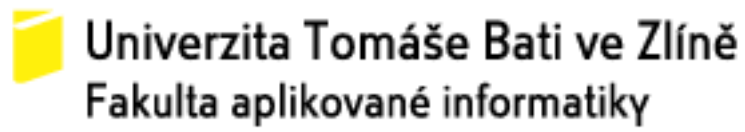


# **Tvorba distribuce OS Linux**

## **Building Linux distribution**

**Martin Zajíc**

\*\*\*nascannované zadání s. 1\*\*\*



Obr. 1. logo

\*\*\*nascannované zadání s. 2\*\*\*



Obr. 2. logo

## ABSTRAKT

*Klíčová slova:*

## ABSTRACT

*Keywords:*

poděkování, motto, úryvky knih, básní atp.

## Prohlašuji, že

- beru na vědomí, že odevzdáním bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo –bakalářskou práci nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- beru na vědomí, že pokud bylo k vypracování bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

## Prohlašuji,

- že jsem na bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně

.....  
podpis diplomanta

## OBSAH

<b>ÚVOD .....</b>	<b>10</b>
<b>I TEORETICKÁ ČÁST .....</b>	<b>10</b>
<b>1 LINUX A SVĚT KOLEM NĚJ.....</b>	<b>12</b>
1.1 OS LINUX.....	12
1.2 DNEŠNÍ POUŽITÍ OS LINUX.....	12
1.3 CO JE TO DISTRIBUCE .....	12
1.4 GNU A LINUX .....	13
<b>2 LINUXOVÉ DISTRIBUCE .....</b>	<b>13</b>
2.1 HLAVNÍ ODLIŠNOSTI.....	13
2.1.1 Cyklus vydání.....	13
2.1.2 Balíčkovací systémy .....	14
2.1.3 Nasazení.....	15
2.1.4 Architektury .....	16
2.1.5 Další rozdíly .....	16
<b>3 DISTRIBUCE LINUX FROM SCRATCH.....</b>	<b>16</b>
3.1 UŽITÍ LINUX FROM SCRATCH.....	17
3.2 DALŠÍ PROJEKTY LFS .....	17
<b>4 SCRIPTOVÁNÍ V JAZYCE BASH .....</b>	<b>17</b>
4.1 ZÁKLADNÍ INFORMACE A PŘÍKLADY .....	18
4.1.1 Komentáře.....	18
4.1.2 Výpis na terminál.....	18
4.1.3 Proměnné.....	18
4.1.4 Roury a přesměrování .....	19
4.1.5 Deskriptor souboru.....	19
4.1.6 Základní příkazy .....	19
4.2 PODMÍNKY A CYKLY .....	20
4.2.1 IF.....	20
4.2.2 CASE .....	21
4.2.3 FOR.....	21
4.2.4 WHILE.....	22
4.2.5 UNTIL.....	22
4.3 FUNKCE,POLE PŘÍKAZY .....	22
4.3.1 Speciální proměnné.....	22
4.3.2 Funkce .....	22
4.3.3 Příkazy .....	23

<b>5</b>	<b>KOMPILACE ZDROJOVÝCH KÓDŮ .....</b>	<b>24</b>
5.1	PŘEKLADAČ (KOMPILÁTOR) .....	24
5.2	DŮLEŽITÉ SOUBORY .....	24
5.2.1	LICENSE (COPYING) .....	24
5.2.2	INSTALL .....	24
5.2.3	Dokumentace .....	24
5.2.4	README .....	24
5.2.5	Configure .....	25
5.2.6	Další soubory a alternativy některých souborů .....	25
5.3	KOMPILACE .....	25
5.4	MOŽNOSTI TVORBY DISTRIBUCE .....	25
5.4.1	Vytvoření ze zdrojových kódů .....	25
5.4.2	Sestavení ze stávající linuxové distribuce .....	26
<b>6</b>	<b>ZABEZPEČENÍ LINUXOVÉHO SYSTÉMU .....</b>	<b>31</b>
6.1	ACCESS CONTROL .....	31
6.1.1	Discretionary Access Control (DAC) .....	31
6.1.2	Mandatory Access Control (MAC) .....	31
6.1.3	Role-based Access Control (RBAC) .....	31
6.1.4	Práva souborů a uživatelů v Linuxu .....	31
6.1.5	Rozšíření zabezpečení .....	31
6.2	INTERNETOVÉ ZABEZPEČENÍ .....	31
6.2.1	Firewall .....	31
6.2.2	Antivirus .....	31
6.2.3	další .....	31
6.3	HYPOTETICKÉ BEZPEČNOSTNÍ HROZBY V LINUXU .....	31
<b>7</b>	<b>TVORBA LIVECD .....</b>	<b>31</b>
7.1	VÝHODY/NEVÝHODY LIVECD .....	31
7.2	MOŽNOSTI TVORBY LIVECD .....	31
7.2.1	Obecný postup .....	31
7.2.2	Linux-Live .....	31
<b>II</b>	<b>PROJEKTOVÁ ČÁST .....</b>	<b>31</b>
<b>8</b>	<b>LFS BY BASH SCRIPTS .....</b>	<b>33</b>
8.1	STRUKTURA A POUŽITÍ .....	33
8.2	NÁVOD PRO SESTAVENÍ .....	33
	<b>ZÁVĚR .....</b>	<b>34</b>
	<b>SEZNAM POUŽITÉ LITERATURY .....</b>	<b>34</b>



---

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK .....	36
SEZNAM OBRÁZKŮ .....	36
SEZNAM TABULEK .....	38
SEZNAM PŘÍLOH .....	39

