

BAKALÁŘSKÁ PRÁCE

Martin Korous



Pedagogická fakulta Jihočeské univerzity
Katedra informatiky

BSD systémy se zaměřením na FreeBSD

Bakalářská práce

Autor: Martin Korous

Vedoucí bakalářské práce: Mgr. Jiří Pech Ph. D.

České Budějovice 2006

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně s použitím literatury a zdrojů uvedených v části Použité zdroje.

V Českých Budějovicích dne

Martin Korous

Anotace

Tato bakalářská práce je o operačních systémech patřících do rodiny BSD. Jejím cílem je seznámit obecnou veřejnost s operačními systémy na Unixovém základě. V první části práce se věnuji historii Unixu a BSD. V další části jsem se zaměřil na známé operační systémy z rodiny BSD. Třetí kapitola se zabývá historií a vývojovým modelem pro desktop asi nejvhodnějšího FreeBSD. Následující část práce popisuje nejdůležitější rozdíly FreeBSD a Linuxu. V poslední části práce se věnuji praktické části instalace FreeBSD, základům jeho používání a správy.

Annotation

This bachelor work treats the operating systems that belong to the BSD family. The work objective is to introduce operating systems based on the UNIX to the public. In the first part I attend to history of UNIX and BSD. In the next part I focused on the well known operating systems from the BSD family. The third part deals with the history and developing model, for the desktop probably the most suitable FreeBSD. The following part describes the most important differences between FreeBSD and Linux. In the final part of my work I attend to the practical part of FreeBSD installation, the basics of its use and control.

Contents

1	Historie Unixu	5
1.1	Úvod	5
1.2	Unix	5
1.3	386BSD	6
2	Přehled BSD systémů	7
2.1	Mac OS X	7
2.2	NetBSD	7
2.3	OpenBSD	8
2.4	FreeSBIE	8
2.5	DragonFly BSD	8
2.6	OpenDarwin	8
2.7	m0n0wall	9
2.8	DesktopBSD	9
3	FreeBSD	10
3.1	historie	10
3.2	Cíle Projektu FreeBSD	12
3.3	Vývojový model	13
3.4	Současnost	14
4	Porovnání Linuxu a FreeBSD	16
4.1	System	16
4.1.1	Linux	16
4.1.2	FreeBSD	16
4.2	licence	17
4.2.1	BSD licence	17
4.2.2	GPL licence	17
4.3	Souborové systémy	18

4.3.1	BSD filesystemy	18
4.3.2	Linuxové souborové systémy	19
5	FreeBSD a praxe	21
5.1	instalace	21
5.1.1	výběr instalace	21
5.1.2	Instalace z CD-ROM	21
5.1.3	Číslování disků systémem BIOS	24
5.1.4	Alokace místa na disku	25
5.1.5	Výběr balíčků	28
5.1.6	Výběr instalačního media	29
5.1.7	Zahájení instalace	30
5.1.8	Po instalační nastavení	30
5.2	Instalace aplikací: balíčky a porty	31
5.2.1	Hledání programů	31
5.2.2	Používání balíčkovacího systému	33
5.2.3	Kolekce portů	35
5.2.4	Po-instalační aktivity	40
5.3	X Window System	41
5.3.1	Úvod	41
5.3.2	Konfigurace	41
5.3.3	Window Manager	43
5.4	Zvuková karta	45
5.5	Lokalizace	46
5.5.1	i18n/i10n	46
5.5.2	Používání lokalizace	46
5.6	IPv6	50
5.7	Síťové služby	51
5.7.1	Inetd	51
5.7.2	Apache HTTP Server	53
5.8	Kompilace kernelu	54
5.8.1	Proč vlastní kernel?	54
5.8.2	Konfigurace kernelu	55
5.8.3	Kompilace a instalace kernelu	57
6	Závěr	58
	Použité zdroje	59

Chapter 1

Historie Unixu

1.1 Úvod

Téma této bakalářské práce jsem si vybral, protože se o operační systémy unixového typu hodně zajímám a už delší čas je používám jako hlavní (jediný) operační systém na mém počítači. Zalíbila se mi filosofie Unixu, která říká: "Všechno je soubor". Druhé základní pravidlo zní: "Každý program dělá jen jednu věc, ale dělá jí dobře". Unix se může pro mnohé jevit jako neuspořádaná hromada programů náhodně roztroušených v mnoha adresářích, ale po bližším seznámení se v něm lze nalézt hlubší logiku. Logiku, která se vyvíjí již čtvrté desetiletí.

V této bakalářské práci se budu věnovat unixům, které patří do vývojové větve BSD. BSD vzniklo už před 28 lety na Californské univerzitě v Berkeley, z něj vznikly další systémy jako třeba OpenBSD, NetBSD nebo FreeBSD. A právě FreeBSD se budu podrobněji věnovat. Podívám se na jeho vývojový model a na rozdíly Linuxu a FreeBSD. V praktické části se budu věnovat jeho instalaci a základnímu nastavení a používání.

1.2 Unix

Původní Unix byl vytvořen roku 1969 Kevinem Thompsonem a Denisem Ritchiem v Bellových laboratořích. Unix nahradil Multics, který sice byl schopný práce, ale nedělal to, pro co byl vyvíjen - například podporu více uživatelů. Původní verze byla napsána v assembleru PDP-7, ale v roce 1973

bylo jádro přepsáno do jazyka C, který byl k tomuto účelu navržen. Vývoj unixu je rozdělen do mnoha větví, mezi nejvýznamnější patří System III (později System V) a Berkley Unix, projekt Computer Sciences Research Group Kalifornské univerzity v Berkeley (CSRG). Toto vydání bylo později přejmenované na Berkeley Software Distribution, krátce BSD. V roce 1992 byly zdrojové kódy BSD uvolněny jako "open source" s výjimkou proprietárního kódu AT&T. Tato verze nebyla použitelná, ale William Jolitz přepsal zbývající kód a uvolnil ho v roce 1992 jako 386BSD. Ve stejný čas, další skupina ex-CSRG členů zakládá komerční společnost nazvanou Berkeley Software Design Inc (BSDI), vydávající beta verzi operačního systému BSD/386.

1.3 386BSD

Autoři 386BSD byli William Jolitz a Lynne Jolitz. Lynne Jolitz byla první uživatel a kritik. Na její podnět byl BSD systém zredukován a přepsán. 386BSD vycházející z BSD se vrátil na počátek unixu, ale v moderní formě. Většina kódu pochází z 4BSD Kalifornské univerzity v Berkeley. 386BSD mělo rozškátulkovanou strukturu umožňující "roll back", počátky práce verzí dynamických komponent, které byly kompletně nezávislé. Tato zamýšlená modularita zvyšovala pružnost a práce různých částí (sdílené knihovny, nahratelné moduly do jádra) byla bez potíží. První verze byla vydána uvnitř Kalifornské univerzity a US Department of Energy v roce 1989.

Chapter 2

Přehled BSD systémů

2.1 Mac OS X

Mac OS X je poslední verze Mac OS, operačního systému pro počítače Macintosh od Apple, který kombinuje populární rysy klasického uživatelského rozhraní Macintoshů se stabilitou Unixu. Mac OS X je použitelný od Mac hardware s procesorem G3, ale plně využíváný teprve procesory G4 a G5. Mac OS X byl vytvořen kombinací Darwina a open source Unixu, který je založen na BSD a Mach microkernelu, s GUI pojmenovaným Aqua, vyvinutým společností Apple. Operační systém byl do komerčního provozu uvolněn v březnu roku 2001. Vzhled rozhraní Aqua GUI je napodobován mnoha dalšími a je i inspirací pro ostatní operační systémy. Vzhled rozhraní imitující vzhled Aqua existuje i pro mnoho programů pro Microsoft Windows, jako Winamp. Dnešní Mac OS X začal vznikat v roce 1989 jako NeXTSTEP, objektově-orientovaný operační systém vyvíjený společností Steva Jobse NeXT po jeho odchodu z Apple v roce 1986. Po neuspokojivém prodeji počítačů pro něj navržených, byl NeXTSTEP v devadesátých letech portován na řadu dalších platform, ale nikdy se nestal příliš populární.

2.2 NetBSD

NetBSD je operační systém vycházející z Berkeley Networking Release 2 (Net/2), 4.4BSD-Lite a 4.4BSD-Lite2. NetBSD pracuje na padesáti devíti různých architekturách systémů a na sedmnácti různých CPU od alphy přes procesory i386 po sparc a na dalších se pracuje. Projekt si klade za hlavní

cíl poskytovat volně dostupný a redistribuovatelný systém, který mohou profesionálové, experimentátoři, výzkumníci i kdokoli jiný požívat, jak se jim zamane.

2.3 OpenBSD

OpenBSD je multiplatformní unix-like operační systém založeným na 4.4BSD. Je to systém vyvíjený s důrazem na bezpečnost. Díky tomu se výborně hodí pro instalace na místech, která mají zajišťovat ochranu dat nebo jsou vystavena vysokému riziku. Tedy hlavně firewally, aplikační servery v "demilitarizovaných zónách", systémy detekce průniku a podobná zařízení. Proto se OpenBSD na své domovské stránce na titulní stránce pyšní napisem: Pouze jedna vzdálena díra v implicitní instalaci za 8 let. Vysoké bezpečnosti je zajištěno důkladným bezpečnostním auditem zdrojového kódu a implementací velkého množství dalších technických opatření, která se navzájem překrývají a brání zneužití chyb. OpenBSD je možno zdarma získat z jejich FTP serverů.

2.4 FreeSBIE

FreeSBIE (Free System Burned In Economy) je operační systém založeným na FreeBSD. FreeSBIE je systém na live CD, to znamená, že bootuje přímo z CD a nepotřebuje vůbec pevný disk.

2.5 DragonFly BSD

DragonFly je operační systém unixového typu vycházející z FreeBSD 4.x (tj. podobný vývoj jako u OpenBSD vycházejícího z NetBSD), který se snaží naplňovat stejné cíle jako FreeBSD 5.x ovšem podstatně jinak.

2.6 OpenDarwin

V dubnu 2002 Internet System Consortium a Apple Computer založili projekt opendarwin.org pro kooperativní vývoj Darwina. Darwin je operační systém jehož jádro je založeno na MAC OS X. OpenDarwin má za cíl vylepšit MAC OS X vytvořením komunity vývojářů, kteří pracují s a na MAC OS X.

2.7 m0n0wall

m0n0wall je projekt, jehož cílem je vytvořit kompletní firewall, který by na standardním nebo jednodeskovém PC poskytl všechny důležité vlastnosti komerčních firewallů, měl jednoduché ovládání a byl za přijatelnou cenu (free software). m0n0wall je postaven na kostře FreeBSD, webovém serveru, PHP a několika dalších utilitách. Celá konfigurace systému je uložena v jednom XML souboru, aby byla zachována maximální jednoduchost. m0n0wall je určitě první unixový systém, který svou konfiguraci zpracovává pomocí PHP a ne obvyklými shellovými skripty a který má celou konfiguraci uloženou ve formátu XML.

2.8 DesktopBSD

DesktopBSD je asi nejmladší projekt z rodiny BSD, první stabilní verze vyšla v březnu 2006. Tento operační systém kombinuje stabilitu FreeBSD a funkčnost prostředí KDE se speciálně vyvíjeným software, poskytuje operační systém snadný pro používání i instalaci.

Chapter 3

FreeBSD

3.1 historie

První verze FreeBSD vyšla v prosinci 1993, která byla vytvořena jako výsledek práce na "Unofficial 386BSD Patchkit" (sada oprav systému 386BSD) posledními třemi koordinátory zmiňované sady: Natem Williamsem, Rodem Grimesem a Jodanem K. Hubbardem. Původním cílem bylo vytvořit přechodnou vylepšenou verzi 386BSD, aby se tak vyřešilo množství problémů, se kterými si mechanismus sady oprav nedovedl poradit. Původní názvy tohoto projektu, ovlivněné výše uvedeným faktem, byly 386BSD 0.5 nebo 386BSD Interim. Tou dobou totiž 386BSD, operační systém Billa Jolitze, vážně trpěl díky téměř ročnímu zanedbávání. Jak se s přibývajícím dnem sada oprav ukázala být stále nedokonalejší, začali vývojáři postupně nabývat názoru, že by se mělo něco stát, až se rozhodli, že se pokusí Billovi pomoci zmíněnou přechodnou "pročišťovací" verzí. Jejich plány však byly nečekaně zmařeny, když se Bill Jolitz náhle rozhodl celý projekt zastavit, aniž by udal směr dalšího vývoje. Ostatní se rychle rozhodli, že jejich cíl zůstává hodnotným i bez Billovy podpory, a tak si osvojili název "FreeBSD", vymyšlený Davidem Greenmanem.

