

Obsah

Předmluva	17
-----------------	----

I. Kontext 25

1. Filozofické otázky	27
-----------------------------	----

Kultura? Jaká kultura?	27
------------------------------	----

Životnost systému Unix	28
------------------------------	----

Fakta hovořící proti studiu unixové kultury	29
---	----

Co je na Unixu špatného	30
-------------------------------	----

Co je na Unixu dobrého	31
------------------------------	----

Software s otevřeným aplikačním prostředím	31
--	----

Přenositelnost aplikací nezávisle na platformě a otevřené standardy	32
---	----

Internet a síť WWW	32
--------------------------	----

Komunita otevřeného aplikačního prostředí	33
---	----

Nic než flexibilita	33
---------------------------	----

Improvizace je v Unixu veselou zábavou	34
--	----

To, co si odnesete z hodin Unixu, můžete použít kdekoli	35
---	----

Základy unixové filozofie	35
---------------------------------	----

Pravidlo modularity: Pište jednoduché součásti, které lze propojit přes bezchybná rozhraní.	37
---	----

Pravidlo srozumitelnosti: Srozumitelnost je lepší než chytrost.	38
--	----

Pravidlo kompozice: Navrhujte programy tak, aby je bylo možné připojit k dalším programům.	38
--	----

Pravidlo rozdělení: Oddělujte zásadu od mechanismu, rozhraní pak od toho, co je za ním.	39
---	----

Pravidlo jednoduchosti: Navrhujte jednoduché programy, složitější pište pouze, není-li jiného východiska.	40
---	----

Pravidlo šetrnosti: Velký program pište pouze v případě, když odzkoušíte, že jiná možnost není.	41
---	----

Pravidlo transparentnosti: Navrhujte programy tak, aby bylo co nejsnazší jejich ladění a kontrola.	41
--	----

Pravidlo robustnosti: Robustnost je dítětem transparentnosti a jednoduchosti.	42
--	----

Pravidlo reprezentace: Zahaňte znalosti do dat tak, aby logika programu mohla být hloupá, stejně jako robustní.	43
---	----

Pravidlo nejmenšího překvapení: Při návrhu rozhraní se snažte, aby obsahovalo co nejméně překvapení.	43
--	----

Pravidlo ticha: Nemá-li program na srdci něco důležitého, nemá raději říkat nic.	44
---	----

Pravidlo opravy: Máte-li selhat, selžete hlasitě a co nejdříve.	44
--	----

Pravidlo hospodárnosti: Programátorův čas je drahý.	
Raději tedy šetřete jeho čas než čas počítače.	45
Pravidlo generování: Vyhýbejte se ručním improvizacím. Můžete-li, píšte programy, které budou psát další programy.	46
Pravidlo optimalizace: Nejprve vytvořte prototyp, zprovozněte jej a pak se teprve pusťte do jeho optimalizace.	46
Pravidlo rozmanitosti: Nedůvěřujte všem tvrzením typu „jediný správný způsob“.	47
Pravidlo rozšiřitelnosti: Navrhujte pro budoucnost, protože ta nastane dříve, než se nadějete.	48
Unixová filozofie v jedné lekci	48
Uplatňujeme unixovou filozofii	48
Počítá se rovněž postoj	49
2. Příběh dvou kultur	51
Původ a historie Unixu, 1969-1995	51
Genesis: 1969-1971	52
Exodus: 1971-1980	54
TCP/IP a unixové války 1980-1990	56
Povstání proti nadvládě: 1991-1995	62
Původ a historie hackerů, 1961-1995	64
Při hře ve studentských lesících: 1961-1980	65
Internetová fúze a hnutí svobodného softwaru: 1981-1991	66
Linux a reakce pragmatika: 1991-1998	68
Hnutí otevřeného zdroje: od roku 1998 dodnes	70
Poučení z unixových dějin	72
3. Kontrasty: Porovnání unixové filozofie s ostatními	73
Charakteristické prvky stylu operačního systému	73
Co je jednotícím záměrem operačního systému?	74
Souběžné zpracování více úloh – multitasking	74
Spolupracující procesy	75
Vnitřní hranice	76
Atributy souborů a struktury záznamů	77
Binární souborové formáty	78
Upřednostňovaný styl uživatelského rozhraní	78
Budoucí uživatelé	79
Bariéry přechodu na úroveň vývojáře	79
Porovnání operačních systémů	80
VMS	82
MacOS	83
OS/2	85
Windows NT	87