## ÚVOD

text

# I. TEORETICKÁ ČÁST

## 1 LINUX A SVĚT KOLEM NĚJ

### 1.1 OS Linux

Linux je v informatice označení pro unixový operační systém (původně pouze jeho jádro). Linux je šířen v podobě distribucí, které je snadné nainstalovat nebo přímo používat (tzv. LiveCD). Zároveň se díky použitým licencím jedná o volně šiřitelný software, takže je možné ho nejen volně používat, ale i dále upravovat a distribuovat (kopírovat, sdílet). Tím se odlišuje od proprietárních systémů (např. Microsoft Windows či Mac OS X), za které je nutné platit a dodržovat omezující licence. [6]



Obr. 1. Typický obrázek tučnaka spojovaný s linuxem

### 1.2 Dnešní použití OS Linux

V současnosti je Linux součástí trhu se stolními počítači. Zpočátku se vývojáři Linuxu zaměřovali na síťové systémy a služby, kancelářské aplikace představovaly poslední bariéru, kterou bylo nutno překonat. V tomto trhu dominuje Microsoft, proto v posledních několika letech vznikala spousta alternativních projektů, jejichž cílem je nabídnout Linux jako vhodnou volbu na pracovní stanice. V rámci těchto projektů vznikají snadno použitelná uživatelská rozhraní i kancelářské aplikace (textové editory, tabulkové procesory,...), kopatibilní s aplikacemi Microsoft Office (LibreOffice, Koffice, AbiWord,...) [5]. Co se týče použití na serverech, má Linux pověst stabilní a spolehlivé platformy, na které běží databázové a další služby takových společností jako je Amazon, americká pošta [17], německá armáda [16], Google, Facebook a další. Velmi oblíbený je Linux u poskytovatelů internetového přístupu a služeb, kde se používá jako firewall, proxy server nebo webový server. Počítač s Linuxem najdete i u každého správce některého UNIXového systému, který jej používá jako pohodlnou administrativní stanici. Clustery linuxových počítačů se podílely na vzniku filmů jako Titanic nebo Shrek. Na poštách slouží jako centrály řídící směrování zásilek, velké vyhledávací stroje pomocí nich prohledávají Internet. To je jen ukázka několika z mnoha tisíc náročných úkolů, které dnes Linux na celém světě vykonávají [5].

Stojí také za zmínku, že moderní Linux běží nejen na pracovních stanicích, středně a velkých serverech, ale i na "hračkách", jako jsou PDA či mobilní telefony, ve spoustě zařízení spotřební elektroniky, a dokonce i v experimentálních náramkových hodinkách, Linux je jediný OS na světě, který pokrývá takto širokou škálu HW. [5]

### 1.3 Co je to distribuce

Distribuce, pokud je řeč o distribucích operačního systému, je balíkem jádra operačního systému a doplňujícího SW. Distribuce nejsou jenom záležitostí Linuxu ikdyž jsou často milně spojovány pouze s Linuxem. Kromě Linuxových distribucí existují například distribuce s jádrem BSD (OpenBSD, NetBSD, FreeBSD,...), HURD (ArchHurd), XNU (PureDarwin) a mnoho jiných. Přídavný SW distribuce je čistě volbou autora distribuce, může se tak jednat například o SW projektu GNU (nejvýznamějšími komponenty jsou GCC, glibc, Emacs nebo GNOME) [18], SW projektu BSD, proprietární (uzavřený) SW nebo jakýkoliv jiný SW jehož licence dovoluje použití v distribuci.

## 1.4 GNU a Linux

Rekurzivní akronym pro GNU is Not Unix, kde GNU znamená GNU is Not Unix,... Projekt založený v roce 1984 Richardem Stallmanem a společností Free Software Foundation na vytvoření svobodného a otevřeného operačního systému (dále jen OS) na základě OS UNIX. Zpočátku soubor systémových programů, kterým chyběla hlavní součást - jádro (kernel). Později se jako jádro použil projekt Linuse Torvaldse Linux. V mínění veřejnosti se společný OS GNU/Linux přejmenoval pouze na Linux podle jádra. Jedna z mála distribucí, které dodržují správné pojmenování tohoto OS, je Debian GNU/Linux. GNU obsahuje příkazy každodenní potřeby jako tar, awk, top, dd, ale třeba i Samba, xfig apod. De facto veškerý software vydávaný v rámci projektu GNU je licencován GNU General public license (GPL). Dodnes je vyvíjeno původní microkernel jádro GNU/HURD, avšak ani za 21let vývoje jej vývojáři nedokázali dovést ke stabilnímu vydání. [19] [18] [20]

## 2 LINUXOVÉ DISTRIBUCE

Linuxová distribuce je v informatice označení pro snadno použitelný Linuxový systém, přičemž název je odvozen od jádra Linuxu, které je základní součástí každé distribuce. Distribuce jsou vytvářeny proto, aby uživatel nemusel jádro a doplňující software sám náročným způsobem skládat do funkčního celku. Obsažený software je volně dostupný na Internetu (typicky open source software). Pro odlišení jsou distribuce pojmenovány (např. Ubuntu, Fedora, ...), přičemž každá je jinak zaměřena (pro nezkušeného uživatele, pro vývojáře, výuku atp.).[10]

### 2.1 Hlavní odlišnosti

Linuxové distribuce se liší v mnoha více či méně zásadních ohledech ohledech. Nejvýraznějšími rozdíly, které dělají distribuce distribucemi jsou: Cyklus vydání, balíčkovací systém, základní SW vybava, nasazení,...

