1]:
                konecHry()

    taktovaciFrekvence.tick(30)

```

Seznam uživatelů IRC (kapitola 11, příklad 4)

```

#!/usr/bin/env python
# Seznam uzivatelu IRC (orig. IRC User List)
# napsal Tom Hudson pro knihu Raspberry Pi Uzivatelska prirucka
# http://tomhudson.co.uk/

import sys, socket, time

RPL_NAMREPLY    = '353'
RPL_ENDOFNAMES = '366'

irc = {
    'hostitel':    'chat.freenode.net',
    'port':        6667,
    'kanal':        '#raspiuserguide',
    'intervaljmen': 5
}

```

```

uzivatel = {
    'prezdivka':      'botnick',
    'uzivatelskejmeno': 'botuser',
    'nazevhostitele':  'localhost',
    'navezserveru':    'localhost',
    'skutecnejmeno':   'Raspberry Pi Names Bot'
}

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

print 'Probiha pripojeni k serveru %(hostitel)s:%(port)s...' % irc
try:
    s.connect((irc['hostitel'], irc['port']))
except socket.error:
    print 'Doslo k chybe pripojeni k serveru IRC ↵
    %(hostitel)s:%(port)s' % irc
    sys.exit(1)

s.send('NICK %(prezdivka)s\r\n' % uzivatel)
s.send('USER %(uzivatelskejmeno)s %(nazevhostitele)s %(navezserveru)s : ↵
%(skutecnejmeno)s\r\n' % uzivatel)
s.send('JOIN %(kanal)s\r\n' % irc)
s.send('NAMES %(kanal)s\r\n' % irc)

prijimaci_pamet = ''
jmena = []

while True:
    prijimaci_pamet += s.recv(1024)
    radky = prijimaci_pamet.split('\r\n')
    prijimaci_pamet = radky.pop()
    for radek in radky:
        odpoved = radek.rstrip().split(' ', 3)
        kod_odpovedi = odpoved[1]
        if kod_odpovedi == RPL_NAMREPLY:
            seznam_jmen = odpoved[3].split(':')[1]
            jmena += seznam_jmen.split(' ')
        if kod_odpovedi == RPL_ENDOFNAMES:
            print '\r\nUzivatele pro %(kanal)s:' % irc
            for jmeno in jmena:
                print jmeno
            jmena = []
            time.sleep(irc['intervaljmen'])
            s.send('NAMES %(kanal)s\r\n' % irc)

```

Vstup a výstup portu GPIO (kapitola 12)

```
#!/usr/bin/env python
# Příklad vstupu a výstupu portu GPIO pocítace Raspberry Pi
# (orig. Raspberry Pi GPIO Input/Output example)
# napsal Gareth Halfacree pro knihu Raspberry Pi Uživatelská příručka
```