Své počáteční plány si vytyčili po konzultaci s tehdejšími uživateli systému, a jakmile bylo jasné, že projekt byl na cestě stát se snad dokonce i realitou, kontaktovali společnost Walnut Creek CDROM za účelem vylepšení distribučních kanálů systému FreeBSD k těm řadám nešťastníků, kteří neměli snadný přístup na internet. Společnost Walnut Creek CDROM nejenom že podpořila myšlenku distribuce systému na CD ROM, ale dokonce šla tak

daleko, že poskytla počítač, na kterém mohl tým pracovat, a rychlé připojení k internetu. Bez nebývalé víry společnosti Walnut Creek CDROM v to, co v té době bylo absolutně neznámým projektem, je velmi nepravděpodobné, že by se systém FreeBSD dostal tak daleko a tak rychle tam, kde je dnes.

Přibližně v té době se na obzoru začala shromažďovat neočekávaná bouřková mračna, když společnost Novell a univerzita Berkeley vyřesily svůj dlouhotrvající soudní spor o právní statut Net/2 univerzity Berkeley. Podmínkou soudního urovnání bylo přiznání univerzity Berkeley, že velké části jejího díla Net/2 byly tvořeny pozměněným kódem a majetkem společnosti Novell, která jej před nějakým časem získala od společnosti AT&T. Čeho se univerzitě na oplátku dostalo, bylo udělení "požehnání" společností Novell verzi 4.4BSD-Lite, když byla konečně vydána, jako nezatížená právním dluhem, a prohlášení, že všem současným uživatelům Net/2 je doporučen přechod na novou verzi. To se vztahovalo i na systém FreeBSD a projekt měl od počátku srpna 1994 zastavit distribuci svých produktů založených na Net/2. V rámci podmínek této dohody bylo povoleno před daným termínem vydat poslední verzi, tou bylo FreeBSD 1.1.5.1.

Celý projekt se pak vydal svízelnou cestou svého, v pravém slova smyslu přetvoření na základech poněkud nekompletní verze 4.4BSD-Lite. Verze "Lite" byly skutečně nedokonalé. Částečně proto, že univerzita z nich musela odstranit rozsáhlé části zdrojového kódu vyžadovaného pro vytvoření systému schopného zaváděcího procesu (z důvodu různých právních požadavků), a částečně také z důvodu, že port pro Intel verze 4.4 byl značně nekompletní. Přechod celého projektu trval až do listopadu 1994, kdy došlo k uvolnění nové verze FreeBSD 2.0 na internet a později (v prosinci) také k jeho vydání na CD ROM. Ačkoli nebylo vše ještě úplně propracováno, výsledkem byl významný úspěch, který byl následován ještě větším, po vydání další verze FreeBSD 2.0.5 v červnu 1995.

Verze FreeBSD 2.1.5 byla vydána v srpnu roku 1996 a jak se později ukázalo, získala si mezi členy komerčních a ISP komunit takovou popularitu, že společně s vytvořením větve 2.1-STABLE došlo k vydání další verze. Byla jí verze FreeBSD 2.1.7.1, která byla vydána v únoru roku 1997 a završila konec hlavního vývoje větve 2.1-STABLE. Dále u této větve probíhala pouze vylepšování zabezpečení a odstraňování dalších kritických chyb (RELEASE_2_1_0).

Systém FreeBSD se od hlavní vývojové větve ("CURRENT") oddělil v listopadu 1996 ve formě větve RELENG_2.2 a následná první plná verze byla vydána v dubnu 1997. Další verze větve 2.2 následovaly během léta a podzimu 1997. Poslední verzí byla FreeBSD 2.2.8, která spatřila světlo světa v listopadu 1998. První oficiální verze 3.0 byla vydána v říjnu 1998, což bylo začátkem konce větve 2.2.

20. ledna 1999 došlo ke vzniku nových vývojových větví, 4.0-CURRENT a 3.X- STABLE. Z větve 3.X-STABLE byla dne 15. února 1999 vydána verze 3.1 následována 15. května 1999 verzí 3.2 a 16. září 1999 verzí 3.3. Nejnovější verzí této větve je 3.5.1, která byla vydána pár dní po verzi 3.5 (24. června 2000), opravovala bezpečnostní chybu v autentizačním mechanismu Kerberos.

13. března 2000 se přiblížil vznik nové větve 4.X-STABLE 5.0-CURRENT. Poslední verze, která vyšla ve větvi 4.X je 4.11-RELEASE.

Dlouho očekávaná verze 5.0-RELEASE byla oznámena 19. ledna 2003. Vyvrcholení skoro tří leté práce, tato verze se vydala na cestu zdokonalení multiprocessorů, podporu aplikačních vláken a uvedla podporu pro ultraSPARC a ia64 platformy. Tato větev je stále aktivní poslední verze je 5.4-RELEASE.

Současná verze 6.0-RELEASE vyšla v listopadu 2005. Dlouhodobé vývojové projekty se nadále zabývají vývojovou větví 6.0-RELEASE a průběžně jak práce postupuje vpřed se vydávají na CDROM a na síti.

3.2 Cíle Projektu FreeBSD

Cílem projektu FreeBSD je poskytnout programové vybavení, které bude moci být používáno pro jakýkoli účel bez jakýchkoli případných závazků.

Ke kódu, který je obsažen ve zdrojovém stromu a je chráněn licenčním prohlášením GNU General Public License (GPL) nebo Library General Public License (LGPL), se již závazky vztahují, ačkoli se spíše týkají právoplatného přístupu než obvyklého opaku.

3.3 Vývojový model

Vývoj systému FreeBSD je velmi otevřeným a flexibilním procesem, jelikož je v pravém slova smyslu budován z příspěvků stovek lidí po celém světě.

System CVS

Centrální zdrojový strom systému FreeBSD je udržován systémem CVS (Concurrent Version System - Systém souběžných verzí), volně dostupným řídicím nástrojem pro údržbu zdrojového kódu, který je dodáván současně se systémem FreeBSD. Primární CVS strom je umístěn na počítači v městě Concord CA, USA. Odtud je kopírován na bezpočet zrcadlových počítačů po celém světě. CVS strom, stejně jak jako stromy větví -CURRENT a -STABLE, které jsou jím udržovány, lze snadno zkopírovat na počítač.

Skupina Vývojářů

Vývojáři jsou lidé, kteří mají povolení k zápisu do zdrojového stromu CVS, čímž získávají oprávnění k provádění změn ve zdrojovém kódu FreeBSD. V originále je pro tyto členy týmu používán výraz "committer". Nejlepším způsobem, jak zasílat vývojářům své návrhy k posouzení, je prostřednictvím příkazu *send-pr(1)*. Systém se někdy může zdát zahlcený, a tak se dají vývojáři rovněž kontaktovat prostřednictvím elektronické pošty na adrese jcvs-committers@FreeBSD.org.

Ústřední tým

Pokud by projekt FreeBSD byl společnost, pak by ústřední tým tohoto projektu byl ekvivalentem představenstva. Hlavním úkolem tohoto týmu je zajistit, aby projekt jako celek byl v dobrém stavu a ubíral se správným směrem. Jednou z funkcí tohoto týmu je nabízení podílení se na tvorbě systému FreeBSD obětavým a odpovědným vývojářům, stejně tak jako přijímání nových členů ústředního týmu na místa členů odcházejících. Většina současných členů ústředního týmu kdysi začínala jako vývojáři, jejichž oddanost projektu jim vynesla členství v samém ústředním týmu.

Vnější přispěvatelé

Poslední, nikoli však nevýznamnou, velkou skupinou vývojářů jsou sami uživatelé, kteří nám téměř neustále přispívají zpětnou vazbou a řešením chyb v systému. Základním způsobem, jak se udržovat ve styku s poněkud necentralizovaným vývojem systému FreeBSD, je přihlásit se do technických diskuzních skupin, kde jsou takové záležitosti probírány.

3.4 Současnost

Systém FreeBSD je volně dostupným systémem, založeným na 4.4BSD-Lite2, pro i386, i486, Pentium, Pentium Pro, Celeron, Pentium II, Pentium III, Pentium IV (nebo kompatibilní), Xeon, DEC Alpha a počítače založené na UltraSPARC. Systém je primárně založen na programovém vybavení vytvořeném skupinou CSRG univerzity Berkeley, s některými vylepšeními od NetBSD, OpenBSD, 386BSD a nadace Free Software Foundation.

Od verze FreeBSD 2.0, vydané koncem roku 1994, se výkon, vlastnosti a stabilita systému přímo dramaticky zlepšily. Největší změnou je systém přepracované virtuální paměti se sjednocenou virtuální/souborovou vyrovnávací pamětí, která nejenom zvyšuje výkon, ale rovněž omezuje množství spotřeby paměti systému, čímž vzniká přijatelnější minimální konfigurace o velikosti 5 MB. Mezi další vylepšení patří plná podpora NIS klienta a serveru, podpora TCP transakcí, PPP s vytáčením na požádání, integrovaná podpora DHCP, vylepšený subsystém SCSI, podpora ISDN, podpora síťových adapterů ATM, FDDI, Fast a Gigabit (1000 Mb/s), vylepšená podpora nejnovějších řadičů Adaptec a stovky opravených chyb.

Kromě základního vybavení systém FreeBSD nabízí kolekci programového vybavení s tisíci obvykle vyhledávaných programů. Ve chvíli psaní této bakalářské práce bylo v kolekci portů přesně 14227 programů, od WWW serverů, přes hry, jazyky až po editory a téměř všechno co se do těchto kategorií nedá zařadit. Celá kolekce portů vyžaduje prostor o velikosti přibližně 440 MB, přičemž všechny porty jsou uloženy jako změny svých původních zdrojů, což velmi ulehčuje změny aktualizaci nabídky a výrazně se tím snižují požadavky na volné místo, než tomu bylo u starší verze 1.0. Pokud chceme nějaký port nainstalovat, jednoduše vstoupíme do adresáře ve kterém se program nachází a použijeme příkaz *"make install"* a systém už se postará o

zbytek, stáhne zdrojové kódy, rozbálí je, zkompile a nainstaluje. Hodně programů je také nabízeno ve formě předem kompilovaného programového balíku, takže pokud necheme kompilovat přímo ze zdrojového kódu, můžeme instalovat jednoduchým příkazem *"pkg_add"*.

Velké množství dodatečných užitečných dokumentů se nachází v adresáři */usr/share/doc/*. Rovněž se můžeme prostřednictvím jakéhokoli prohlížečího programu, schopného zobrazovat HTML, podívat na lokálně instalované manuály na následujícím URL:

file:/usr/share/doc/

Kromě toho si také můžeme prohlédnout hlavní (a nejčastěji aktualizované) kopie na stránce <http://www.freebsd.org>

Chapter 4

Porovnání Linuxu a FreeBSD

FreeBSD je "unix like" operační systém založený na Berkeley Software Distribution. Přestože, Linux je také "unix like" operační systém, liší se v mnoha věcech. Uvedu jen pár nejdůležitějších.

FreeBSD je registrovaná obchodní značka FreeBSD Foundation.
Linux je registrovaná obchodní značka Linuse Torvaldse.
Unix je registrovaná obchodní značka sdružení The Open Group

4.1 System

4.1.1 Linux

Linux je jenom kernel. Distribuce (Red Hat, Debian, Suse atd.) uživatelům poskytují instalátor a potřebné nástroje. Na adrese <http://www.linux.org/dist> je přes 370 linuxových distribucí. Zatímco uživatelům dává maximální flexibilitu, existence tolika distribucí jen zvyšuje obtížnost přenést jednu schopnost z jedné na druhou. Distribuce se neliší jen instalací a dostupnými programy, ale také adresářovou strukturou, instalováním programů a postupem patchování.

4.1.2 FreeBSD

FreeBSD je jenom jedno. FreeBSD je kompletní operační systém (jádro a systém) s respektovaným dědictvím zakotveném v kořenech vývoje Unixu.

4.2 licence

4.2.1 BSD licence

Licence, pod kterou byl operační systém BSD šířen, v krátkosti povoluje dělat se zdrojovým kódem cokoli (včetně použití v closed-source projektu). Jediné, co vyžaduje, je to, aby při vychvalování těch vlastností výsledného produktu, které souvisí s přejatým kódem, bylo explicitně řečeno, že kód je založen na práci originálního autora. Licence obsahuje tyto čtyři body:

1. Redistribuce ve formě zdrojového kódu musí zachovat informaci o autorských právech, seznam podmínek používání a poznámku o tom, že autor není zodpovědný za případné problémy vzniklé používáním jeho produktu.
2. Redistribuce v binární podobě musí uvést informaci o autorských právech, podmínky používání a onu poznámku v dokumentaci a/nebo v jiných materiálech přiložených k distribuci.
3. Všechny reklamní materiály hovořící o vlastnostech nebo použití tohoto software musí obsahovat následující sdělení:

Tento produkt obsahuje software vyvinutý na Kalifornské univerzitě v Berkeley a jejími přispěvateli.

4. Ani jméno Kalifornské univerzity ani jméno přispěvatelů nesmí být použito ke schvalování či prosazování produktů založených na tomto software bez specifického, předem daného písemného povolení.

FreeBSD licence obsahuje pouze první 2 body.