#### 2.1.1 Cyklus vydání

V zásadě rozlišujeme 3 vydávací cykly:

**Dlouhý cyklus vydání** Tento cyklus vydání je delší než jeden rok. Používají ho distribuce, které nezakládají na aktuálnosti SW, ale na jeho stabilitě. Především pak **Debian (stable)** a **Red Hat Enterprise Linux** (dále pouze RHEL). Vývojáři Debianu před nedávnem (7. 2009) rozhodli o tom, že bude Debian vydáván v pravidelných intervalech a to tak, že na konci každého lichého roku dojde k zmrazení větve *testing*, ta je následně nějaký čas testována aby byly odstraněny všechny bugy, po jejich odstranění je prohlášena za stabilní a balíčky v ní obsažené jsou přesunuty do větve *stable* <sup>1)</sup>. RHEL vychází v nepravidelných cyklech (cca 2-3roky) a v mezičase vychází updaty. <sup>2)</sup>

**Krátký cyklus vydání** Typicky půl roku až rok. Používá ho většina desktopových distribucí. Tento vydávací cyklus je kompromisem mezi stabilitou a aktuálností

<sup>1)</sup>časový odstup mezi vydáním verze 5 lenny a aktuální 6 squeeze byl 2 roky. Nejdelší odstup mezi vydáními byl 3 roky a to mezi vydáním 3.0 woody a 3.1 sarge

<sup>2)</sup>časový odstup mezi verzí 5 a 6, byl 3.5roku a v mezičase vyšlo 5 update verzí.

SW. Např. **Ubuntu** <sup>3)</sup>, **Mandriva** <sup>4)</sup> (dříve Mandrake), **Fedora** <sup>5)</sup>, ...

**Průběžné aktualizace (rolling-updates)** Distribuce používající rolling-updates se vyznačují především aktuálností SW, většinou udržují větev pro uživatele a testovací větev kde je SW podroben krátkému testování, než se přesune do uživatelské větve. Tyto distribuce většinou vydávají jednou za čas instalační liveCD aby bylo možné je nainstalovat i na aktuální HW. **Gentoo**, **Arch Linux**, **Debian sid (experimental)** <sup>6)</sup>, ...

### 2.1.2 Balíčkovací systémy

Balíčkovací systémy jsou standartní součástí nejen Linuxových distribucí. Balíčkovacím systémem je i systém aktualizací MS Windows nebo oblíbené úložiště aplikací (markety) ve smartphonech (Android market, BlackBerry App World, Ovi Store, Windows Marketplace,...).

**DPKG** *Debian package management system* Jedná se o základní “nízko úroňový” balíčkovací systém pro debian, umožňuje pouze instalaci mazání a výpis balíčků (balíčky jsou vždy s příponou deb).

Nejčastěji se používají programy, které mají rozšířenou funkcionalitu, jsou založené a volají dpkg. Nejznámějším je **APT** (*Advanced Packaging Tool*)[11], od apt se upouští a doporučuje se používat jeho frontend **Aptitude**, protože aptitude dokáže lépe řešit konflikty balíčků, poskytuje jednotné rozhraní pro správu balíčků <sup>7)</sup> a GUI založené na ncurses Pro tento balíčkovací systém existuje i velké množství GUI: synaptic (GTK), Ubuntu Software Center (GTK), KPackage (Qt), ...

Tento balíčkovací systém používají kromě již zmíněného debianu i jeho deriváty jako je Ubuntu, ale i další distribuce a OS. Existuje port pro MacOSX nebo Opensolaris[11].

**RPM** *Red Hat Package Manager* Druhý nejčastěji používaný balíčkovací systém. Je používán množstvím velkých distribucí včele s RHEL (Red Hat Enterprise Linux), také je používán distribucemi Fedora, Mandriva, SUSE, ArkLinux, ...

RPM označuje jak název základního balíčkovacího systému tak i název balíčků samotných (vždy s příponou rpm). RPM se především vyznačuje nepřenositelností z jedné distribuce na druhou (narozdíl od deb balíčků kde to více méně funguje). Proto se lze často setkat s názorem, že rpm neumí pořádně řešit závislosti, tímto názorem se většinou vyznačují lidé, kteří kombinují repozitáře různých distribucí založených na RPM.

RPM se často vyznačuje ještě jedním specifikem a to tím, že na rozdíl od DPKG kde se nejčastěji používají obrovské repozitáře s obrovským množstvím balíčků se u RPM používá velké množství repozitářů s už ne tak velkým množstvím software. Např. Debian používá jeden repozitář pro každou jeho verzi (Stable, Testing, Unstable) kdyžto u RPM distribucí se setkáme s repozitáři zvlášť pro multimedia a core system apd...

**PACMAN** Jedná se pravděpodobně o jeden s nejrychlejších balíčkovacích systémů a je součástí distribuce ArchLinux jejích derivátů larch, FaunOS, Archie, Chakra

<sup>3)</sup>**Ubuntu**: cyklus vydání je půl roku, u jeho derivátů pak zpravidla o něco málo delší. Vydání probíhá pravidelně v dubnu a říjnu, přičemž každé dva roky vyjde LTS verze s podporou na dva roky. Poslední verze: 11.04

<sup>4)</sup>**Mandriva**: cyklus dlouhý půl roku, vychází verze s číslem roku a verze spring. Poslední verze: 2010 Spring

<sup>5)</sup>**Fedora**: cyklus vydání každého půl roku. Poslední verze: 14

<sup>6)</sup>testovací větev debianu, která se jmenuje Sid používá právě rolling-updates, po otestování SW obsažený ve větvi putuje do větve *testing*

<sup>7)</sup>APT je několik aplikací (apt-cache, apt-get,...), které každý poskytují jinou funkci

a je také používán v distribuci DeLi Linux a Frugalware Linux (ve Frugalware linux byl vytvořen fork a jeho vývoj probíhá odděleně). Pacman má výhodu proti jiným balíčkovacím systémům nejen v rychlosti, ale také v tvoření balíčků. Na rozdíl od DPKG, kde existují rozsáhlé návody jak vytvořit balíček stačí u Pacmana vytvořit podle jasných pravidel script s názvem PKGBUILD, který obsahuje všechno od licencí, závislostí, zdrojů kontrolních součtů až po návod jak postupovat při sestavení balíčku, poté stačí napsat makepkg a balíček se sám vytvoří bez jakéhokoliv zásahu.