BeOS	90
MVS	91
VM/CMS	94
Linux	96

Tak dlouho se chodí se džbánem pro vodu, až se ucho utrhne	97
--	----

II. Návrh 101

4. Modularita: Ať je to bez chyb, ať je to jednoduché	103
Zapouzdření a optimální velikost modulu	105
Kompaktnost a ortogonalita	107
Kompaktnost	107
Ortogonalita	109
Pravidlo jediné pravdy	111
Kompaktnost a silné jednotné ústředí	112
Hodnota nestrannosti	114
Software je mnohovrstvá záležitost	114
Shora dolů nebo zdola nahoru	115
Spojovací vrstvy	117
Případová studie: Jazyk C jako tenká spojovací vrstva	117
Knihovny	119
Případová studie: Zásuvné moduly editoru GIMP	120
Unix a objektově orientované jazyky	121
Kódování směrem k modularitě	123
5. Textualita: Dobré protokoly základem optimálních metod	125
Jak je důležité používat textový formát	127
Případová studie: Formát unixových souborů hesel	129
Případová studie: Formát .newsrsc	130
Případová studie: Grafický souborový formát PNG	131
Metaformáty datových souborů	132
Styl DSV	132
Formát RFC 822	133
Formát typu cookie	134
Záznamový formát	135
XML	136
Formát Windows INI	138
Konvence unixových formátů pro textové soubory	139
Kompresí souborů: pro a proti	141
Návrh aplikačního protokolu	142
Případová studie: SMTP, Simple Mail Transfer Protocol	143

Případová studie: POP3, Post Office Protocol	143
Případová studie: IMAP, Internet Message Access Protocol	145
Metaformáty aplikačních protokolů	146
Klasický internetový aplikační protokol	146
HTTP jako univerzální aplikační protokol	147
BEEP: Blocks Extensible Exchange Protocol	149
XML-RPC, SOAP a Jabber	149
6. Transparentnost: Budiž světlo	151
Případové studie	153
Případová studie: audacity	153
Případová studie: přepínač –v programu fetchmail	154
Případová studie: GCC	157
Případová studie: kmail	157
Případová studie: SNG	159
Případová studie: Databáze terminfo	161
Případová studie: Datové soubory hry Freeciv	164
Jak tvořit návrhy transparentních a přístupných programů	166
Zen transparentnosti	166
Jak tvořit transparentní a přístupný kód	167
Transparentnost a jak se vyhnout přepjatému ochranářství	168
Transparentnost a podoby vhodné k úpravám	169
Transparentnost, diagnostika poruch a zotavení po chybě	170
Jak navrhovat programy s ohledem na jejich údržbu	171
7. Multiprogramování	173
Jak oddělit řízení složitosti od ladění výkonu	175
Taxonomie metod pro meziprocsovou komunikaci v systému Unix	176
Přenesení úlohy na specializované programy	176
Roury, přesměrování a filtry	177
Zapouzdření	182
Bezpečnostní zapouzdření a Bernsteinovo řetězení	182
Podřízené procesy (slave processes)	184
Rovnocenná meziprocsová komunikace	185
Problémy a metody, jimž byste se měli vyhýbat	191
Stárnoucí metody unixové meziprocsové komunikace	192
Vzdálená volání procedur (RPC, Remote Procedure Calls)	193
Vlákná – hrozba, nebo nebezpečí?	195
Segmentace procesu na úrovni návrhu	197

8. Minijazyky: To prostě musí zpívat samo	199
Jak na taxonomii jazyků	201
Uplatnění minijazyků	203
Případová studie: Příkaz sng	203
Případová studie: Regulární výrazy	203
Případová studie: Program Glade	207
Případová studie: Minijazyk m4	209
Případová studie: XSLT	209
Případová studie: Nástroje tvůrce dokumentů	211
Případová studie: Syntaxe řízení běhu programu fetchmail	215
Případová studie: Minijazyk awk	216
Případová studie: PostScript	218
Případová studie: Jazyky bc a dc	219
Případová studie: Emacs Lisp	220
Případová studie: JavaScript	221
Jak navrhovat minijazyky	222
Volba vhodného nastavení složitosti	222
Rozšiřování a vkládání jazyků	224
Tvorba vlastní mluvnice	225
Makra – Pozor!	226
Jazyk nebo aplikační protokol?	227
9. Generování kódu: Jak posunout specifikaci zase o stupínek výše	229
Datové programování	230
Případová studie: Program ascii	231
Případová studie: Statistické filtrování nevyžádané pošty	232
Případová studie: Hacking metatříd v programu fetchmailconf	233
Generování jed noučelového kódu	238
Případová studie: Generování kódu pro výstup v programu ascii	238
Případová studie: Generování kódu HTML pro tabulkové seznamy	241
10. Konfigurace: Z postele je třeba vstávat pravou nohou	243
Co všechno by mělo být konfigurovatelné	243
Kde hledat konfigurační soubory	245
Řídící soubory	246
Případová studie: Soubor .netrc	248
Přenositelnost na další operační systémy	249
Proměnné prostředí	249
Uživatelské proměnné prostředí	251
Kdy použít proměnné prostředí	251