4.2.2 GPL licence

General Public Licence (GPL) je podstatně delší a složitější, takže jí sem nebudu přepisovat jako tomu bylo u BSD licence. GPL licence říká, že uživatel programu musí mít stejná práva jako jeho autor. Programátor napíše nějaký program a ať ho zdarma distribuuje pomocí internetu nebo ho

někomu prodá na CDROM, vždy pokud uživatel chce, musí dodat i zdrojové kódy maximálně za cenu nákladu na dopravu k příjemci.

Pokud je v nějakém projektu i jenom malá část kódu převzatá z kódu licencovaného pod GPL musí být i celý projekt licencován pod GPL.

LGPL je varianta GPL, která se používá pro knihovny. Pokud je GPL kód v samostatné knihovně licencované pod LGPL, náš program může být pod libovolnou licencí.

4.3 Souborové systémy

4.3.1 BSD filesystemy

V operačním systému FreeBSD existuje jen jediný souborový systém, který je považovaný za jeho nativní, a je jako jediný doporučený na používání pod tímto operačním systémem. Tento souborový systém se nazývá *Berkeley Fast File System (FFS)*, častěji se ale označuje zkratkou UFS, která znamená *Unix File System*, protože se používá na všech unixových systémech, které nemají vlastní souborový systém.

S příchodem větve FreeBSD 5 se začala používat nová verze souborového systému označovaná jako UFS2. Byla vyvinuta, aby odstranila, některá omezení původního souborového systému. Jednou z častých námitek vůči souborovému systému UFS je, že počet inodů je dopředu daný při vytváření souborového systému. Tím je dopředu omezený maximální počet souborů, které se dají na takový systém uložit. Opačnou nevýhodou předalokovaného pevného počtu inodů je plýtvání místem, když je uzlů zbytečně moc. UFS2 proto přichází s úpravou, která umožňuje inody podle potřeby vytvářet a nepotřebné naopak rušit. Na souborovém systému UFS2 je proto vždy dostatek inodů, ale zároveň tam nejsou žádné nepotřebné inody, které by jen zbytečně zabírali místo.

UFS používá pro zabezpečení konzistence dat souborového systému metodu, která se nazývá *SoftUpdates*. Operační systém dohlíží nad používáním vyrovnávací paměti zápisu takovým způsobem, aby byly dodrženy závislosti mezi údaji, které se zapisují. Pořadí zápisu údajů na souborový systém je potom takový, že kdykoliv dojde k výpadku napájení, data zůstanou konzistentní.

Další zlepšení UFS2 se týká posunutí některých jeho limitů. Rozsahy datových struktur byly zvětšeny tak, aby souborový systém dokázal sledovat současné tempo zvyšování kapacit pevných disků a velikosti ukládaných souborů. Byla zvětšena i datová struktura, do které se ukládají časové známky vytvoření, modifikace a přístupu k souboru, takže "v dohledné době" 136 bilionů let nehrozí přetečení údajů.

Nejvíce diskutované jsou možnosti UFS2 vytvářet tzv. snapshoty souborového systému, to je jakýsi snímek stavu souborového systému v aktuálním čase, a možnost provádět kontrolu integrity dat na souborovém systému, která se vykoná právě na vykonaném snapshotu. Toto umožní dělat kontrolu integrity dat během běžného používání.

4.3.2 Linuxové souborové systémy

Linux má na výběr trochu více souborových systémů: *ext3*, *reiserFS*, *JFS* a *XFS*, všechny patří mezi tzv. žurnálovací souborové systémy. Žurnálovací filesystem si uchovává informace o operacích, které provedl a je pak v případě výpadku schopen rychle se dostat zpět do konzistentního stavu. Změny jsou evidovány jako tzv. transakce. Jedná se o nezávislé atomické operace. Po každé transakci následuje potvrzení, když dojde k uskutečnění daného úkonu (např. zápis na disk). Proto pokud systém "spadne", můžeme najít v záznamech informace o provedených změnách a vrátit vše do původního stavu.

ext2/ext3

V Linuxu byl dlouhou dobu standardem souborový systém *ext2*, jeho první verze vyšla v roce 1993 a stále je dostatečně široce podporován. Tento filesystem ještě není žurnálovací, ale souborový systém *ext3*, který vznikl v roce 1999 už žurnálovací je a navíc je s *ext2* oboustranně kompatibilní. Pokud chceme z *ext2* udělat *ext3* stačí jenom vytvořit žurnál, žádná záloha dat a formátování není potřeba. A to je jeden z mnoha důvodů proč je *ext3* nejrozšířenější souborový systém na linuxu.

JFS

JFS byl vytvořen firmou IBM původně pro AIX, záhy byl portován na OS/2, poté zpět na AIX a následně na Linux. JFS byl vytvořen s důrazem na vysokou spolehlivost a rychlost. Postupem času uvolněný pod open source

licencí.

reiserFS

ReiserFs (v3) byl představen v roce 2001 Hansem Reiserem z týmu Namesys a jeho výhodou je excelentní výkon při práci s malými nebo mnoha soubory (při použití velkých souborů je jeho výkon průměrný až podprůměrný). V dřívějších dobách obsahoval pár nepříjemných chyb a jeho programy nebyly příliš spolehlivé, ovšem současný stav by už měl být plně srovnatelný s ostatními filesystemy.

XFS

XFS je filesystem s žurnálem, který vyvinula (představen byl v roce 1993) SGI pro svůj komerční IRIX. V květnu 2000 byl XFS uvolněn pod open source licenci.

Chapter 5

FreeBSD a praxe

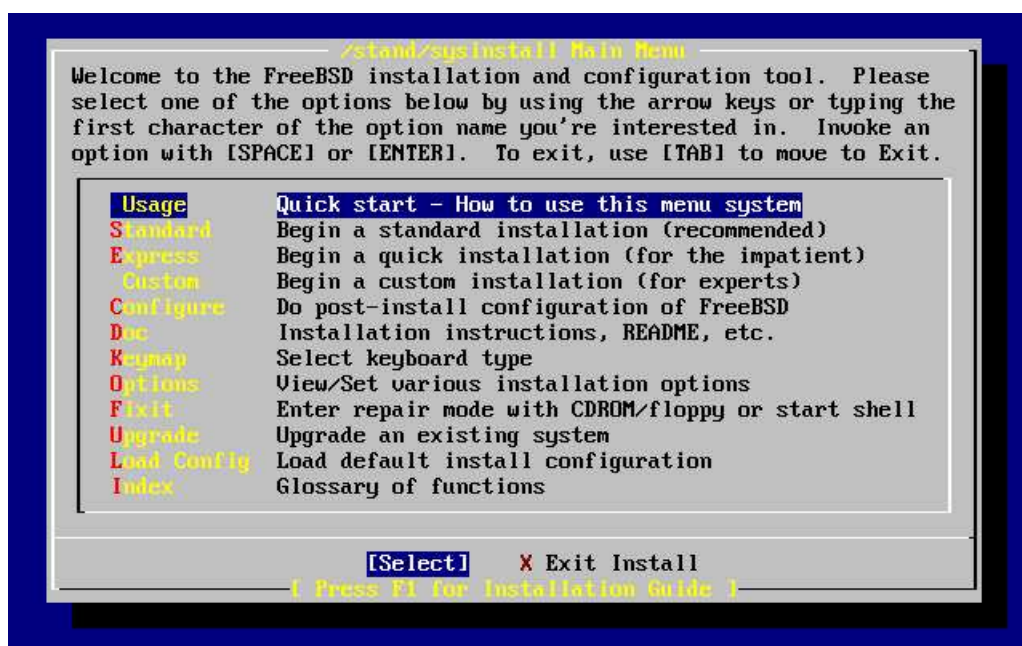
5.1 instalace

5.1.1 výběr instalace

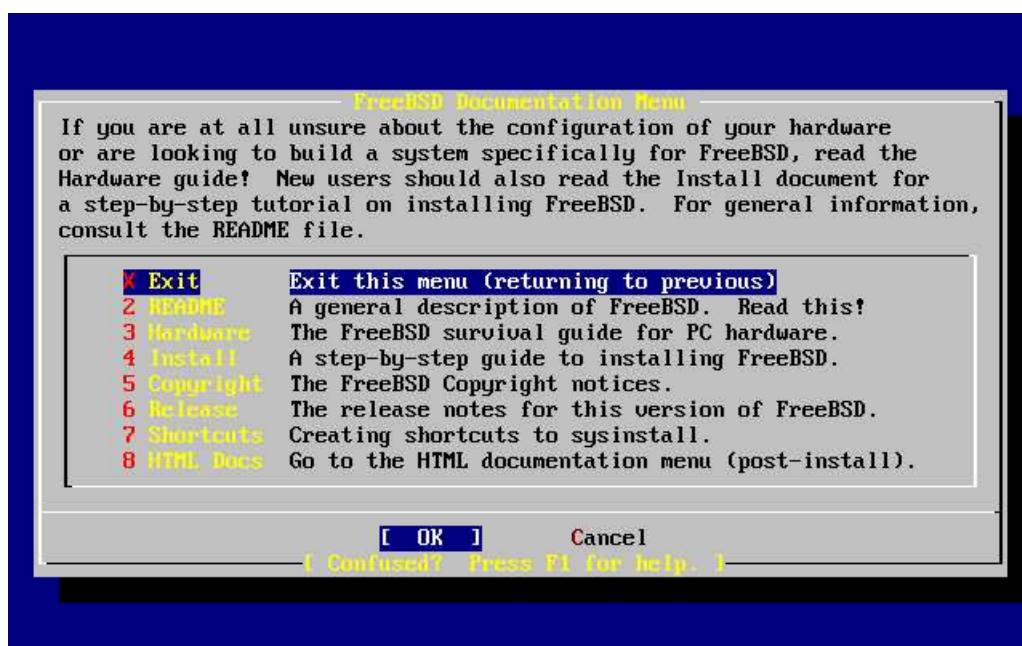
Je mnoho způsobů, jak nainstalovat FreeBSD: z CD-ROM nebo DVD, z diskety, z kazety, z anonymního FTP a z NFS. Pokud například chceme instalovat z CD, musíme si nejdříve stáhnout ISO z některého FTP mirroru nebo koupit od některého distributora. Na FTP se nachází 3 ISO soubory, první které má v názvu bootonly je pro síťovou instalaci. Pro instalaci z CD potřebujeme ISO soubor disc1. Disc2 není pro instalaci důležitý, nacházejí se na něm pouze balíčky dalších programů, druhý disk využijí ti co nemají rychlé připojení k internetu.

5.1.2 Instalace z CD-ROM

CDčko je bootovatelné, takže se po startu PC vloží do mechaniky a nastaví se z ní bootování. Spustí se BTX loader, po kterém se objeví jednoduché menu s výběrem kernelu do kterého chceme nabootovat, například je tam možnost zákázat ACPI nebo nabootovat do "Safe Modu". Po nabootování se spustí instalační utilita "sysinstall" a zobrazí se hlavní menu:



Sysinstall je konzolová utilita pro konfiguraci a instalaci systému, kterou můžeme pustit i kdykoliv po instalaci. Menu se ovládá pomocí kurzorových šipek, enteru, mezerníku a dalších kláves, které jsou popsány v nápovědě. Pokud si nejsme jistí, vybereme nejdříve položku "Doc", kde si můžeme zjistit, zda je náš hardware podporován nebo si přečíst o celém instalačním procesu:



Pro změnu klávesnice použijeme položku "Keymap", to je potřeba pouze pokud máme nestandardní klávesnici. Pro další změnu dalších nastavení instalace použijeme položku "Options":



Až budem připraveni přistoupíme k vlastní instalaci. Pro nové uživatele je doporučena standardní instalace. U expresní instalace nejsou mezi jednotlivými nastaveními instalace upřesňující informace pro nové uživatele a volitelná (Custom) instalace umožňuje některé kroky přeskočit.

5.1.3 Číslování disků systémem BIOS

Než si nainstalujeme a provedeme konfiguraci operačního systému FreeBSD, měli bychom si, obzvláště pokud máme více pevných disků, být vědomi jedné věci.

U počítačů, jež používají systém DOS, nebo jiné, na systému BIOS závislých operačních systému (WINxxx), je BIOS schopen pozměnit obvyklé pořadí disků, čemuž se příslušný operační systém přizpůsobí. Uživateli je tak umožněno zavádět z jiného disku, než z tzv. "hlavního primárního disku". To je obzvláště výhodné pro uživatele, kteří zjistili, že nejjednodušším a nejlépeším způsobem, jak systém zálohovat, je zakoupit identický sekundární pevný disk a pomocí aplikace Ghost nebo xcopy, přenést kopie programů z primárního disku na sekundární. Pokud potom dojde k poškození primárního disku, jeho napadení virem, nebo jinému poškození, vzniklému chybou operačního systému, lze vše snadno opravit tím, že jednoduše systému BIOS zadáte, aby logicky disky vyměnil. Je to podobné, jako kdybychom přehodili kabel z jednoho disku na druhý, ale nemusíme při tom samozřejmě otevírat kryt počítače.

Dražší systémy s řadiči SCSI nezřídka obsahují rozšířenou verzi systému BIOS, který umožňuje podobným způsobem upravit nastavení SCSI disků až pro sedm disků.