Na tomto principu fungují jak source repozitáře v ArchLinuxu zvané ABS tak i uživatelské repozitáře AUR.

Pro pacman existuje velké množství wrapperů <sup>8)</sup> pacman-color (přidává do výstupu barvy). Defaultní Pacman také neumí pracovat uživatelským repozitářem AUR proto vzniklo několik wrapperů, které přidávají podporu vyhledávání a instalace balíčků z AUR např. Clyde nebo yaourt. Jiné wrappery zase přidávají podporu stahování z více zdrojů naráz aby byl balíček stáhnut co nejdříve např. PowerPill nebo Bauerbill.

Všechny tyto programy pracují s repozitářem a balíčky stejně jsou tedy kompatibilní a mají dokonce i stejnou syntaxi.

Existují i GUI pro pacman, nejznámější je asi shaman (Qt).

**Portage** Portage je balíčkovací systém GNU/Linuxové distribuce Gentoo Linux, který je podobný systému portů z FreeBSD. Portage je napsán v programovacím jazyce Python. Hlavním příkazem je zde příkaz emerge. Tento balíčkovací systém využívá balíčky ebuild, které zajistí zkompileování podle nastavených systémových proměnných jako jsou například USE, CFLAGS, CHOST a LINGUAS. Portage automaticky dohledá závislosti, stáhne požadované balíčky a program nainstaluje.

Existují různé grafické nadstavby, např. Kuroo (Qt) nebo Porthole (GTK).

**Další,...** Další balíčkovací systémy je např. **slapt-get** (balíčkovací systém, který používá nejstarší dodnes vyvíjená distribuce Slackware a její deriváty (BackTrack, VectorLinux, Slax,...) balíčkovací systém používá obyčejné Tar balíčky komprimované Gzipem), **Equo**, **Smart Package Manager**,...

### 2.1.3 Nasazení

Linuxové distribuce se dále liší podle druhu zařízení na, které jsou určeny. Nejčastěji je to server nebo desktop, ale existují i jiné druhy zařízení na, kterých linux běží.

**User choice (volba uživatele)** Některé distribuce si nedělají starosti s určením na, kterém druhu zařízení budou použity. Poskytují pouze základní sadu nástrojů (programů), které uživatel doplňuje podle svých požadavků. Toto je typická vlastnost Debianu (u Debianu už dnes existují přímo různé konfigurace distribuce jako server, desktop nebo laptop), Archlinuxu, LFS nebo Gentoo.

Tyto distribuce je možné většinou nasadit, na všechny níže zmíněné zařízení.

**Desktop** Dalšími zařízeními na které je linux určen jsou desktopové (personální) počítače. Tyto distribuce poskytují grafické programy pro běžné použití např. v kanceláři. Typicky např. Ubuntu (desktop), Fedora, Mandriva,...

**Server** Serverové distribuce obsahují základní nástroje pro nasazení na serverových počítačích a neobsahují narozdíl desktopových distribucí grafický server Xorg, ale obsahují serverové nástroje jako je např. webový server apache. Typicky serverovou distribucí je např. Ubuntu (server), CentOS, RHEL,...

---

<sup>8)</sup>wrapper je program, který funguje pouze s jiným programem a rozšiřuje jeho funkcionalitu

**Enterprise** Enterprise distribuce jsou distribuce určené pro podnikové nasazení a na rozdíl od běžných distribucí mají placenou podporu ze strany dodavatele a delší testovací dobu např. SLED (SUSE Linux Enterprise Desktop) nebo RHEL.

**Embedded** Jedná se o distribuce směřované na určité zařízení. Nejznámější je např. openwrt což je distribuce určená pro síťové nasazení, např. routery, gatewaye apd,...  
Linux, ale najde i v embedded zařízeních jako jsou settopboxy, televize, IP kamery, pračky, mikrovlnky, ledničky a jiné jednoúčelové zařízení,...

**Mobilní** Asi nejnovějším prostředím ve, kterém momentálně již dominuje systém linux jsou mobilní telefony. Zde snad ani není třeba zmiňovat momentálně nejprodávanější OS pro mobilní telefony od společnosti Google, Android. Existují, ale i další distribuce pro mobilní telefony jako je OpenMoko používaný v mobilu Neo FreeRunner nebo Bada od společnosti Samsung.

další,...

#### 2.1.4 Architektury

Linuxové distribuce jsou také směřovány na různé procesorové architektury, většinou distribuce nesměřují jenom na jednu architekturu, ale na více architektur současně. Příkladem za všechny je distribuce Debian, která je známá také podporou pro největší množství architektur. Např. MS Windows 7 je dostupný pouze pro architektury x86 a x86\_64 (x64), zatímco Debian podporuje oficiálně 10 architektur a neoficiálně 16 architektur. Nejčastěji však většina distribucí podporuje architektury x86 a x86\_64 a v poslední době také ARM.

#### 2.1.5 Další rozdíly

Rozdíly ikdyž ne tak zásadní jsou v softwarové výbavě, ta sice jde ruku v ruce s nasazením linuxu, ale ve světě desktopů a hlavně distribuce Ubuntu se v posledních několika letech oběhují distribuce, které se odlišují pouze v základním uživatelském prostředí. Oficiálně společnost Canonical (tvůrce Ubuntu) vytváří 4 distribuce, které se odlišují od Ubuntu pouze v základním uživatelském prostředí a další 4 distribuce, které se liší v softwarové výbavě.

Velký rozdíl zavádí do linuxových distribucí GoboLinux. Tato distribuce překopává základní rozdělení Linuxového filesystému na více intuitivní a uživatelsky příznivější.