Přenositelnost na jiné operační systémy	253
Přepínače příkazového řádku	253
Přepínače příkazového řádku od -a po -z	254
Přenositelnost na jiné operační systémy	258
Jak si mezi tolika metodami vybrat	259
Případová studie: Program fetchmail	259
Případová studie: Server XFree86	261
Při porušení těchto pravidel	262
11. Rozhraní: Návrhové vzory uživatelského rozhraní v prostředí systému Unix	263
Jak uplatnit Pravidlo nejmenšího překvapení	264
Dějiny návrhu rozhraní v systému Unix	266
Vyhodnocení návrhů rozhraní	267
Výhody a nevýhody rozhraní příkazového řádku a vizuálních rozhraní	269
Případová studie: Dva způsoby tvorby programu Kalkulačka	272
Transparentnost, výmluvnost a schopnost konfigurace	274
Návrhové vzory unixového rozhraní	276
Filtry	276
Vzor kouzlo	278
Vzor zdroj	278
Vzor cíl	279
Vzor překladač	279
Vzor typu ed	280
Vzor typu rogue	280
Vzor „automat oddělený od rozhraní“	283
Vzor serveru s rozhraním příkazového řádku	288
Vzory rozhraní založených na jazyce	289
Jak uplatnit unixové návrhové vzory rozhraní	290
Vzor polyvalentního programu	290
Webový prohlížeč a univerzální uživatelské rozhraní	291
Mlčeti zlato	294
12. Optimalizace	297
Nepředstírejte činnost, raději si lehněte	297
Dvakrát měř, jednou řež	298
Co není místní, je škodlivé	300
Průchodnost versus zpoždění	301
Dávkování operací	302

Prokládání operací	303
Ukládání výsledků operací do mezipaměti	303

13. Složitost: Tak jednoduše, jak jen to jde.

Nikoli však jednodušeji.	305
-------------------------------	-----

Hovoříme-li o složitosti	306
--------------------------------	-----

Tři zdroje složitosti	306
Kompromisy mezi složitostí rozhraní a implementace	307
Nevyhnutelná, volitelná a náhodná složitost	309
Mapování složitosti	310
Když jednoduchost nestačí	311

Příběh pěti editorů	312
---------------------------	-----

Editor ed	313
Editor vi	314
Editor Sam	315
Editor Emacs	316
Editor Wily	317

Správná velikost editoru	318
--------------------------------	-----

Jak rozpoznat problémy spojené se složitostí	318
Když kompromis nefunguje	321
Je Emacs argumentem proti unixové tradici?	322

Správná velikost softwaru	324
---------------------------------	-----

I. Implementace 327

14. Jazyky: Programovat, nebo neprogramovat v jazyce C? 329

Unixová hojnost jazyků	329
------------------------------	-----

Proč ne jazyk C?	330
------------------------	-----

Interpretované jazyky a smíšené strategie	332
---	-----

Hodnocení jazyků	333
------------------------	-----

C	333
---------	-----

C++	335
-----------	-----

Příkazové prostředí	337
---------------------------	-----

Perl	340
------------	-----

Tcl	342
-----------	-----

Python	344
--------------	-----

Java	347
------------	-----

Emacs Lisp	350
------------------	-----

Jak se to vše vyvine	351
----------------------------	-----

Volba platformy pro systém X	353
------------------------------------	-----

15. Nástroje: Taktika vývoje	355
Operační systém, jenž je k vývojářům přátelský	355
Volba editoru	356
Co byste měli vědět o editoru vi	357
Co byste měli vědět o editoru Emacs	357
Antináboženská volba: Používat oba	358
Speciální generátory kódu	358
Generátory yacc a lex	359
Případová studie: Program Glade	362
Program make: Automatizace návodů	363
Teoretické základy	363
Využití nástroje make pro vývoj v jiných jazycích než C a C++	365
Pomocné oddíly	365
Jak tvořit soubory makefile	367
Systémy správy verzí	370
Proč sledování verzí?	370
Ruční správa verzí	371
Automatizovaná správa verzí	371
Unixové nástroje pro správu verzí	372
Ladění při běhu programu	375
Profilování	375
Jak kombinovat nástroje s editorem Emacs	376
Editor Emacs a program make	376
Editor Emacs a ladění při běhu programu	377
Editor Emacs a správa verzí	377
Editor Emacs a profilování	378
Jako integrované vývojové prostředí, jenže lepší	378
16. Opětne použití: Nesnažte se znovu vynalézat kolo	379
O hloupém Honzovi	380
Transparentnost jako klíč k opětovnému použití	383
Od opětovného využití k otevřenému zdroji	384
V životě jsou nejlepší věci přístupné všem	385
Kam se dívat?	387
Sporné otázky týkající se softwaru s otevřeným kódem	389
Problematika licencí	389
Co lze kvalifikovat jako otevřený zdroj	390
Standardní licence otevřeného kódu	391
Kdy budete potřebovat právníka	393