Uživatelé, kteří jsou zvyklí této vlastnosti využívat, mohou být překvapeni, pokud výsledky v případě systému FreeBSD nejsou takové, jaké očekávali. Operační systém FreeBSD totiž se systémem BIOS nespolečně pracuje a nezná "logické mapování disků systémem BIOS". To může vést ke zmatečným situacím, obzvláště v případě, že geometrie disků jsou fyzicky identické, případně byl disk vytvořen jako přesná kopie jiného disku.

Pokud chceme používat systém FreeBSD, vždy před jeho zavedením obnovíme přirozené číslování disků systému BIOS a ponecháme tak. Pokud

potřebujeme disky přehodit, nic nám v tom nebrání. Nepřehodíme však číslování , nýbrž počítač otevřem a přehodíme propojky a kabely.

5.1.4 Alokace místa na disku

Ve FreeBSD se zařízení pojmenovávají jinak, než je tomu v Linuxu. Síťová karta se v linuxu vždy jmenuje "eth0", kdežto ve FreeBSD se může jmenovat "vr0", "rl0" nebo úplně jinak. FreeBSD totiž dává jména zařízením podle názvu ovladače plus číslo pořadí. Takže síťová karta Realtek se jmenuje rl0, druhá síťová karta by se jmenovala rl1 a síťová karta via-rhine vr0 atd. Jiné je to i u pevných disků, v Linuxu se IDE disky jmenují "hd" plus písmeno určující pořadí (hda,hdb,...), ve FreeBSD je to "ad" plus číslo pořadí (ad0,ad1,...).

Nejprve musíme programem fdisk vytvořit oddíly pro operační systém FreeBSD. V následující tabulce je seznam všech pevných disku počítače a vybereme ten na který chceme nainstalovat FreeBSD.

Obrazovka Fdisku je rozdělena do tří částí. První část ukazuje informace o vybraném disku, včetně jména disku, jeho geometrii a celkovou velikost. Druhá část ukazuje informace o jednotlivých oddílech, kde začínají a končí, jejich velikost, popis, podtyp a jméno ve FreeBSD. Na následujícím obrázku jsou vidět 2 malé nepoužité oddíly na kterých jsou uloženy informace o oddílech. Také ukazuje velký FAT oddíl, který by se v dosu nebo windows pravděpodobně jmenoval C: a jeden rozšířený oddíl. Třetí část ukazuje dostupné příkazy pro fdisk.

```

Disk name:      ad0      FDISK Partition Editor
DISK Geometry: 16383 cyls/16 heads/63 sectors = 16514064 sectors (8063MB)

Offset      Size(ST)      End      Name  PType      Desc  Subtype      Flags
-----
0           63           62      -      6      unused      0
63          4193217      4193279  ad0s1  2      fat         14      >
4193280     1008          4194287  -      6      unused      0      >
4194288     12319776      16514063  ad0s2  4      extended    15      >

The following commands are supported (in upper or lower case):

A = Use Entire Disk      G = set Drive Geometry    C = Create Slice      F = `DD' mode
D = Delete Slice         Z = Toggle Size Units    S = Set Bootable      I = Wizard m.
T = Change Type          U = Undo All Changes     Q = Finish

Use F1 or ? to get more help, arrow keys to select.

```

Pokud chceme pro FreeBSD použít celý disk zmáčkneme **A** což znamená příkaz "Use entire disk". Všechny existující oddíly budou smazány a přepsány malým oddílem označeným jako unused (nepoužitý) pro uložení informací o oddílech a jedním velkým oddílem pro FreeBSD. Potom šipkami vybereme oddíl určený pro FreeBSD a klávesou **S** ho označíme jako bootovací.

Pokud chceme smazat existující oddíl a uvolnit místo pro FreeBSD, kurzorovými šipkami ho vybereme a klávesou **D** smažeme. Klávesou **C** potom vytvoříme nový oddíl, při tom budeme dotázáni na jeho velikost. Implicitní hodnota je maximální možná velikost aktuálního volného místa na disku.

Když jsme hotový stiskneme **Q**. Změny budou uloženy v paměti ale ještě nebudou zapsány na disk.

Dále máme možnost nainstalovat Boot Manager. Této možnosti využijeme pokud FreeBSD instalujeme na jiný disk než na první nebo pokud máme v počítači více operačních systémů, které chceme střídat. Pokud je FreeBSD jediný operační systém na počítači vybereme možnost standardní bootovací manažer. A pokud používáme bootovací manažer třetí strany schopný bootování FreeBSD použijeme možnost none.

Jelikož operační systém FreeBSD lze použít na počítačích kompatibilních s počítači IBM-PC, musíme vzít v potaz diskové oddíly BIOS. Tyto oddíly se liší od tradičních oddílů BSD. Pevný disk může mít až čtyři diskové oddíly BIOS. Pokud nebude systém FreeBSD nový disk sdílet s žádnými jinými operačními systémy, můžeme je využít ve vyhrazeném režimu. V opačném případě mu bude muset postačit jeden z diskových oddílů BIOS. Systém FreeBSD označuje tyto oddíly jako části (slices), aby nedošlo k jejich případné záměně s tradičními BSD oddíly. Zmiňované části můžeme vytvořit i na discích, které jsou systému FreeBSD vyhrazeny, ale jsou použité i v počítači s jiným operačním systémem. To proto, aby nedošlo k popletení utility fdisk jiného operačního systému.

Příklad využití diskové části: `/dev/da0s1e`. Uvedený zápis lze přečíst jako: SCSI disk, jednotka číslo 0 (první SCSI disk), část 1 (diskový oddíl BIOS 1) a BSD oddíl e. V případě vyhrazeného režimu by byl disk jednoduše přidán jako `/dev/da1e`.

Dále tedy musíme vytvořit tradiční BSD oddíly, k tomu se teď dostaneme do editoru oddílů (DiskLabel editor). Editor oddílů je rozdělen do tří sekcí.

První řádek ukazuje jméno disku, na kterém právě pracujeme a část kde budeme vytvářet oddíly. A také ukazuje volné místo diskové části.

Prostřední část zobrazuje vytvořené oddíly s jejich jménem, jejich velikost a další nastavení.

V dolní části jsou klávesy pro ovládání editoru oddílů.

Disklabel může oddíly vytvořit automaticky s implicitní velikostí. Pro automatické vytvoření oddílů stiskneme **A** a uvidíme něco podobného jako na následujícím obrázku:

```

FreeBSD Disklabel Editor

Disk: ad0      Partition name: ad0s1  Free: 0 blocks (0MB)

Part      Mount      Size Newfs  Part      Mount      Size Newfs
-----
ad0s1a    /                128MB UFS   Y
ad0s1b    swap             503MB SWAP
ad0s1e    /var             256MB UFS+S Y
ad0s1f    /tmp             256MB UFS+S Y
ad0s1g    /usr             6919MB UFS+S Y

The following commands are valid here (upper or lower case):
C = Create      D = Delete  M = Mount pt.  W = Write
N = Newfs Opts  Q = Finish   S = Toggle SoftUpdates
T = Toggle Newfs U = Undo    A = Auto Defaults  R = Delete+Merge

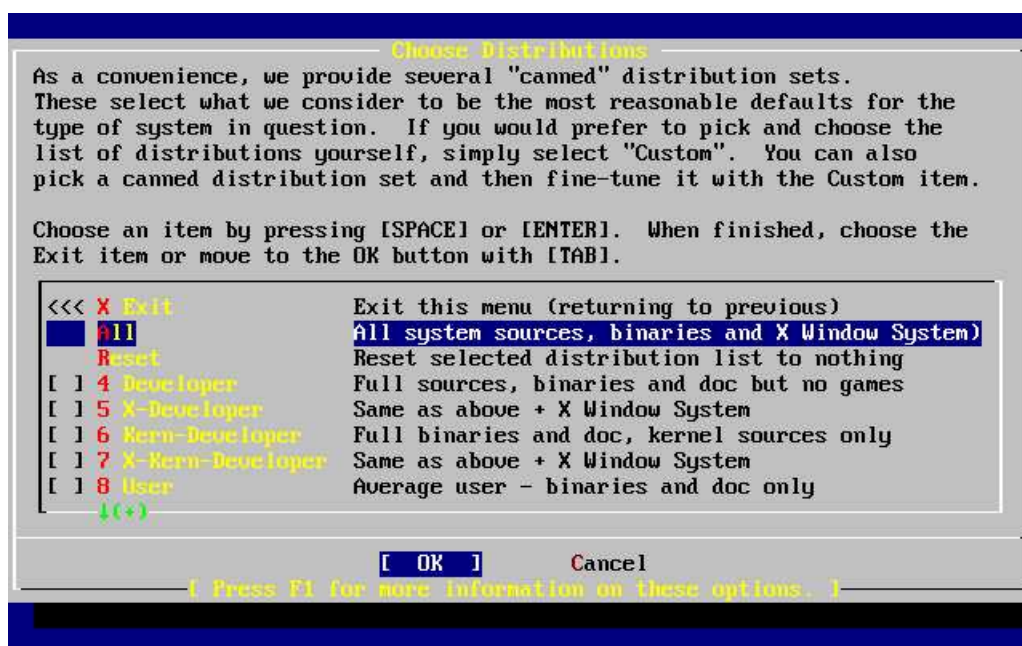
Use F1 or ? to get more help, arrow keys to select.

```

Pokud nám implicitní nastavení vyhovuje akceptujeme ho klávesou **Q** v opačném případě oddíl smažeme (**D**) a vytvoříme je ručně. Oddíl vytvoříme klávesou **C**, první oddíl musí být připojen jako **/** – root. Dialogové okno se nás zeptá na velikost oddílu. Velikost můžeme zadat v blocích, megabytech, gigabytech, nebo cylindrech. Pak se nás dialog zeptá zda oddíl bude filesys-tem (souborový systém) nebo swap (odkládací oddíl). Swap by měl mít dvoj až trojnásobnou velikost fyzické paměti RAM. U filesystému musíme určit přípojný bod, hlavní oddíl bude připojen do **/**. Další oddíly budou připojeny obvykle do **/var**, **/usr** případně jinam, záleží jaké máme požadavky.

5.1.5 Výběr balíčků

Rozhodnutí jaký distribuční balík nainstalujeme závisí na tom jak chceme systém používat a na velikosti disku. Z přednastavených možností můžeme nainstalovat minimální systém nebo úplně všechno. Pro zkušenější uživatele je i možnost ručně vybrat balíčky, které se mají nainstalovat. Pokud si vybereme instalaci grafického uživatelského prostředí od verze FreeBSD 5.3 se instaluje jako implicitní X server Xorg.



Po vybrání některé distribuce se nás instalátor zeptá, zda chceme nainstalovat také kolekci portů. Kolekce portů je snadný způsob jak nainstalovat další software. Kolekce neobsahuje zdrojové kódy programů, ale soubory s informacemi jak zdrojové kódy automaticky stáhnout, zkompileovat a nainstalovat. Instalační program nekontroluje místo na disku, proto kolekci portů nainstalujeme jen pokud víme, že máme dostatek místa na disku. Pro FreeBSD 6.0 je to přibližně 440 MB.

5.1.6 Výběr instalačního media

Teď se rozhodneme odkud chceme FreeBSD nainstalovat, máme na výběr několik možností, instalaci z CD/DVD nebo nejnovější verze balíčků se sítě. Když vybereme instalaci například z FTP, musíme nastavit připojení k internetu. Instalátor nám nabídce, přes které zařízení chceme instalovat, můžeme jich mít víc (modem, síťová karta atd.).

Bohužel, já osobně mám s instalací z internetu špatné zkušenosti. U FreeBSD 6.0 ani u starší verze se mi nepodařilo instalaci rozjet. Přesto, že po instalaci z CD jsem se sítí, ani internetem neměl nikdy žádné problémy a vše bezvadně fungovalo. Obvykle se instalace zastavila u hlášky "Resolving host".