### 3 DISTRIBUCE LINUX FROM SCRATCH

LFS je projekt poskytující návod, který vás provede krok za krokem sestavením vlastní Linuxové distribuce ze zdrojových kódů. [7]



Obr. 1. Logo Linux From Scratch



### 3.1 Užití Linux From Scratch

Spousta lidí se může divit, proč se otravovat se sestavováním LFS, když je možné si jednoduše stáhnout už sestavenou distribuci Linuxu. Avšak i přes některé obtíže jenž sestavování LFS skýtá, má tento proces své výhody. [3]

- LFS učí uživatele jak Linux funguje uvnitř.
- Sestavování LFS učí o všem co v systému běží, jak jednotlivé pracují a závisí jedna na druhé. A jak upravit systém podle chuti.
- Sestavením LFS získáme velice kompaktní Linuxový systém
- Pokud instalujete běžnou Linuxovou distribuci, nainstalujete i množství programů, které pravděpodobně nikdy nepoužijete. LFS jde sestavit i ve velikosti nepřesahující 100MB
- LFS je extrémě flexibilní. Máte moc systém sestavit přesně podle vašich potřeb.
- LFS můžete zabezpečit na míru. Nemusíte čekat, jako u binárních distribucí, až někdo sestaví balíček s bezpečnostním patchem, který potřebujete. Můžete se si zkompilovat SW s patchem sami.

### 3.2 Další projekty LFS

**BLFS** Beyond Linux From Scratch - projekt jehož cílem je vytvořit návod pro rozšíření základního sestavení LFS a umožnit tak uživateli co nejjednodušeji dotvořit vlastní distribuci.

**ALFS** Automated Linux From Scratch - cílem projektu je vytvořit nástroje pro automatické sestavování LFS a BLFS

**CLFS** Cross Linux From Scratch - jak už název napovídá, projekt umožňuje cross-compilaci, tedy sestavení LFS na mnoha jiných typech systémů a architektur.

**HLFS** Hardened Linux From Scratch - se snaží o vytvoření maximálně bezpečného sestavení LFS, odolného proti hrozbám hackerů a jiným bezpečnostním hrozbám.

**Hints** jedná se o kolekci dokumentů, které popisují jak vylepšit vaše sestavení LFS jinak než je popsáno v LFS a BLFS.

**LiveCD** projekt jehož hlavním cílem je vytvořit LiveCD vhodné jako hostitelský systém pro sestavení LFS.

## 4 SCRIPTOVÁNÍ V JAZYCE BASH

(**B**ourne **A**gain **S**hell)

Interpretovaný, neobjektový jazyk, určený především pro administraci a automatizaci \*nix operačních systémů. Skripty obvykle většinu požadované činnosti vykonávají voláním systémových utilit. Podpora syntaxe Bashe je u textových editorů obvyklá. Bash je nainstalován prakticky na každém desktopovém Linuxu a na mnoha dalších \*nix systémech.[13]

## 4.1 Základní informace a příklady

### 4.1.1 Komentáře

Komentáře jsou v bashi jako v jiných jazycích značeny znakem # (sharp)

```
# cokoliv za tímto znakem je komentář
```

Každý script by měl začínat komentářem, který říká jaký interpret shellu se má použít, pokud toto není uvedeno je použit defaultní interpret, což může znamenat problémy pokud jsme napsali script např. pro bash a v systému je defaultně jiný odlehčený interpret.

```
#!/bin/bash
```

Jaký defaultní interpret je použit můžeme zjistit pomocí příkazu:

```
$ echo "/bin/sh -> `readlink -f /bin/sh`"  
/bin/sh -> /bin/bash
```

### 4.1.2 Výpis na terminál

K výpisu na terminál slouží příkaz 'echo'

```
echo hello world
```

echo dokáže vypisovat také proměnné

```
echo $PS1
```

více v manuálových stránkách **man echo**

### 4.1.3 Proměnné

Celý linuxový systém využívá proměnných a funkcí uložených přímo v BASHi. Vypsát všechny proměnné a funkce známé aktuálnímu interpretu příkazů je možno příkazem *set*[14].

Mezi základní proměnné v systému patří např.

```
$USER #uloženo uživatelské jméno  
$LANG #jazyk systému  
$PS1  #nastavení uživatelského promptu  
$BASH #uložena adresa defaultního interpretu příkazů
```

Ukázka práce s proměnnými.

```
$ jedna="Lokální proměnná"
$ export DVA="Proměnná exportovaná do podřízeného shellu"
$ readonly TRI="Proměnná pouze pro čtení, ale jen na lokální úrovni"
$ export TRI
$ export
declare -x DVA="Proměnná exportovaná do podřízeného shellu"
declare -rx TRI="Proměnná pouze pro čtení, ale jen na lokální úrovni"
$ readonly
declare -rx TRI="Proměnná pouze pro čtení, ale jen na lokální úrovni"
$ echo $jedna
Lokální proměnná
$ TRI="Nová hodnota"
bash: TRI: readonly variable
$ bash
$ TRI="Nová hodnota"
$ echo $jedna
$ echo $DVA
Proměnná exportovaná do podřízeného shellu
$ echo $TRI
Nová hodnota
$ unset TRI
```

#### 4.1.4 Roury a přesměrování

Opravdu důležitým prvkem jsou roury a přesměrování. Roura se značí pomocí operátoru `|` a připojuje výstup jednoho procesu na vstup druhého procesu.

```
cat /var/log/auth.log | grep ssh
```

Pro přesměrování slouží operátory:

```
> #přesměrování standardního výstupu do souboru,
    jestliže soubor existuje bude přepsán
>> #jako předchozí, ale data přidá na konec souboru
< #přesměrování standardního vstupu do souboru
<<text #jako předchozí, ale při výskytu řetězce text zašle
    znak konce souboru
```