IV. Komunita 395

17. Přenosnost: Přenosný software a návod, jak držet krok se standardy	397
Vývoj jazyka C	398
Počátky jazyka C	399
Standardy jazyka C	400
Unixové standardy	402
Standardy a unixové války	402
Přízrak na hostině vítězů	405
Unixové standardy ve světě otevřeného softwaru	405
IETF a proces tvorby standardů RFC	406
Specifikace jako DNA, kód jako RNA	409
Programování přenosných aplikací	412
Přenosnost a volba jazyka	412
Jak předcházet závislostem na systému	416
Nástroje pro usnadnění přenosnosti	416
Zavádění mezinárodní podpory	417
Přenosnost, otevřené standardy a otevřené zdroje	417
18. Dokumentace: Vysvětlete svůj kód celému světu prostřednictvím webu	419
Různá pojetí dokumentace	420
Unixový styl	422
Důraz na dlouhé dokumenty	422
Kulturní styl	423
Džungle unixových dokumentačních formátů	424
Program troff a stolní nástroje tvůrce dokumentů	424
Program TEX	426
Texinfo	427
POD	427
HTML	427
DocBook	427
Dnešní chaos a možná řešení	428
DocBook	428
Definice typů dokumentů	429
Další DTD	430
Řetězec nástrojů pro práci s dokumenty DocBook	430
Migrační nástroje	433

Editační nástroje	434
Související standardy a postupy	434
SGML	434
Další odkazy na prameny týkající se formátu XML-DocBook	434
Nejlepší dostupné techniky při psaní unixové dokumentace	435
19. Otevřený zdroj: Programování v nové unixové komunitě	437
Unix a otevřený zdroj	438
Nejlepší dostupné techniky spolupráce s vývojáři otevřeného softwaru	440
Nejlepší aktualizací techniky	440
Užitečné techniky pojmenování projektů a archivů	443
Užitečné vývojové techniky	446
Užitečné techniky tvorby distribucí	449
Užitečné komunikační techniky	452
Logika udělování licencí: Pro jakou licenci se rozhodnout?	454
Proč byste měli používat standardní licenci	454
Rozmanitost udělování licencí na otevřené zdroje	455
MIT nebo Licence konsorcia X (X Consortium License)	455
Klasická licence BSD (BSD Classic License)	455
Umělecká licence (Artistic License)	456
Licence GPL (General Public License)	456
Licence MPL (Mozilla Public License)	456
20. Budoucnost: Možná nebezpečí a možné příležitosti	457
Podstata a náhoda v unixové tradici	457
Plan 9: A taková byla budoucnost	460
Sporná místa v návrhu systému Unix	462
Unixový soubor je jen velký ranec plný bajtů	462
Unixová podpora grafických uživatelských rozhraní je slabá	464
Mazání souborů je tu navěky	464
Unix předpokládá existenci statického souborového systému	465
Návrh řízení úloh nebyl zdařilý	465
Rozhraní API systému Unix nepoužívá výjimky	466
Mechanismy ioctl(2) a fcntl(2) jsou ostudou	467
Model zabezpečení systému Unix může být příliš primitivní	467
Unix má mnoho různých druhů názvů	468
Souborové systémy mohou být označeny za škodlivé	468
Směrem ke globálnímu adresovému prostoru sítě Internet	468

Problémy v unixovém prostředí	469
Problémy v unixové kultuře	471
Několik důvodů, proč věřit	473
A. Glosář zkratek	475
B. Reference	479
Bibliografie	479
C. Příspěvatelé	487
D. Kořen bez kořenů: Unixová paradoxní témata úvah	
Mistra Foo	491
Úvod redaktora	491
Mistr Foo a deset tisíc řádků	492
Mistr Foo a dítě skriptů	493
Mistr Foo a metodolog	495
Mistr Foo přednáší o grafickém uživatelském rozhraní	495
Mistr Foo a unixový fanatik	496
Mistr Foo přednáší o povaze systému Unix	496
Mistr Foo a koncový uživatel	497
Tiráž	498
Rejstřík	499