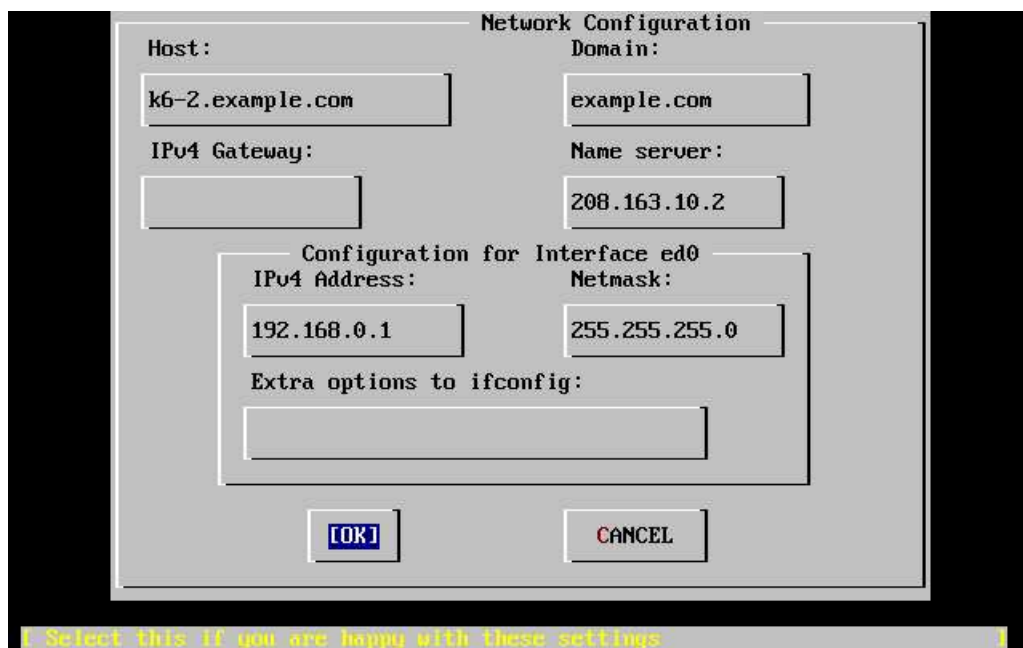
5.1.7 Zahájení instalace

Po vybrání instalačního media se nás instalátor zeptá, zda si jsme jistí a opravdu chceme začít instalovat. Ještě teď je možnost instalaci ukončit bez žádných následků, protože do této chvíle instalátor na pevný disk nezačal instalovat a ani fdiskem nezměnil jeho oddíly. Pokud proběhla instalace v pořádku instalátor zahlásí:

Congratulations! You now have FreeBSD installed on your system.

5.1.8 Po instalační nastavení

Než poprvé nabootujeme do svého FreeBSD, musíme ještě provést základní konfiguraci. Instalátor se nás postupně ptá zda chceme nastavit připojení k internetu, jestli se chceme k počítači vzdáleně připojovat přes ssh, jestli chceme PC používat jako NFS server nebo klient, nastavení hesla roota, přidání dalších uživatelů atd. Je to jednoduché, většinou se jenom odpovídá ano nebo ne, případně je nám nabídnut nějaký konfigurační soubor k editaci nebo vyplnění podobného formuláře:



The screenshot shows a 'Network Configuration' window with the following fields and values:

Host:	Domain:
k6-2.example.com	example.com
IPv4 Gateway:	Name server:
	208.163.10.2
Configuration for Interface ed0	
IPv4 Address:	Netmask:
192.168.0.1	255.255.255.0
Extra options to ifconfig:	

At the bottom are 'OK' and 'CANCEL' buttons. A yellow status bar at the very bottom reads: 'Select this if you are happy with these settings'.

5.2 Instalace aplikací: balíčky a porty

FreeBSD obsahuje bohatou kolekci systémových nástrojů jako součást základního systému. Nicméně jsou to jen nástroje, nutné pro instalaci dalšího software, který budeme potřebovat pro svou práci. FreeBSD poskytuje dvě kompletní technologie pro instalaci software: Kolekce portů (pro instalaci ze zdrojových kódů) a balíčky (instalace binárních souborů). Jeden i druhý způsob může být použit pro instalaci nejnovějších verzí programů z lokálního media nebo z internetu. Kolekce portů nejsou seriové porty na zadní straně počítače, port je odvozen od slova převádění (porting) programu z jedné verze systému Unix do jiné.

Výhody balíčků:

1. Zkomprimované balíčky jsou obvykle menší než zkomprimované zdrojové kódy.
2. Balíčky se nemusí kompilovat. Instalace je podstatně rychlejší než v případě kolekce portů.
3. Balíčky nepotřebují rozumět procesu kompilace ve FreeBSD.

Výhody portů:

1. Balíčky jsou kompilované s tradičním nastavením, protože se použijí na velkém počtu systémů. Instalováním z portů můžeme generovat kód, který je specifický pro procesory Pentium IV nebo Athlon (na příklad).
2. Některé aplikace mohou být kompilovány s podporou, kterou chceme nebo nechceme. Na příklad Apache může být kompilován s podporou wide. Instalací z portů můžeme změnit implicitní nastavení a podporu pro wide zakázat.
3. Licence některých programů zakazuje distribuci v binární formě.
4. Někteří lidé nevěří binárním distribucím. Když mají zdrojové kódy mohou se podívat na potenciální problém.
5. Pokud máme lokální patche, potřebujeme zdrojové kódy, abychom je mohli aplikovat.

5.2.1 Hledání programů

Předtím než můžeme nainstalovat nějaký program, potřebujeme vědět jak se jmenuje. Seznam programů dostupných pro FreeBSD neustále roste. Naštěstí

je zde několik cest jak najít ten který hledáme.

a) Na adrese <http://www.freebsd.org/ports> je aktuální seznam všech dostupných aplikací pro FreeBSD. Programy můžeme vyhledávat podle jména nebo podle kategorií.

b) Web <http://www.freshports.org> sleduje změny ve stromu portů.

c) Když nevíme jméno aplikace, kterou chceme, můžeme zkusit <http://www.freshmeat.net> pro nalezení, a potom zkontrolovat na stránkách FreeBSD jestli už byla aplikace portovaná.

d) Pokud víme přesné jméno portu a chceme zjistit v jaké je kategorii, můžeme použít příkaz "whereis soubor" kde soubor je jméno programu. Příklad:

```
whereis wesnoth
/usr/ports/games/wesnoth
```

Ale pozor, takhle příkaz whereis vrátí pouze první nalezený soubor. Což je například u mého oblíbeného terminálu rxvt docela nepříjemné, protože řekne, kde se nachází upravená verze pro japonštinu, ve které nefungují české znaky. Doporučuji používat jinou výše popsanou metodu nebo používat whereis s přepínačem -a:

```
whereis -a rxvt
/usr/ports/japanese/rxvt /usr/ports/x11/rxvt
```

e) Další z možností jak najít nějaký port je použít zabudovaný mechanismus kolekce portů. K tomu potřebujeme být v adresáři /usr/ports/ a použít "make search name=program" , kde program je jméno hledané aplikace. Příklad:

```
make search name=wesnoth
Port: wesnoth-1.0.2
Path: /usr/ports/games/wesnoth
Info: A fantasy turn-based strategy game
Maint: mezz@FreeBSD.org
WWW: http://www.wesnoth.org/
```

Pro hledání více do hloubky můžeme použít "make seach key=string", kde string je nějaké hledané slovo. Takto se nehledá jenom v názvu programech,

ale také v komentáři a popisu. Hledání do hloubky můžeme použít když víme co má program dělat, ale nevíme jak se jmenuje. V obou případech hledání nerozlišuje malá a velká písmena.

5.2.2 Používání balíčkovacího systému

Pro instalaci balíčků můžeme použít utilitu *pkg_add(1)*, jak pro lokální balíček, tak pro instalaci ze sítě. Balíček nejprve stáhneme a potom příkazem *pkg_add jmeno-verze.tgz* nainstalujeme:

```
pkg_add lsof-4.56.4.tgz
```

Pokud nemáme lokální zdroj balíčků (FreeBSD CD-ROM) bude jednodušší použít přepínač *-r*. Ten zajistí, že se balíček automaticky stáhne ze sítě a nainstaluje:

```
pkg_add -r lsof
```

Tento příklad stáhne správný balíček ze sítě a nainstaluje ho bez žádného dalšího uživatelského zásahu. Pokud chceme nastavit alternativní mirror balíčku, místo hlavního mirroru FreeBSD nastavíme proměnnou *PACKAGESITE* k překrytí implicitního nastavení. *pkg_add(1)* používá pro stahování souborů *fetch(3)*, který ctí různé proměnné prostředí, včetně *FTP_PASSIVE_MODE*, *FTP_PROXY* a *FTP_PASSWORD*. Tyto proměnné potřebujeme nastavit, pokud jsme za firewallem nebo používáme HTTP/FTP proxy. V manuálových stránkách *fetch(3)* je seznam všech proměnných. Ještě malá poznámka k tomu proč bylo v předchozím příkladu použito *lsof* místo *lsof-4.56.4.tgz*. Při instalaci ze sítě se číslo verze a celý název souboru nepoužívá. *pkg_add* automaticky stáhne nejnovější verzi balíčku.

Balíčky jsou distribuovány ve formátu *.tgz* nebo *.tbz*. Ty můžeme nalézt na adrese <ftp://ftp.FreeBSD.org/pub/FreeBSD/ports/packages/> nebo na CD FreeBSD. Třídění balíčků je podobné jako v */usr/ports*, balíčky jsou rozdělené do kategorií a taky lze nalézt všechny v adresáři *All*.

pkg_info je utilita pro vypsání všech instalovaných balíčků a jejich popis.

```

pkg_info
ImageMagick-6.2.5.5_3 Image processing tools
autoconf-2.13.000227_5 Automatically configure source code on
many Un*x platforms
...

```

pkg_version je utilita pro shrnutí všech verzí nainstalovaných balíčků. Ty budou porovnány s aktuálně nalezenými balíky v kolekci portů.

```

pkg_version
ImageMagick          =
autoconf              =
...

```

Symbols ve druhém sloupci ukazují relativní staří instalovaných verzí s verzemi nalezenými v lokálním stromu portů.

= – Instalovaná verze je shodná s verzí v lokálním stromu portů.

i – Instalovaná verze je starší než verze v lokálním stromu portů.

! – Instalovaná verze je novější než verze v lokálním stromu portů (pravděpodobně zastaralá kolekce portů)

? – Port se nepodařilo nalézt v kolekci portů (pravděpodobně byl z kolekce odstraněn nebo přejmenován).

* – Několik verzí balíčku.

Pokud například chceme zjistit, které programy jsou zastaralé a je možné je aktualizovat stačí napsat příkaz:

```
pkg_version | grep \ <
```

Pro smazání nainstalovaného balíčku použijeme utilitu *pkg_delete(1)*.

```
pkg_delete xchat-2.0.4
```

Všechny informace o balíčcích jsou uloženy v adresáři `/var/db/pkg/`. Kde můžeme najít seznam všech instalovaných souborů balíčku a jeho popis.

5.2.3 Kolekce portů

Než můžeme instalovat z portů, musíme nejprve získat kolekci portů, která obsahuje Makefiles, patche a popis souborů v /usr/ports.

Když jsme instalovali operační systém FreeBSD, instalátor se nás ptal, jestli chceme kolekci portů nainstalovat. Pokud jsme zvolili ne, máme teď 2 způsoby jak kolekci získat.

Metoda CVSup

Toto je rychlá metoda jak získat aktuální kopii kolekce portů. Nejprve nainstalujeme balík net/cvsup-without-gui

```
pkg_add -r cvsup-without-gui
```

a potom CVSup pustíme:

```
cvsup -L 2 -h cvsup.FreeBSD.org  
/usr/share/examples/cvsup/ports-supfile
```

cvsup.FreeBSD.org vyměníme za server, který je k nám blíž například cv-sup.cz.FreeBSD.org. Pokud chceme používat vlastní soubor ports-supfile, překopírujeme ho nejlépe do domovského adresáře roota. Editujeme, třeba změnímá implicitní server a potom CVSup můžeme spouštět:

```
cvsup -L 2 /root/ports-supfile
```

Když CVSup pustíme později, tak automaticky stáhne a aplikuje poslední změny v kolekci portů.

Metoda Portsnap

Portsnap(8) je alternativní systém pro distribuci kolekce portů. Od verze FreeBSD 6.0 je součástí základního systému, ve starších verzích se musí ručně doinstalovat z sysutils/portsnap. Pokud adresář /usr/ports neexistuje tak ho musíme vytvořit:

```
mkdir /usr/ports
```

stáhneme aktuální komprimovaný snapshot kolekce portů:

```
portsnap fetch
```

snapshot rozbálíme:

portsnap extract

a máme kolekci portů nainstalovanou. Aktualizace změn ve stromu se provádí:

portsnap update

Metoda Sysinstall

Kolekci portů můžeme instalovat i pomocí sysinstall. Jako root spustíme /stand/sysinstall a v menu zvolíme instalaci kolekce portů. Takto ale bude nainstalovaná starší verze stromu z instalačního media, pokud máme přístup k internetu je lepší používat jednu z výše uvedených metod.

Instalce portů

Nejprve si vysvětlíme co je v kolekci portů skeleton. Skeleton je minimální sada souborů, v které jsou informace jak program automaticky stáhnout, zkompileovat a nainstalovat. Každý skeleton obsahuje:

Makefile - Soubor obsahující různé příkazy specifikující jak bude aplikace kompilována a kde v systému bude nainstalována.

distinfo - Tento soubor obsahuje informace o souborech, které musí být staženy pro sestavení portu a kontrolní součty pro ověření, zda nebyly soubory poškozeny nebo pozměněny během stahování. K tomu se používá *md5(1)*.