```
import RPi.GPIO as GPIO
```

```
GPIO.setup(11, GPIO.OUT)
```

```
GPIO.setup(12, GPIO.IN)
```

```
GPIO.output(11, False)
```

```
while True:
```

```
    vstupni_hodnota = GPIO.input(12)
```

```
    if vstupni_hodnota == False:
```

```
        print "Tlačítko bylo stisknuto. Dioda LED se rozsvěcuje."
```

```
        GPIO.output(11, True)
```

```
        while vstupni_hodnota == False:
```

```
            vstupni_hodnota = GPIO.input(12)
```

```
        print "Tlačítko bylo uvolněno. Dioda LED zhasína."
```

```
    if vstupni_hodnota == True:
```

```
        GPIO.output(11, False)
```


Režimy zobrazení HDMI

Pomocí hodnot v tabulkách B.1 a B.2 pro parametr `hdmi_mode` v souboru `config.txt` lze nastavit výstup videa HDMI. Další informace naleznete v kapitole 6, „Konfigurace počítače Raspberry Pi“.

Tabulka B.1: První skupina HDMI (CEA)

Hodnota	Popis
1	VGA (640x480)
2	480p 60 Hz
3	480p 60 Hz (poměr stran 16:9)
4	720p 60 Hz
5	1080i 60 Hz
6	480i 60 Hz
7	480i 60 Hz (poměr stran 16:9)
8	240p 60 Hz
9	240p 60 Hz (poměr stran 16:9)
10	480i 60 Hz (povoleno zečtyřnásobení pixelů)
11	480i 60 Hz (povoleno zečtyřnásobení pixelů) (poměr stran 16:9)
12	240p 60 Hz (povoleno zečtyřnásobení pixelů)
13	240p 60 Hz (povoleno zečtyřnásobení pixelů) (poměr stran 16:9)
14	480p 60 Hz (povoleno zdvojnásobení pixelů)
15	480p 60 Hz (povoleno zdvojnásobení pixelů) (poměr stran 16:9)
16	1080p 60 Hz
17	576p 50 Hz
18	576p 50 Hz (poměr stran 16:9)
19	720p 50 Hz
20	1080i 50 Hz
21	576i 50 Hz
22	576i 50 Hz (poměr stran 16:9)

Hodnota	Popis
23	288p 50 Hz
24	288p 50 Hz (poměr stran 16:9)
25	576i 50 Hz (povoleno zečtyřnásobení pixelů)
26	576i 50 Hz (povoleno zečtyřnásobení pixelů) (poměr stran 16:9)
27	288p 50 Hz (povoleno zečtyřnásobení pixelů)
28	288p 50 Hz (povoleno zečtyřnásobení pixelů) (poměr stran 16:9)
29	576p 50 Hz (povoleno zdvojnásobení pixelů)
30	576p 50 Hz (povoleno zdvojnásobení pixelů) (poměr stran 16:9)
31	1080p 50 Hz
32	1080p 24 Hz
33	1080p 25 Hz
34	1080p 30 Hz
35	480p 60 Hz (povoleno zečtyřnásobení pixelů)
36	480p 60 Hz (povoleno zečtyřnásobení pixelů) (poměr stran 16:9)
37	576p 50 Hz (povoleno zečtyřnásobení pixelů)
38	576p 50 Hz (povoleno zečtyřnásobení pixelů) (poměr stran 16:9)
39	1080i 50 Hz (omezené vymazávání)
40	1080i 100 Hz
41	720p 100 Hz
42	576p 100 Hz
43	576p 100 Hz (poměr stran 16:9)
44	576i 100 Hz
45	576i 100 Hz (poměr stran 16:9)
46	1080i 120 Hz
47	720p 120 Hz
48	480p 120 Hz
49	480p 120 Hz (poměr stran 16:9)
50	480i 120 Hz
51	480i 120 Hz (poměr stran 16:9)
52	576p 200 Hz
53	576p 200 Hz (poměr stran 16:9)
54	576i 200 Hz
55	576i 200 Hz (poměr stran 16:9)
56	480p 240 Hz
57	480p 240 Hz (poměr stran 16:9)

Hodnota	Popis
58	480i 240 Hz
59	480i 240 Hz (poměr stran 16:9)

Tabulka B.2: Druhá skupina HDMI (DMT)

Hodnota	Popis
1	640×350 85 Hz
2	640×400 85 Hz
3	720×400 85 Hz
4	640×480 60 Hz
5	640×480 72 Hz
6	640×480 75 Hz
7	640×480 85 Hz
8	800×600 56 Hz
9	800×600 60 Hz
10	800×600 72 Hz
11	800×600 75 Hz
12	800×600 85 Hz
13	800×600 120 Hz
14	848×480 60 Hz