#### 4.1.5 Deskriptor souboru

BASH rozlišuje 3 deskriptory souboru

**0** standardní vstup

**1** standardní výstup

**2** standardní chybový výstup

```
{
  #funkce a různé procedury
} 2>&1 | tee $BUILD_DIR/LOG_$PROGRAM.log
```

#### 4.1.6 Základní příkazy

**cp** kopíruje soubory

**rm** ruší soubory

**mkdir** vytváří adresáře

**rmdir** ruší prázdné adresáře

**ln** vytvoří odkazy na soubory

**chmod** změni přístupová práva k souborům

**ls, dir, vdir** vypíše obsah adresářů

**find** vyhledávání souborů

**which** zobrazí absolutní cestu k programu

**df** vypisuje informace o připojených FS

**ps** informace o spuštěných procesech

**cat, less** výpis souboru na obrazovku

**xargs** spustí zadaný příkaz a zbylé argumenty čte ze standardního vstupu

**grep** tiskne řádky, které odpovídají zadanému vzoru

**wc** vypíše počet písmen, slov a řádků

**sort** setřídí řádky

## 4.2 Podmínky a cykly

### 4.2.1 IF

Obecná struktura:

```
if výraz;  
    then příkazy  
elif výraz;  
    then příkazy  
else příkazy  
fi
```

Ukázka syntaxe:

```
if [ "$USER" == "root" ]; then  
    echo "Ahoj admin";  
elif [ "$USER" == "Martin" ]; then  
    echo "Ahoj Martine";  
else  
    echo "Ahoj nějaký jiný uživateli";  
fi
```

**POZOR!!!** za znakem [ musí být mezera! Jedná se o program a za mezerou jsou jeho argumenty.

Místo [ je možno použít *test*. Jsou to stejné programy svázané pevným odkazem. Stejná ukázka s použitím programu *test*[14]:

```
if test "$USER" == "root" ; then  
    echo "Ahoj admin";  
elif test "$USER" == "Martin" ; then  
    echo "Ahoj Martine";  
else  
    echo "Ahoj nějaký jiný uživateli";  
fi
```

Podmínky je samozřejmě možno spojovat pomocí operátorů && (a zároveň platí) a || (nebo platí). Operátor || má velké využití ve scriptech pro testování jestly funkce skončila úspěšně[14].

```
#testujeme dvě podmínky
if [ $USER == "root" ] && [ $LANG == "cs_CZ" ]; then
    echo "Jsi český admin"
fi
#pokud funkce skončí chybou script skončí s příznakem 1
funkce || exit 1
```

Operátory testování výrazů:

```
[ výraz ] - délka řetězce je nenulová
[ -z výraz ] - délka řetězce je nulová
[ výraz1 == výraz2 ] - řetězce jsou shodné
[ výraz1 != výraz2 ] - řetězce jsou různé
[ výraz1 -eq výraz2 ] - čísla jsou shodná
[ výraz1 -le výraz2 ] - výraz1 <= výraz2
[ výraz1 -lt výraz2 ] - výraz1 < výraz2
[ výraz1 -ge výraz2 ] - výraz1 >= výraz2
[ výraz1 -gt výraz2 ] - výraz1 > výraz2
[ výraz1 -ne výraz2 ] - čísla jsou různé
```

Operátory testování souborů:

```
[ výraz1 -ef výraz2 ] - soubory sdílejí stejný i-uzel
[ výraz1 -nt výraz2 ] - první soubor je novější
[ výraz1 -no výraz2 ] - první soubor je starší
[ -e výraz ] - soubor existuje
[ -d výraz ] - soubor je adresář
[ -f výraz ] - soubor je obyčejný soubor
[ -L výraz ] - soubor je symbolický odkaz
[ -w výraz ] - soubor je zapisovatelný
[ -x výraz ] - soubor je spustitelný
```

#### 4.2.2 CASE

Obecná struktura:

```
case slovo in
    vzory ) příkazy;;
esac:
```

Výběr grafického prostředí pomocí case:

```
case $1 in
    gnome) exec gnome-session;;
    kde) exec openbox-session;;
    *) echo "vyber gnome nebo kde";;
    #*) znamená cokoliv jiného
esac
```

#### 4.2.3 FOR

příkaz for funguje stejně jako v jiných programovacích jazycích, ale jeho syntaxe je trochu jiná.

```
#příkaz bude postupně do proměnné $cislo dosazovat hodnoty
#a echo je bude vypisovat na obrazovku
for cislo in 10 20 30 40 50 60 70 80 90 100; do
    echo $cislo
done
```

#### 4.2.4 WHILE

```
cislo=0
# Podmínka je splněna jestliže $cislo != 100
while [ "$cislo" -ne 100 ]; do
    cislo=$((cislo + 10))
    echo $cislo
done
```

#### 4.2.5 UNTIL

```
cislo=0
# Cyklus pokračuje dokud není splněna podmínka
until [ "$cislo" -eq 100 ]; do
    cislo=$((cislo + 10))
    echo $cislo
done
```

### 4.3 Funkce,pole příkazy

#### 4.3.1 Speciální proměnné

Informace o názvu skriptu, počtu předaných argumentů a argumenty samotné jsou uloženy ve speciálních proměnných. Tyto proměnné se používají stejným způsobem ve scriptech i ve funkcích[14].