Files - Tento adresář obsahuje patche potřebné pro kompilaci a instalaci. Patche jsou obvykle malé soubory specifikující podrobně změny v souborech. Patche ve formátu prostého textu a obvykle říkají: "Smaž řádek 10" nebo "Zaměň na řádku 26 ...". Patche jsou také známe jako diffy, protože jsou generovány programem *diff(1)*.

pkg-descr - Tento soubor obsahuje obvykle více řádkový popis programu.

pkg-plist - Soubor obsahuje seznam všech souborů, které bude port instalovat. Také systému říká, které soubory mají být odstraněny při odinstalování programu.

Některé porty obsahují ještě více souborů, jako třeba pkg-message. Systém je používá pro speciální situace. Například ve zmíněném souboru jsou informace, které se uživateli zobrazí po nainstalování portu.

Systém kolekce portů předpokládá, že máme funkční připojení k internetu. Pokud ne musíme zdrojové kódy umístit do adresáře `/usr/ports/distfiles/`

Nejprve se musíme přesunout do adresáře portu, který chceme nainstalovat.

```
cd /usr/ports/games/wesnoth
```

V tomto adresáři můžeme vidět skeleton portu. Další krok je zkompileovat nebo "sestavit" port. To se jednoduše udělá příkazem `make`:

```
make
```

Až bude kompilace hotová, u velkých programů, může trvat i několik hodin, vrátí nás to zpět na příkazový řádek. Dalším krokem je instalace do systému, to se provede přidáním parametru "install" za příkaz `make`:

```
make install
```

Hotovo. Program je nainstalován a můžeme ho začít používat. Dobrý nápad je smazat pracovní podadresáře, kde byly ukládány dočasné soubory během kompilace. Nejen kvůli volnému místu na disku, později by dočasné soubory mohly způsobit problémy při kompilaci novější verze portu.

```
make clean
```

Tyto tři kroky `make`, `make install`, `make clean` se dají provést všechny najednou:

```
make install clean
```

Některé shelly si udržují cache všech příkazů dostupných v adresářích v proměnné `PATH`, pro rychlejší vykonávání příkazů uživatele. Pokud používáme některý z těchto shellů musíme nejdříve použít příkaz pro obnovení této cache, po instalaci portu, teprve potom onen čerstvě nainstalovaný program můžeme pustit. Více informací nalezneme v dokumentaci našeho shellu.

Kolekce portů používá pro stahování *fetch(3)* stejně jako systém binárních balíčků. Uživatelé kteří nemohou být připojeni po celý čas kompilace můžou jednoduše všechny potřebné soubory stáhnout, odpojit se od internetu a teprve pak začít kompilovat. Stačí příkaz "make fetch" spustit v hlavním adresáři portů (/usr/ports/) a potřebné soubory budou staženy. Tento příkaz pracuje i v nižší úrovni například: /usr/ports/net. Pokud port závisí na nějakých knihovnách a portech, tak závislosti nebudou staženy. V tom případě zaměníme fetch-recursive za fetch a všechny závislosti budou staženy také.

Mazání portů se provádí stejně jako u binárních balíčků. Pomocí utility *pkg_delete(1)*.

Portupgrade

Portupgrade je utilita pro snadnou aktualizaci nainstalovaných portů. Není součástí základního systému, musí se doinstalovat z portu sysutils/portupgrade. Nejprve oscanujeme všechny instalované porty pomocí "pkgdb -F", to spraví všechny nekonzistentní reporty. To je dobrý nápad dělat před každou aktualizací. Teď když pustíme portupgrade s přepínačem -a tak budou aktualizovány všechny nainstalované porty v systému. Pokud přidáme -i tak budeme muset potvrdit každou aktualizaci.

portupgrade -ai

Pokud chceme aktualizovat jednu určitou aplikaci a nikoliv všechny dostupné porty použijeme "portupgrade jmeno_portu". Přidáním -R se nejprve aktualizují všechny porty na kterých aplikace závisí.

portupgrade -R firefox

Chceme-li pro aktualizace použít binární balíček místo instalace ze zdrojových kódů, použijeme -P. S tímto nastavením portupgrade hledá balíčky v adresáři, který je uveden v proměnné PKG_PATH, pokud žádné nenalezne pokusí se balíčky najít na vzdáleném serveru, pokud i na vzdáleném serveru nic nenajde použije kolekci portů.

portupgrade -PR firefox

Další informace můžeme nalézt v manuálových stránkách portupgrade(1).

Portmanager

Portmanager je další utilita pro snadnou aktualizaci nainstalovaných portů. Je dostupná v portu `sysutils/portmanager`.

Všechny instalované porty mohou být aktualizovány jednoduše:

```
portmanager -u
```

Pokud chceme potvrdit každou aktualizaci, můžeme přidat přepínač `-i`. Portmanager může být použit také pro instalaci nových portů do systému. Bohužel obvyklý příkaz `make install clean` nejprve aktualizuje všechny dřívější závislosti vybraného portu.

```
portmanager games/wesnoth
```

Když se vyskytnou problémy v závislostech vybraného portu, můžeme použít portmanager pro rebuild závislostí a opravení potíží. Na konci bude překompilován i problematický port.

```
portmanager graphics/gimp -f
```

Více informací najdeme v manuálových stránkách portmanageru.

Místo na disku

Používáním kolekce portů potřebujeme hodně místa na disku. Po kompilaci a instalaci portu bychom neměli zapomínat mazat dočasné pracovní adresáře příkazem `make clean`. Celou kolekci portů můžeme uklidit následujícím příkazem:

```
portsclean -C
```

Pokud se nám nahromadilo hodně starých zdrojových kódu, můžeme je ručně smazat, nebo použít příkaz pro odstranění všech zdrojových kódů, na které se neodkazuje žádný port:

```
portsclean -D
```

Utilita `portsclean` je součástí balíku `portupgrade`. Pro odstranění portů, které už nepotřebujeme můžeme použít nástroj `sysutils/pkg_cutleaves`.

5.2.4 Po-instalační aktivity

Po instalaci nové aplikace si budeme chtít přečíst dokumentaci, editovat potřebné konfigurační soubory, postarat se o automatické spouštění aplikace po bootu (pokud je to daemon) a podobně. Přesné kroky pro konfiguraci každé aplikace jsou samozřejmě odlišné. Nicméně pokud nainstalujeme novou aplikaci a naše první otázka je "Co teď?", tady jsou tipy pro pomoc:

- Můžeme použít `pkg_info(1)` abychom zjistili, které soubory byly nainstalovány a kam. Například

```
pkg_info -L apache-2.2.0_7
```

ukáže všechny soubory, které patří balíku apache. Speciální pozornost bychom měli věnovat souborům v adresáři *man/*, kde jsou manuálové stránky, v adresáři *etc/* budou konfigurační soubory a v adresáři *doc/* bude obsáhlejší dokumentace.

Pokud si nejsme jistí jakou verzi jsme nainstalovali, můžeme to zjistit pomocí:

```
pkg_version | grep -i apache
```

Tento příkaz nelezne všechny instalované verze programu apache.

- Když už víme kde jsou manuálové stránky nainstalovány, můžeme si je přečíst pomocí *man(1)*, podívat se na vzorové konfigurační soubory, které mohou být dodány s dokumentací.
- Pokud má aplikace webovou stránku, můžeme se podívat na dokumentaci na webu, na často kladené otázky a podobně. Pokud nevíme adresu můžeme zkusit jestli nám jí neřekne:

```
pkg_info apache-2.2.0_7
```

- Porty které se mají spouštět při bootu (obvykle internetové servery), většinou instalují vzorové scripty do `/usr/local/etc/rc.d/`

5.3 X Window System

5.3.1 Úvod

FreeBSD používá X11, aby svým uživatelům poskytl grafické uživatelské rozhraní. X11 je open-source implementace X Window System včetně Xorg a XFree86. Od verze FreeBSD 5.3 je implicitní X11 server Xorg, který je vyvíjen X.org foundation. X není první window system napsaný pro UNIX, ale je nejpopulárnější. Původní vývojový tým X pracoval na jiném window systému, který se jmenoval W (jako Window). X bylo další písmeno v abecedě. X používá "server-klient" model, X server běží na počítači ke kterému je připojen monitor, klávesnice a myš. Server má za úlohu ovládat display, zpracovávat vstupní signály od klávesnice a myši atd. Každá grafická aplikace je klient, a klienti posílají zprávy serveru typu: "Vykresli okno na těchto souřadnicích" a server posílá zprávy jako: "Uživatel kliknul na tlačítko OK".

5.3.2 Konfigurace

Nejprve je potřeba X Window systém nainstalovat pomocí *pkg_add(1)* nebo použitím kolekce port.

Před konfigurací X11 potřebujeme vědět některé údaje o hardwaru.

Specifikace monitoru - Specifikace jsou použity X11 pro stanovení rozlišení a obnovovací frekvence. Ty jdou obvykle najít v dokumentaci monitoru nebo na stránkách výrobce. Potřebujeme dva údaje: horizontální a vertikální frekvence.

Video adapter chipset - Chipset určuje jaký ovladač má X11 použít pro komunikaci s grafickou kartou.

Video adapter memory - Paměť určí rozlišení a barevnou hloubku, na které systém poběží.

Tyto údaje potřebujeme vědět v případě, že autodetekce nebude pracovat korektně.

Nejprve vygenerujeme jako root základní konfigurační soubor:

Xorg -configure

X11 se pokusí oskenovat náš hardware a vygeneruje podle něj v domovském adresáři roota základní konfigurační soubor `xorg.conf.new`. Ted ověříme jestli Xorg podle tohoto konfigu dokáže komunikovat s hardware počítače.

```
Xorg -config xorg.conf.new
```

Jestli vidíme černo šedé pozadí a můžeme pohybovat kurzorem myši, byla automatická konfigurace úspěšná. Pro ukončení testu použijeme klávesovou zkratku `Ctrl+Alt+Backspace`.

Nyní soubor otevřeme v libovolném textovém editoru a doladíme ho k obrazu svému. Nejprve přidáme obnovovací frekvence pro náš monitor. Ty jsou obvykle vyjádřené jako horizontal a vertical synchronization rate, tyto patří do sekce Monitor:

```
Section "Monitor"  
Identifier "Monitor0"  
VendorName "PTS"  
ModelName "DX-987"  
HorizSync 30.0 - 86.0  
VertRefresh 50.0 - 160.0  
EndSection
```

X má schopnost používat DPMS (Energy Star). Program `xset(1)` kontroluje jak dlouho se s počítačem nepracovalo a po určité době může monitor přepnout do úsporného režimu a vypnout ho. Pokud chceme používat DPMS přidáme do sekce monitor řádek:

```
Option "DPMS"
```

Nyní v sekci screen vybereme rozlišení a hloubku barev.

```
Section "Screen"  
Identifier "Screen0"  
Device "Card0"  
Monitor "Monitor0"  
DefaultDepth 16  
SubSection "Display"  
Viewport 0 0  
Depth 16  
modes "1280x1024" "1024x768" "800x600" "640x480"  
EndSubSection  
EndSection
```

Hloubku barev určuje hodnota za *DefaultDepth*, musí existovat podsekcce s takovou hloubkou. Použité rozlišení se píše na řádku *modes*, pokud napíšeme za sebe rozlišení více, můžeme je střídat za běhu X serveru. Nakonec soubor uložíme a překopírujeme na místo, kde Xorg hledá konfigurační soubory, obvykle je to */etc/X11/xorg.conf* nebo */usr/X11R6/etc/X11/xorg.conf*

```
cp /root/xorg.conf.new /etc/X11/xorg.conf
```

Nyní můžeme X server spustit příkazem *startx(1)*. X server může být také spuštěn pomocí *xdm(1)*.

5.3.3 Window Manager

Filosofie X serveru se podobá filosofii Unixu (tools, not policy). To znamená, že X server nezkouší úlohám diktovat, jak mají vypadat. Místo toho uživateli poskytne nástroje a ten se rozhodne, jak je použije. Tedy neříká, jak mají vypadat okna, jakými klávesovými zkratkami se mezi nimi bude přepínat, jaké budou mít okna titulky a podobně. Místo toho poskytuje možnost zavolat aplikaci zvanou Window Manager (správce oken). To window manager, říká jak mají okna vypadat, jakými klávesovými zkratkami se budou ovládat apod. Pro X existuje velké množství window manageru od malých a nenáročných jako Icewm, Fluxbox, Window maker až po kompletní balíky desktopových aplikací jako KDE a GNOME. Uživatel má opravdu hodně na výběr, stačí se podívat do */usr/ports/x11-wm/*.