15	1024×768 43 Hz, nekompatibilní s počítačem Raspberry Pi
16	1024×768 60 Hz
17	1024×768 70 Hz
18	1024×768 75 Hz
19	1024×768 85 Hz
20	1024×768 120 Hz
21	1152×864 75 Hz
22	1280×768 (omezené vymazávání)
23	1280×768 60 Hz
24	1280×768 75 Hz
25	1280×768 85 Hz
26	1280×768 120 Hz (omezené vymazávání)
27	1280×800 (omezené vymazávání)
28	1280×800 60 Hz
29	1280×800 75 Hz
30	1280×800 85 Hz

Hodnota	Popis
31	1280×800 120 Hz (omezené vymazávání)
32	1280×960 60 Hz
33	1280×960 85 Hz
34	1280×960 120 Hz (omezené vymazávání)
35	1280×1024 60 Hz
36	1280×1024 75 Hz
37	1280×1024 85 Hz
38	1280×1024 120 Hz (omezené vymazávání)
39	1360×768 60 Hz
40	1360×768 120 Hz (omezené vymazávání)
41	1400×1050 (omezené vymazávání)
42	1400×1050 60 Hz
43	1400×1050 75 Hz
44	1400×1050 85 Hz
45	1400×1050 120 Hz (omezené vymazávání)
46	1440×900 (omezené vymazávání)
47	1440×900 60 Hz
48	1440×900 75 Hz
49	1440×900 85 Hz
50	1440×900 120 Hz (omezené vymazávání)
51	1600×1200 60 Hz
52	1600×1200 65 Hz
53	1600×1200 70 Hz
54	1600×1200 75 Hz
55	1600×1200 85 Hz
56	1600×1200 120 Hz (omezené vymazávání)
57	1680×1050 (omezené vymazávání)
58	1680×1050 60 Hz
59	1680×1050 75 Hz
60	1680×1050 85 Hz
61	1680×1050 120 Hz (omezené vymazávání)
62	1792×1344 60 Hz
63	1792×1344 75 Hz
64	1792×1344 120 Hz (omezené vymazávání)
65	1856×1392 60 Hz

Hodnota	Popis
66	1856×1392 75 Hz
67	1856×1392 120 Hz (omezené vymazávání)
68	1920×1200 (omezené vymazávání)
69	1920×1200 60 Hz
70	1920×1200 75 Hz
71	1920×1200 85 Hz
72	1920×1200 120 Hz (omezené vymazávání)
73	1920×1440 60 Hz
74	1920×1440 75 Hz
75	1920×1440 120 Hz (omezené vymazávání)
76	2560×1600 (omezené vymazávání)
77	2560×1600 60 Hz
78	2560×1600 75 Hz
79	2560×1600 85 Hz
80	2560×1600 120 Hz (omezené vymazávání)
81	1366×768 60 Hz
82	1920×1080 (1080p) 60 Hz
83	1600×900 (omezené vymazávání)
84	2048×1152 (omezené vymazávání)
85	1280×720 (720p) 60 Hz
86	1366×768 (omezené vymazávání)

Rejstřík

A

Access Point, 72
Adafruit, 203
adaptér, kompatibilní, 59
ADC, 207
adresář
– bin, 51
– boot, 50
– dev, 51
– etc, 51
– home, 51
– lib, 51
– media, 51
– mnt, 51
– virtuální, 50
AES, 74
analogově-digitální převodník, 207
animace, 137
apt, 52
Arch Linux, 27
ARM, 25
ARMv6, 26
auto-MDI, 38
automatická změna velikosti, 81

B

BBC, 17
Bcast, 61
BCM2835, 25
bezdrátové
– připojení, 38
– připojení k síti, 68
bitová kopie, 86
blikání diody LED, 184
booleovská logika, 144
boot, 50
bricking, 96
bridge network, 38
broadcast, 146
Broadcom, 16

C

cd, 43
cfdisk, 77
CISC, 25
Ciseco, 199
cloudová aplikace, 115
CRT, 28
cyklus, 154

D

DAC, 207
Debian, 27
– představení, 44
démon, 124
deska
– rozšiřující, 199
– s tištěnými spoji, 192
detekce tlačítka, 188
DHCP, 38
diagnostika
– klávesnice, 57
– myši, 57
– napájení, 58
– síť, 61
– spouštění, 60
– zobrazení, 60
digitálně-analogový převodník, 207
dioda, blikání, 184
Disk Google, 116
displej, připojení, 28
dmesg, 69
– zřetěžení výstupu, 69
DNS, 66
domácí kino, 107
DSI, 30
DVI-D, 29

E

elektronické příslušenství, 173
email, 168
emergency kernel, 63
ESSID, 72
EXT4, 103