**\$0** název skriptu

**\$#** počet předaných argumentů

**\$IFS** seznam znaků, který je použit k oddělování slov atp., např. když shell čte vstup

**\$1 až \$9** první až devátý argument předaný skriptu

**\$n** libovolný n-tý argument předaný skriptu

**\$\*** obsahuje všechny argumenty oddělené prvním znakem z \$IFS

**\$@** jako předchozí, ale k oddělení se nepoužívá první znak z \$IFS

#### 4.3.2 Funkce

Provádění funkcí je mnohem rychlejší než provádění skriptů, protože funkce si shell udržuje trvale předzpracované v paměti. Funkce musí být definována dříve než bude použita. Příkaz export lze použít i pro funkce, ale musí být zapnutý mód allexport[14].

```
$ set -o allexport
$ prvni_funkce() {
> echo "Jsem první funkce a vypisuji text"
> }
$ export prvni_funkce
$ prvni_funkce
Jsem první funkce a vypisuji text
$ bash
$ prvni_funkce
Jsem první funkce a vypisuji text
```

Pomocí klíčového slova `local` můžeme také vytvořit lokální proměnné funkce. Jestliže bude existovat globální proměnná se stejným názvem, bude ve funkci potlačena[14].

```
#!/bin/bash
jedna="První globální proměnná"
dva="Druhá globální proměnná"
lokalni_promena() {
    local jedna="První lokální proměnná"
    echo $jedna
    echo $dva
}
lokalni_promena
echo $jedna
echo $dva
```

### 4.3.3 Příkazy

Příkazy můžeme rozdělit na zabudované a normální. Zabudované příkazy nemůžeme spustit jako externí programy, ale většinou mají své ekvivalenty ve formě externích programů. Normální příkazy jsou externí programy a jejich vykonání je pomalejší než u zabudovaných příkazů[14].

**break** vyskočí z cyklu

**:** nulový příkaz

**continue** spustí další iteraci cyklu

**.** provede příkaz v aktuálním shellu

**eval** vyhodnotí zadaný výraz

**shift** posune poziční parametry

**read** načte uživatelský vstup, jako argument se použije název proměnné, do které se má uložit

**stty** mění a vypisuje charakteristiky terminálové linky

**exec** spustí nový shell nebo jiný zadaný program a nebo upraví deskriptor souboru

**exit** **n** ukončení skriptu s návratovým kódem **n** (**n** = 0 - úspěšné ukončení, **n** = 1 až 125 - chyba, ostatní **n** jsou rezervovány)

**printf** není dostupný ve starých shellech a při vytváření formátovaného výstupu byste mu měli dávat přednost před příkazem `echo` podle specifikace X/Open

## 5 KOMPILACE ZDROJOVÝCH KÓDŮ

### 5.1 Překladač (kompilátor)

Překladač (též kompilátor, anglicky compiler z to compile – sestavit, zpracovat) je v nejčastějším smyslu slova nástrojem používaným programátory pro vývoj softwaru. Kompilátor slouží pro překlad algoritmů zapsaných ve vyšším programovacím jazyce do jazyka strojového, či spíše do strojového kódu. Z širšího obecného hlediska je kompilátor stroj, respektive program, provádějící překlad z nějakého vstupního jazyka do jazyka výstupního. Z matematického hlediska je kompilátor funkce, která mapuje jeden nebo více zdrojových kódů podle překladových parametrů na kód ve výstupním jazyce. [8]

### 5.2 Důležité soubory

Zdrojové kódy programů jsou uloženy souborech rozličných formátu podle účelu a jazyka v němž je program napsán. Ale každý program by měl obsahovat i další soubory, které mají význam pro uživatele, kteří chtějí program zkompileovat nebo ho dále upravovat. Tyto soubory jsou, ale spíše dobrým mravem a záleží na programátorovy jestly je jeho projekt bude obsahovat.

#### 5.2.1 LICENSE (COPYING)

Soubor obsahující licenci pod kterou je program vydán. Licence je důležitá, protože určuje jak smí uživatel s programem nakládat. U programů s otevřeným zdrojovým kódem se neastějí používají licence GNU GPL a BSD.

**GPL** v jednoduchosti říká, že můžete s programem libovolně nakládat, avšak pokud provedete jakékoliv modifikace, musíte zdrojové kódy opět uvolnit pod stejnou licenci (*jedná se o licenci tzv. virovou*). Anglický originál licence je lze možno nalézt na [opensource.org](http://opensource.org) Pod touto licenci je šířeno Linuxové jádro i sada nástrojů projektu GNU, které Linuxové jádro standardně distribuováno.

**BSD** Umožňuje volné šíření licencovaného obsahu, přičemž vyžaduje pouze uvedení autora a informace o licenci, spolu s upozorněním na zřeknutí se odpovědnosti za dílo. [9] Anglický originál licence je lze možno nalézt na [opensource.org](http://opensource.org). Pod touto licenci šířeno např. jádro stejnojmenného operačního systému BSD nebo operační systém HAIKU. Často bývá licence uvedena jako součást souboru README.

#### 5.2.2 INSTALL

Soubor INSTALL obsahuje instalační informace pro uživatele. Obsahuje informace o tom jak program zkompileovat a informace o jeho nastavení. U menších projektů bývají často tyto informace součástí souboru README, naopak u velkých projektů se často tyto informace přesunují na internet.

#### 5.2.3 Dokumentace

Dokumentace obsahuje informace o používání programů a je velmi důležitá především u knihoven, protože obsahuje popis funkcí, které daná knihovna poskytuje. Většina distribucí poskytuje dokumentaci v samostatných balíčcích, protože spousta uživatelů dokumentaci nijak nevyužije a ušetří se tím místo v uživatelské počítači a především se tím sníží zatížení a množství přenesených dat z repozitářů distribucí.

#### 5.2.4 README

Soubor obsahuje základní informace o programu. Může, a často i obsahuje informace ze všech výše uvedených souborů. Avšak někdy není vůbec, protože je vše uvedeno ve výše uvedených souborech.



### 5.2.5 Configure

Soubor s nastaveními pro překlad. Jedná se v zásadě o BASH script, kterým můžete nastavit co a jak se má v programu kompilovat. Různými volbami, které jsou uvedeny v INSTALL, README, nebo v nějaké online dokumentaci, můžete nastavit *např. cesty kam se mám program nebo jeho části nainstalovat nebo také které části programu se mají přeložit*. Configure je skriptem BASHe a proto je také tak volána, výsledkem configure je vygenerování souboru, které řídí překlad.