Po nainstalování některého window managera nebo více je potřeba X serveru říct, který má spustit. To se zapíše do souboru */.xinitrc*. Do tohoto souboru se zapisují příkazy, které se mají provést po startu X serveru, typicky spustit nějaký window manager. Jak se který window manager pouští se dočteme v jeho dokumentaci, jen zmíním že pro KDE je to *startkde*, pro GNOME *gnome-session* a pro IceWM je to *icewm* nebo *icewm-session*

```
echo icewm > /.xinitrc
```

Já nemám moc rád obrázky operačních systémů. Protože ty obrázky neukazují, jak vypadá systém, ale ukazují jak vypadá window manager a jeho theme. Ale protože lidé potřebují něco vidět, na následujících dvou obrázcích je moje pracovní plocha s window managerem IceWM, hodně upraveným konfigem a vlastním themem:

5.4 Zvuková karta

Před konfigurací zvukové karty potřebujeme znát jméno modelu, použitý chipset a jestli je karta PCI nebo ISA. Pokud nevíme jakou máme zvukovkou kartu v počítači můžeme si to zjistit utilitou *pciconf*. *pciconf* s přepínačem *-l* vypíše seznam všech zařízení, které byli nalazeny při bootu operačního systému. Když přidáme ještě *-v* *pciconf* vypíše i jméno výrobce, podrobnější popis zařízení, třídu a podtřídu. Nás bude zajímat třída *multimedia* a podtřída *audio*.

```
pciconf -lv
p      cm0@pci0:17:5:      class=0x040100      card=0x87141565
chip=0x30591106 rev=0x50 hdr=0x00
vendor = 'VIA Technologies Inc'
device = 'VT8233/33A/8235/8237 AC97 Enhanced Audio Con-
troller'
class = multimedia
subclass = audio
```

Ve výpisu všech zařízení najdeme podobné řádky, předchozí příklad říká, že byla nalezena zvuková karta VIA Technologies s chipem VT8233. Na stránce <http://www.freebsd.org/releases/6.0R/hardware.html> teď zjistíme jak se jmenuje modul s ovladačem pro naši zvukovou kartu. Pro tento příklad je to modul *snd_via8233*. Abychom mohli zvukovou kartu používat, musíme modul s ovladačem nahrát do našeho kernelu, to se provede příkazem *kldload(8)*:

```
kldload snd_via8233
```

Abychom tento příkaz nemuseli zadávat po každém nabootování systému, můžeme zapsat do souboru */boot/loader.conf*, aby se modul nahrával automaticky po nabootování systému: *snd_via8233.load="YES"*. Ostatní dostupné moduly, jsou v seznamu */boot/defaults/loader.conf*. Pokud si nejsme jisti jaký modul nahrát můžeme zkusit *snd_driver*. To je meta ovladač, nahrávající nejběžnější ovladače zvukových karet. To je nejrychlejší cesta jak najít správný ovladač.

Nejrychlejší cesta jak otestovat zvukovou kartu je poslat data do */dev/dsp*

```
cat jmeno_souboru > /dev/dsp
```

kde *jmeno_souboru* je libovolný soubor. Tento příkaz vyvolá šum, který potvrdí, že zvuková karta funguje. Hlasitost může být změněna pomocí *mixer(8)*, další informace jsou v manuálových stránkách. Nyní nám už nic nebrání nainstalovat si náš oblíbený multimediální přehrávač jako třeba *audio/mpg123*, *audio/xmms* nebo jiný.

5.5 Lokalizace

5.5.1 i18n/i10n

Vývojáři zkrátili výraz "internationalization" (internacionalizace) na termín *i18n*, který vytvořili spojením prvního a posledního písmene tohoto výrazu a číslem, které udává počet písmen mezi nimi. Termín *i10n* vznikl stejným způsobem odvození z výrazu "localization" (lokalizace). Vzájemnou kombinací *i18n/i10n* metod, protokolů a aplikace uživatelů umožňují používat jím zvolený jazyk.

Internacionalizované aplikace jsou programovány pomocí nástrojů internacionalizačních knihoven. Díky tomu mohou programátoři vytvořit jednoduchý soubor a přeložit všechny zobrazované nabídky a texty do jakéhokoli jazyka.

5.5.2 Používání lokalizace

Nastavení lokalizace se zakládá na třech hlavních složkách: jazykový kód, kód země a kódování. Lokalizační názvy jsou z těchto částí tvořeny následujícím způsobem.

LanguageCode_CountryCode.Encoding

Aby bylo možno systém FreeBSD lokalizovat do konkrétního jazyka, musí uživatel zjistit kódy dané země a jazyka (kódy zemí určují aplikacím, které variace daného jazyka použít). Kromě toho se podle nich rozhodují internetové prohlížeče, SMTP/POP servery, WWW servery atd. Seznam dostupných jazykových kódů zjistíme příkazem *locale -a* pro češtinu jsou k dispozici následující:

cs_CZ.ISO8859-2
cs_CZ.UTF-8

Některé jazyky používají jiné kódování než ASCII. Kódování jazyků může být 8-bitové, široké, či víceznakové. Starší aplikace je neumí rozeznat a pleťou si je s řídicími znaky. Novější aplikace již 8-bitové znaky rozeznají. V závislosti na tom jakou aplikaci jsme nainstalovali, možná budeme muset zkompileovat aplikaci znovu s podporou širokých, případně víceznakových znaků, nebo stávající aplikaci správně nakonfigurovat.

Nastavení lokalizace

Obvykle stačí exportování uživatelského názvu lokalizace jako hodnotu proměnné LANG přihlašovacího příkazového procesoru, což lze provést prostřednictvím uživatelského souboru `/.login_conf` nebo uživatelskou konfigurací přihlašovacího příkazového procesoru (`/.profile`, `/.bashrc`, `/.cshrc`). Tím by měly být nastaveny všechny lokalizační podskupiny (jako `LC_CTYPE`, `LC_TIME` atd.) Ve svých konfiguračních souborech bychom měli nastavit následující dvě hodnoty:

- LANG pro funkce standardu POSIX, *setlocale(3)* a další
- MM_CHARSET pro MIME znakovou sadu aplikací

Existují dvě metody, jakými lze lokalizaci nastavit, přičemž obě dvě jsou popsány dále. První z nich (doporučená) je nastavení proměnných prostředí do přihlašovací třídy. Druhou pak je přidání proměnných prostředí do systémového startovacího souboru příkazového procesoru.

Metoda přihlašovací třídy

Tato metoda umožňuje nastavit proměnné prostředí pro název lokalizace a znakovou sadu MIME každému možnému příkazovému procesoru, namísto přidání stanovených hodnot specifického prostředí do inicializačních souborů všech příkazových procesorů. Nastavení na uživatelské úrovni může být provedeno samotným uživatelem. Nastavení na systémové úrovni vyžaduje superuživatelská práva. Zde je krátký příklad souboru `/.login_conf`, který obsahuje obě proměnné nastavené český jazyk:

```
me:\  
:charset=iso-8859-2:\  
:lang=cs_CZ.ISO8859-2:
```

Pro nastavení na systémové úrovni zapíšeme do souboru `/etc/login.conf`

```
czech:czech users account:\
:charset=iso-8859-2:\
:lang=cs_CZ.ISO8859-2:\
tc=default:
```

Před změnou uživatelských tříd spustíme následující příkaz:

```
cap_mkdb /etc/login.conf
```

to udělá novou konfiguraci viditelnou pro systém. Nyní příkazem *vipw(8)* editujeme soubor s hesly, tak aby vypadal nějak takto:

```
user:password:1111:11:language:0:0:User
Name:/home/user:/bin/sh
```

kde language zaměníme za *czech*, pokud chceme, aby uživatel patřil do třídy *czech*. Pro nové uživatele, které přidáme příkazem *adduser(8)*, můžeme nastavit implicitní třídu v souboru */etc/adduser.conf*

```
defaultclass = czech
```

nebo pokud nenastavíme implicitní třídu, *adduser(8)* se nás na ní zeptá. Třída jde nastavit i když nové uživatele přidáváme příkazem *pw(8)*.

Metoda startovacího souboru příkazového procesoru

Tato metoda není doporučovaná, protože na rozdíl od předchozí není univerzální, protože se u různých shellů liší. Například pro shell *csh* by bylo nastavení v souboru */etc/csh.login*:

```
setenv LANG cs_CZ.ISO8859-2 setenv MM_CHARSET ISO-
8859-2
```

pro příkazový procesor *sh*:

```
LANG=cs_CZ.ISO8859-2; export LANG MM_CHARSET=ISO-
8859-2; export MM_CHARSET
```

Nastavení lokalizace je možné dát i do jiných souborů, ze kterých se spoustějí příkazy, třeba do souboru */.xinitrc*, který se spustí po startu X serveru. (potom je ale české prostředí pouze v grafickém prostředí)

Nastavení konzole

Pro nastavení české klávesnice a českých fontů je nejlepší mít v souboru */etc/rc.conf* uvedeny tyto čtyři řádky:

```
keymap="cs.latin2.qwertz"  
font8x16="iso02-8x16"  
font8x14="iso02-8x14"  
font8x8="iso02-8x8"
```

Zapsání výše uvedených řádků do */etc/rc.conf* může zajistit i instalační program, pokud v menu *Configure* -> *Console* v podmenu *Keymap* zvolíme *Czech ISO (accent)* a v podmenu *Font* zvolíme *ISO 8859-2*. Způsob ovládání klávesnice je uveden v souboru */usr/share/syscons/keymaps/cs.latin2.qwertz.kbd*.

Připojování disků

Pokud někdo chce připojovat WinXX disky s českými názvy souborů, musí u příkazu *mount_msdosfs(8)* případně u *mount_ntfs(8)* navíc uvést parametr *-L*

```
mount_msdosfs -L cs_CZ.ISO8859-2 /dev/ad0s1 /mnt/windows
```

Nastavení české klávesnice

Klávesnici lze nastavit přímo v konfiguračním souboru */etc/X11/xorg.conf* v sekci *InputDevice* s ovladačem klávesnice (*kbd*)

```
Option "XkbLayout" "cz,us"  
Option "XkbOptions" "grp:shift_toggle"
```

V prvním řádku je seznam klávesnic mezi kterými lze přepínat jednoduchou klávesovou zkratkou, která je na druhém řádku (levý a pravý shift). Klávesnici lze také měnit pomocí příkazu *setxkbmap(1)*

```
setxkbmap cz
```

5.6 IPv6

IPv6 (také znám jako "IP nové generace") je nová verze IP protokolu (známý jako IPv4). Stejně jako ostatní BSD systémy i FreeBSD má podporu IPv6 implementovanou. Takže nám nic nebrání v experimentování s IPv6. IPv6 je povolena v základní konfiguraci systému, zjistíme to třeba tak, že zkusíme pingnout localhost, pomocí *ping6(8)*

```
ping6 ::1
PING6(56=40+8+8 bytes) ::1 -j ::1
16 bytes from ::1, icmp_seq=0 hlim=64 time=0.525 ms
16 bytes from ::1, icmp_seq=1 hlim=64 time=0.138 ms
```

Bohužel nativní připojení k internetu protokolem IPv6 není v našich zeměpisných šířkách a délkách běžná věc. Naštěstí existují projekty, které umožňují udělat si ipv4-ipv6 tunel. Takových projektů je více. Já jsem se před několika měsíci zaregistroval a udělal tunel přes www.xs26.net. Bohužel stránka už hodně dlouho není dostupná a sám koordinátor projektu na svých stránkách říká, že projekt odsunul pro nedostatek času na pozadí a je teď v mizerném stavu. Takže pokud chceme používat IPv6 i když ho nemáme nativně musíme použít jiný tunel, například www.freenet6.net. Nicméně, přestože stránka www.xs26.net je nedostupná tunel, který jsem si tam udělal mi v linuxu bez problémů funguje a dlouho ho používám. Chtěl jsem tedy pochopitelně IPv6 i v mém FreeBSD. V linuxu se ipv4-ipv6 tunel spouští:

```
iptunnel add xs26 mode sit local localIPv4 remote remoteIPv4
ttl 64
ifconfig xs26 up
route -A inet6 add default dev xs26
ifconfig xs26 add ffff:ffff:ffff::/48
```

kde *localIPv4* je naše IPv4 adresa, *remoteIPv4* je adresa serveru, přes který vede tunel a *ffff:ffff:ffff::/48* IPv6 rozsah, který mi byl přidělen na www.xs26.net.

K provozování ipv4-ipv6 tunelu na FreeBSD je potřeba tento script přepsat a umístit ho do */usr/local/etc/rc.d/*. V tomto adresáři tedy vytvoříme, třeba soubor *xs26.sh*:

```
#!/bin/sh
echo -n 'IPv6'
ifconfig gif0 create tunnel localIPv4 remoteIPv4 up
ifconfig gif0 inet6 ffff:ffff:ffff:: prefixlen 128
ifconfig xl0 inet6 ffff:ffff:ffff::/128
route add -inet6 default -interface gif0
/usr/sbin/route6d
```

správnou funkčnost tunelu opět vyzkoušíme utilitou *ping6(8)*, pokud vše funguje, jak má, měl by nám pingovaný server nějak takto odpovědět:

```
ping6 irc.ipv6.cesnet.cz
PING6(56=40+8+8 bytes) 3ffe:80ee:29c5:: -j
2001:718:1:4:290:27ff:fedc:70fd
16 bytes from 2001:718:1:4:290:27ff:fedc:70fd, icmp_seq=0
hlim=61 time=48.869 ms
16 bytes from 2001:718:1:4:290:27ff:fedc:70fd, icmp_seq=1
hlim=61 time=50.370 ms
```

Teď nám nic nebrání IPv6 používat, IPv6 podporuje velké množství aplikací od irc klientů, přes internetové prohlíče až po servery jako třeba apache nebo openssl.