externí

- úložiště, 36
- úložné zařízení, 47

F

fdisk, 33, 78
Fedora Remix, 27
Fen Logic, 206
File Manager, 45
firmware, 68
– bezdrátový, 71
firmware-ralink, 71
flooding, 170
fsta, 80
FTP, 168
fyzická struktura, 51

G

Gertboard, 206
Gimp, 120
gparted, 82
GPIO, 20, 173, 179, 182
– vstup a výstup, 217
– výstup portu, 184
groups, 49
GUI, 41–42

H

H.264, 109
had, 213
hardware
– hackování, 173
– nastavení, 91
HDMI, 18, 29, 219

– režimy zobrazení, 219
HDTV, 29
Hello World, 134, 149
historie, 15
hledání software, 53
home, 51
houba, 174
hra, 140
HTML, 15
HTPC, 109
hudba, přehrávání, 107
Hwaddr, 61

CH

chmod, 153
chyba, syntaktická, 151

I

I²C, 181
IDLE, 46
IDLE 3, 46
iface, 65
ifconfig, 125
inet addr, 61
instalace
– LAMP, 123
– Python, 183
– software, 52–53
– WordPress, 127
Interface Name, 72
Internet, 46
IP, 38
IRC, 168
iwconfig, 73
iwlist, 71

J

jádro, nouzové, 63
JPG, 122

K

kabel, překřížený, 37
kabelové
– připojení, 36
– připojení k síti, 65
kancelářský počítač, 115
karta, SD, 32, 86
kino, domácí, 107
klávesnice, 46
– diagnostika, 57
– osvětlení, 57
– připojení, 31
kleště, 174
kolize postav, 143
komentář, 154
kompatibilní adaptér, 59
komponenta, získávání, 177
kompozitní, video, 28
komunita, 19
konektivita, síťová, 166
konektor, RCA, 28
konfigurace, 91
– síť, 65
– zobrazení, 92
kopie, bitová, 86

L

LAMP, 123
– instalace, 123
LEGO, 146
less, 71

LibreOffice, 119

Link encap, 61

Linux, 27

– přehled, 41

– správa, 41

– základy systému, 43

LOGO, 20

ls, 43

LXDE, 44

LXMusic, 107

LXTerminal, 45

M

MAC, 61

Mask, 62

měděná stopa, 193

metabalíček, 118

mezipaměť, 53

– L2, 99

Microsoft Office, 118

Midori, 46

mkdir, 44

moc, 107

modul

– sys, 166

– time, 166

monitor, nastavení, 47

motorový kontrolér, 208

mounting, 47

MTU, 62

multimetr, 58, 175

mv, 43

MySQL, 123

myš, 46

– diagnostika, 57

– připojení, 31

N

napájení

– diagnostika, 58

– připojení, 39

napětí, regulátor, 180

nápověda, 47

NAS, 112

nastavení

– hardware, 91

– monitoru, 47

– přepětí, 97

– přetaktování, 96

– software, 101

negotiation, 58

nepolarizované zařízení, 185

NetSurf Web Browser, 46

NFS, 112

Nibbles, 158

nouzové jádro, 63

O

oddíl, 51

– automatická změna velikosti, 81

– ruční změna velikosti, 82

– správa, 77

– vytvoření, 77

– změna velikosti, 81

ODF, 119

odsávací knot, 175

Office 365, 116

Openbox Configuration Manager, 47

OpenOffice.org, 118

OS X, 27, 34

OTG, 101

overscan, 92

P

- pacman, 52
- páječka, 174
- pájení, 194
- pájka, 174
- paměť, rozdělení, 100
- Parted Magic, 82
- partition, 51
- PHP, 123
- PicoBoard, 146
- pinzeta, 175
- plovoucí pin, 189
- PNG, 122
- počítač, kancelářský, 115
- pokovení, 196
- POP3, 168
- port
 - GPIO, 179
 - USB, 31
- portování, 27
- potenciometr, 209
- potíže, řešení, 57
- Power Management, 73
- pracovní stojan, 175
- print, 155
- programování, 13, 46
- proměnná, 154–155
- propojovací
 - kabel, 173
 - vodič, 199
- Prototyping Pi Plate, 203
- průchodka, 59
- přehrávání hudby, 107
- překřížený kabel, 37