5.7 Síťové služby

5.7.1 Inetd

Inetd je někdy nazýván jako "Internetový super server", protože řídí připojení některých služeb. U příchozího připojení inetd rozhodne, pro který program je určeno, spustí speciální proces a pověří odpovědí na socket. Používat inetd pro servery, které nejsou moc využívány, může snížit celkovou zátěž systému, než když běží každý daemon samostatně (ve standalone modu). Inetd je primárně pro spouštění ostatních demonů, ale některé triviální protokoly jsou ovládány přímo jako třeba chargen, auth a daytime.

Inetd je inicializován *rc(8)* systémem. Inetd je implicitně vypnut, ale může být zapnut během instalace systému, zaleží na volbě uživatele. Umístěním

```
inetd_enable="YES"
```

do souboru */etc/rc.conf* povolíme pouštění inetd po bootu systému.

inetd.conf

Konfigurace inetd je v souboru */etc/inetd.conf*. Pokud modifikujeme tento soubor je potřeba říct inetd, aby ho znova načetl:

```
/etc/rc.d/inetd reload
```

Každá řádka konfiguračního souboru určuje jednoho samostatného deamona, formát je následující:

jméno služby

typ socketu

protokol

ostatní nastavení

uživatel

program

argumenty

příklad pro *ftpd(8)* může vypadat třeba takto:

```
ftp stream tcp nowait root /usr/libexec/ftpd ftpd -l
```

jméno služby - Jméno služby určitého démona, musí být obsaženo v seznamu služeb v */etc/services*. Jméno určuje, na kterém portu bude inetd poslouchat. Pokud je vytvořena nová služba musí být nejprve umístěna v tomto seznamu.

typ socketu - Jeden z stream, dgram, raw, drm nebo seqpacket.

protokol - Jeden z tcp, udp, tcp4, udp4, tcp6, udp6, tcp46 nebo udp46

ostatní nastavení - wait nebo nowait, dgram socket musí použít wait, zatímco démoni používající stream socket použijí nowait. wait obvykle vypne mnohonásobné spojení pro jednotlivého démona, zatímco nowait pustí potomka daemona pro každé další spojení. (na příklad na ftpd může být připojeno více klientů naráz).

Další nastavení oddělené lomítkem může být, maximální počet potomků daemona (maximální počet spojení). Pro maximální počet spojení lze ještě nastavit maximální počet spojení z jedné IP za jednu minutu a maximální počet spojení z jedné IP adresy. To je užitečné pro zabránění DoS (Denial of Service).

uživatel - Pod jakým uživatelem démon poběží.

program - Plná cesta k démonu, pokud je démon přímo v inetd použije se *internal*

argumenty - Argumenty programu, na příklad bez argumentů bude hodnota třeba *ftpd* s argumenty *ftpd -l*

Bezpečnost

Někteří daemoni jako fingerd nejsou vždy žádoucí, protože mohou poskytnout informace, které může použít útočník. Někteří démoni si zase nejsou vědomi bezpečí a mají dlouhé nebo nelimitované time outy na vyřízení příchozích připojení. To umožní útočníkovi zpomalit posílání připojení jednotlivým démonům, ten tak zabere další zdroje. Proto je dobrý nápad omezit připojení, když vidíme, že máme na server příliš mnoho požadavků. TCP wrapper je implicitně zaplý. V manuálových stránkách *hosts_access(5)* najdeme více informací o TCP omezeních.

5.7.2 Apache HTTP Server

Některé síťové služby nemohou být spouštěny z inetd a musí běžet v samostatném módu (stand-alone). V případě Apache je to pro lepší výkon při odpovídání na příchozí HTTP požadavky od internetových prohlížečů. Na výběr máme více různých verzí *www/apache13* nebo několik verzí řady 2 *www/apache22*.

Po úspěšném nainstalování Apache, ho před spuštěním musíme nakonfigurovat. Hlavní konfigurační soubor je ve FreeBSD */usr/local/etc/apache/httpd.conf*. Důležitá nastavení jsou:

ServerRoot "/usr/local"

To specifikuje adresářovou strukturu pro Apache. Binární soubory jsou uloženy v podadresářích *bin/* a *sbin/* v *ServerRootu*, konfigurační soubory jsou v *etc/apache/*.

ServerAdmin you@your.address

Adresa kam se má mailovat, pokud je se serverem nějaký problém.

ServerName www.example.org

Nastavení hostname pokud se liší od jména hostu na kterém je konfigurován (můžeme použít `www` místo opravdové adresy)

```
DocumentRoot "/usr/local/www/data"
```

Adresář kde jsou umístěny dokumenty.

Apache má startovací script umístěn v `/usr/local/sbin/apachectl`, pro spuštění mu dáme jako argument `start`, pro zastavení `stop`, pro nahrání změn v konfigu `reload`, více informací můžeme najít v manuálových stránkách `apachectl(8)`. Pro automatické spouštění Apache po startu systému přidáme do `/etc/rc.conf` řádek:

```
apache_enable="YES"
```

PHP

PHP také známé jako PHP: Hypertext PreProcesor je nejrozšířenější jazyk pro vytváření dynamických webových stránek. Pro novější verze apache a php už není potřeba instalovat modul `www/mod_php`, ale stačí nastavit proměnnou `WITH_APACHE2` na `yes` a do konfiguračního Apache přidat nebo upravit následující řádky:

```
LoadModule php5_module libexec/apache2/libphp5.so  
DirectoryIndex index.php index.html  
AddType application/x-httpd-php .php
```

5.8 Kompilace kernelu

Kernel je jádro operačního systému FreeBSD. Je zodpovědný za řízení paměti, vynucení bezpečné kontroly, networking, přístup k diskům a mnoho dalšího. Protože FreeBSD se stává stále více dynamicky konfigurovatelným je občas nutné zkompilevat kernel s vlastním nastavením.

5.8.1 Proč vlastní kernel?

Dříve mělo FreeBSD něco co se nazývalo monolitický kernel. To znamená, že kernel byl jeden velký program, s pevným seznamem podporovaných zařízení. Když bylo potřeba změnit chování kernelu, musel se zkompilevat nový kernel a rebootovat počítač pro nový kernel.

Dnes se FreeBSD rychle blíží modelu, kde jsou funkce kernelu obsažené v modulech, které mohou být dynamicky nahrávány a odstraňovány, jak je potřeba. To umožňuje kernel přizpůsobit novému hardware podle toho jak je dostupný (jako PCIMCIA karty v notebooků) nebo přidat do kernelu nové funkce, které nebyly nutné když byl kernel původně kompilován. To se nazývá modulární kernel.

Navzdory tomu, je stále nutné mít některé části kernelu statické. Protože některé funkce jsou příliš svázané s kernelem, aby mohli být dynamicky nahrazené. V dalších případech to může být proto, že ještě nikdo nenapsal modulární podporu pro takovou funkci.

Kompilace vlastního kernelu je jedním z nejdůležitějších procesů, téměř každý BSD uživatel tím musí projít. Vlastní kernel poskytne systému mnoho výhod. Generický kernel musí obsahovat podporu pro široký okruh hardware, náš vlastní kernel, může podporovat pouze hardware, který máme v počítači. To má několik výhod:

- Rychlejší bootování. Protože kernel bude detekovat pouze hardware, který je v systému, bootování systému se může až dramaticky zrychlit.
- Menší použití paměti. Vlastní kernel často používá méně paměti než generický kernel, to je důležité, protože kernel musí být vždy v reálné paměti. Z toho důvodu je vlastní kernel použitelný na systémech s malou RAM.
- Další podpora hardware. Ve vlastním kernelu můžeme mít podporu pro další zařízení, která nejsou v generickém kernelu

5.8.2 Konfigurace kernelu

Pokud neexistuje adresář `/usr/src/sys/` znamená to, že nemáme nainstalované zdrojové kódy kernelu. Ty musíme doinstalovat pomocí utility `sysinstall`, v menu vybereme `Configure -> Distributions -> src -> sys`. Adresář `/usr/src/sys` je také přístupný přes adresář `/sys`. Nyní se přesuneme do adresáře `arch/conf` a zkopírujeme generický config na nějaké jiné jméno. Příklad:

```
cd /usr/src/sys/i386/conf
cp GENERIC MujKernel
```

Nyní editujeme *MujKernel* naším oblíbeným textovým editorem. Formát souboru je velmi jednoduchý, každý řádek obsahuje klíčové slovo a jeden nebo dva argumenty, většinou jen jeden. Vše za znakem *#* je považováno za komentář a je ignorováno. Podrobný seznam architektur, nastavení a zařízení najdeme v souboru *NOTES*, který je ve stejném adresáři jako soubor *GENERIC*.

Nejprve musíme zadat architekturu počítače, ta musí být jedna z alpha, amd64, i386, ia64, pc98, powerpc, nebo sparc64.

```
machine i386
```

V generickém kernelu máme podporu pro více procesorů, to je zbytečné, necháme pouze ten, který odpovídá našemu procesoru a ostatní odstraníme.

```
cpu I486_CPU cpu I586_CPU cpu I686_CPU
```

Další volba je jméno kernelu, které se bude vypisovat při bootu.

```
ident MujKernel
```

Ve FreeBSD 5.X a novějších se používá *device.hints(5)* pro nastavení ovladačů hardware. Implicitní lokace kterou *loader(8)* kontroluje je */boot/device.hints*. Použitím tohoto nastavení zkompilujeme ovladače staticky do kernelu a soubor *device.hints* pak není potřeba vytvářet.

```
#hints "GENERIC.hints" # Default places to look for devices.
```

Dále následují různá systémová nastavení jako třeba podpora IPv6, NFS, MSDOSFS a atd.

```
Options INET6
Options NFSSERVER
Options MSDOSFS
...
```

Nyní následuje podpora pro hardware. Protože editujeme původní generický konfig budeme spíše podporu pro hardware ubírat než přidávat. Pokud nemáme SCSI pevné disky můžeme odstranit všech cca 30 řádků, které se týkají SCSI. Určitě nepotřebujeme podporu pro 20 síťových karet, ale necháme si pouze tu kterou máme v počítači:

device vr # VIA Rhine, Rhine II

Po dokončení editace, konfig uložíme a můžeme začít kompilovat kernel.

5.8.3 Kompilace a instalace kernelu

Máme na výběr dvě procedury pro kompilaci kernelu. Pokud máme nainstalované zdrojové kódy pouze kernelu, musíme použít první proceduru, pokud kompilujeme kernel jako součást aktualizace celého systému použijeme proceduru druhou.

Procedura 1

1. Spustíme *config(8)* pro generování zdrojového kódu.

/usr/sbin/config MujKernel

2. Přepneme se do adresáře, který *config(8)* vytvoří.

cd ../compile/MujKernel

3. následujícími třemi příkazy ho zkompilujeme a nainstalujeme

*make depend
make
make install*

Procedura 2

1. Přejdeme do adresáře */usr/src/*

cd /usr/src

2. Kernel zkompilujeme

make buildkernel KERNCONF=MujKernel

3. Kernel nainstalujeme

make installkernel KERNCONF=MujKernel

Chapter 6

Závěr

Ve své bakalářské práci jsem se zabýval hlavně operačním systémem FreeBSD, ale pro lepší přehled jsem se zmínil i o ostatních BSD systémech. V úvodu jsem krátce popsal historii vývoje Unixu a vznik BSD. Od třetí kapitoly už se zabývám pouze operačním systémem FreeBSD, kde popisuji historii vzniku a vývojový model. Ve čtvrté kapitole jsem se zabýval rozdíly oproti operačnímu systému Linux, zmínil jsem hlavně rozdílné licence a souborové systémy. V páté a nejrozsáhlejší kapitole jsem popisoval praktické zkušenosti s FreeBSD od instalace, přes správu kolekce portů a nastavení X Window po nastavení a kompilaci vlastního kernelu.

Při používání FreeBSD jsem nenarazil prakticky na žádné potíže, s výjimkou síťové instalace. A mohu tedy o něm říci, že je to stabilní operační systém vhodný jak pro počítače, které slouží jako pracovní stanice, tak pro počítače, které fungují jako internetové servery.

Tato bakalářská práce není a neměla být učebnicí, ale je to spíše zkrácený popis nejdůležitějších kroků, které potřebujeme, pokud chceme FreeBSD používat ať jako pracovní stanici nebo internetový server.

Bibliography

- [1] Jim Mock: *FreeBSD*, Neocortex, Praha, 2002.

Internetové zdroje

<http://www.freebsd.org>
<http://www.root.cz>
<http://www.distrowatch.com>
<http://www.386bsd.org>
<http://www.levenez.com/unix>
<http://www.wikipedia.org>
<http://www.dragonflybsd.org>
<http://www.netbsd.org>
<http://www.openbsd.org>
<http://www.apple.com/macosx>
<http://m0n0.ch/wall/>
<http://www.opendarwin.org/>
<http://www.freesbie.org/>
<http://www.desktopbsd.org/>
<http://www.freebsdforums.org>
<http://www.bsdforums.org/forums/>