- přepětí, nastavení, 97
- přetaktování, 95
 - nastavení, 96
- přetěžování, 170
- příjímací vyrovnávací paměť, 168
- příkaz
 - except, 167
 - print, 155
 - try, 167
- připojení
 - bezdrátové, 38
 - displeje, 28
 - k bezdrátové síti, 76
 - k síti, 37
 - kabelové, 36
 - klávesnice, 31
 - myši, 31
 - napájení, 39
 - zvukové, 30
- příslušenství, elektronické, 173
- pygame, 158
- Python, 149
 - GPIO, 182

R

- Raspberry Pi, 16
- Raspbmc, 109, 113
- raspi-config, 81
- RCA, 28
- receive buffer, 168
- regulátor napětí, 180
- repozitář, 53
- resolv.conf, 67
- respawning, 164
- rezistor, 174
 - barevné kódy, 175
- režim
 - jádra, 101

- testovací, 99
- rm, 43
- rmdir, 44
- robotika, 146
- LEGO, 146
- root, 49
- Root Terminal, 45
- rozbočovač USB, 31
- rozdělení paměti, 100
- rozšiřující deska, 199
- RTS, 73
- ruční změna velikosti, 82

Ř

- řešení potíží, 57
- řezačka stop, 194

S

- sběrnice, 181
- I²C, 181
- SPI, 182
- UART, 181
- SCL, 181
- Scratch, 45, 133
- senzor, 146
- server, webový, 123
- seznam uživatelů IRC, 215
- Shake, 158
- síť
- bezdrátové připojení, 68
- diagnostika, 61
- kabelové připojení, 65
- konfigurace, 65
- síťová konektivita, 166
- skupina, 49

- Slice of Pi, 199
- SMB, 112
- smyčka, zpětná, 61
- SoC, 95
- software
- hledání, 53
- instalace, 52–53
- nastavení, 101
- odinstalace, 54
- upgrade, 55
- soubor, spustitelný, 153
- SPI, 182
- spouštění
- diagnostika, 60
- možnosti, 95
- správa
- oddílů, 77
- systému Linux, 41
- správce balíčků, 52
- spustitelný soubor, 153
- Squeak, 46
- SSID, 74
- start.elf, 100
- stojan, 174
- streamování
- internetových médií, 110
- médií, 112
- stripboard, 174, 192
- syntaktická chyba, 151
- syntaxe, zvýrazňování, 151
- systém
- na SD kartě, 32
- souborů, 50
- System Tools, 46
- systémové nástroje, 46

Š

šifrování

- AES, 74
- bezdrátové sítě, 74
- WEP, 74–75
- WPA, 74–75
- WPA2, 74–75

T

Task Manager, 46

tavidlo, 194

terminál, 42

testovací režim, 99

ThinkFree Online, 116

tlačítko, 174

- detekce, 188

U

UART, 181

úložiště, externí, 36

UMS, 47

upgrade software, 55

UPnP, 112

USB, 18

- port, 31
- rozbočovač, 31

uživatel, 49

uživatelský účet, 41, 48

V

velikost oddílů, 81

Veroboard, 192

VFAT, 51

virtuální adresář, 50

vodič, 173

- propojovací, 199

voltmetr, 58

vratná pojistka, 57

vstup, 154

- GPIO, 217

vyjednání, 58

vypnutí mezipaměti L2, 99

výpočet mezních hodnot rezistorů, 184

vysílání zpráv, 146

výstup, GPIO, 217

vytvoření

- bitové kopie systému, 87–88
- nového oddílu, 77
- uživatelského účtu, 48

W

webový server, 123

WEP, 74–75

while, 156

Wi-Fi, 38

Win32 Disk Imager, 35

Windows, 27, 35

WordPress, instalace, 127

WPA, 74–75

WPA2, 74–75

X

X Window, 44

x86, 25

Xarchiver, 45

XCF, 122

Y

yum, 52

Z

zařízení, nepolarizované, 185

zdroje

– offline, 177

– online, 177

ZIP, 33

získávání komponent, 177

zkušební model obvodu, 173, 191

zobrazení

– diagnostika, 60

– úprava, 92

Zoho, 116

zpětná smyčka, 61

zvuk, 137

zvukové připojení, 30

zvýrazňování syntaxe, 151