### 5.2.6 Další soubory a alternativy některých souborů

## 5.3 Kompilace

Samotná kompilace sestává s prostudování nejčastěji souborů INSTALL nebo README a postupem podle návodu v nich. Ale obecně ji lze zahrnout do posloupnosti tří “příkazů”.

```
./configure [options]
```

vygeneruje soubory potřebné pro překlad

```
make
```

zkompiluje zdrojové kódy <sup>1)</sup>

```
make install
```

nahraje soubory do systému všechny potřebné soubory

## 5.4 Možnosti tvorby distribuce

Linuxovou distribuci lze vytvořit více způsoby, nejběžnějším způsobem je sestavení ze zdrojových kódů. Je třeba dávat pozor na licenční ujednání a dle GNU/GPL licence (pod kterou je distribuováno linuxové jádro) musí autor uvolnit zdrojové kódy. U linuxového jádra to zahrnuje většinou použité patche.

### 5.4.1 Vytvoření ze zdrojových kódů

Vytvoření ze zdrojových kódů reprezentuje například použitá ”distribuce” Linux From Scratch. Postup se obecně skládá z několika částí, které jsou více rozepsány v projektové části.

1. Prve je třeba oddělit systém a vytvořit toolchain, který slouží k sestavení nové distribuce nezávislé na předchozím hostovském systému.
2. sestavení vlastní sady základních nástrojů pro použití systému. Sada obsahuje překladače, nástroje pro práci se soubory a jiné potřebné programové vybavení.
3. nastavení bootovacího procesu systému.
4. vytvoření tabulky souborových systémů, kompilace jádra systému, nastavení bootloaderu

---

<sup>1)</sup>make neslouží pouze ke kompilaci zdrojových kódů programů, ale i jako univerzální utilita pro překlad projektů. LFS jej například používá pro generování PDF,HTML,... verze knihy z XML souborů

### 5.4.2 Sestavení ze stávající linuxové distribuce

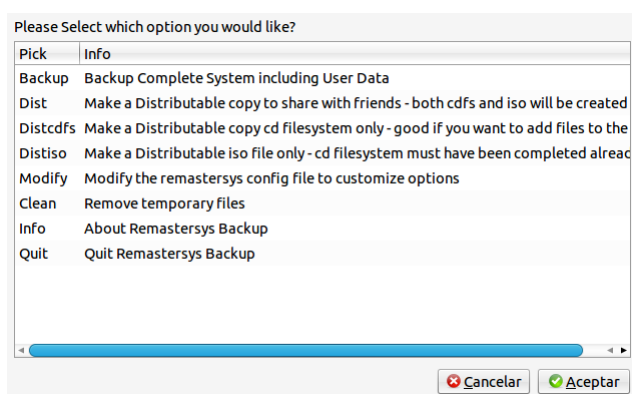
Existuje hned několik nástrojů na sestavení tzv. derivátu Linuxové distribuce. Většinou jsou používány pro sestavení instalačního ISO obrazu na míru potřeb uživatele. To zahrnuje balíčky použité při instalaci, různé nastavení, grafická témata, písma apd,... Populární se staly především deriváty distribuce Ubuntu, pro Ubuntu existuje i nejvíce těchto nástrojů. K dispozici jsou nástroje jak pro desktop s běžným GUI nebo CLI, tak i nástroje webové, které sestaví distribuci v cloudu a uživatel stáhne jenom hotový obraz.

#### Remastersys

Remastersys je jednoduchý grafický nástroj určený pro Ubuntu, Debian a všechny jejich deriváty. Remastersys dokáže převést aktuální běžící distribuci na vašem počítači do ISO obrazu. Je výborným nástrojem pro zálohování dat systému, protože uloží všechny vaše uživatelská data a nastavení.[15]

Výpis možností použití Remastersys: Jedná se opravdu o jednoduchý program.

```
$ remastersys
Usage of remastersys 2.0.18-1 is as follows:
  sudo remastersys backup|clean|dist [cdfs|iso] [filename.iso]
Examples:
  sudo remastersys backup custom.iso
    (to make a livecd/dvd backup and call the iso custom.iso)
  sudo remastersys clean
    (to clean up temporary files of remastersys)
  sudo remastersys dist
    (to make a distributable livecd/dvd of your system)
  sudo remastersys dist cdfs
    (to make a distributable livecd/dvd filesystem only)
  sudo remastersys dist iso custom.iso
    (to make a distributable iso named custom.iso but only
                                     if the cdfs is already present)
```



Obr. 1. Grafické rozhraní programu Remastersys

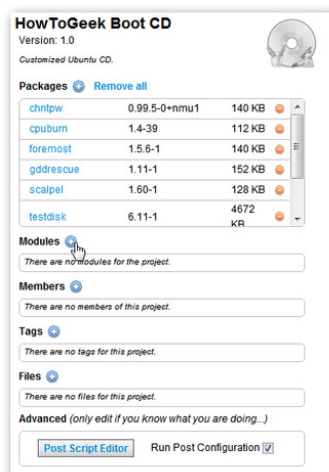
#### The Ubuntu Customisation Kit (UCK)

Program je určen pro Ubuntu a všechny jeho deriváty. Dovoluje odebrat nebo přidat jakékoliv aplikace na instalačním CD. UCK je určeno spíše pokročilejším uživatelům, protože po vytvoření LiveCD, do něj přihlásí uživatele prostřednictvím chroot a dovolí

mu tak změnit prakticky cokoliv. Aplikace má jednoduché grafické rozhraní se spoustou dialogů, které vás provedou vytvořením LiveCD.[15]

## Reconstructor

Jedinou placenou aplikací je program Reconstructor. Je určen pro vytvoření vlastního instalačního CD ze stávajícího ISO obrazu. Umožňuje změnit wallpaper, témata, icony, aplikace a mnoho jiného. Jedná se o webovou aplikaci a její cena je \$5.[15]



Obr. 2. Webové rozhraní Reconstructoru

## Revisor

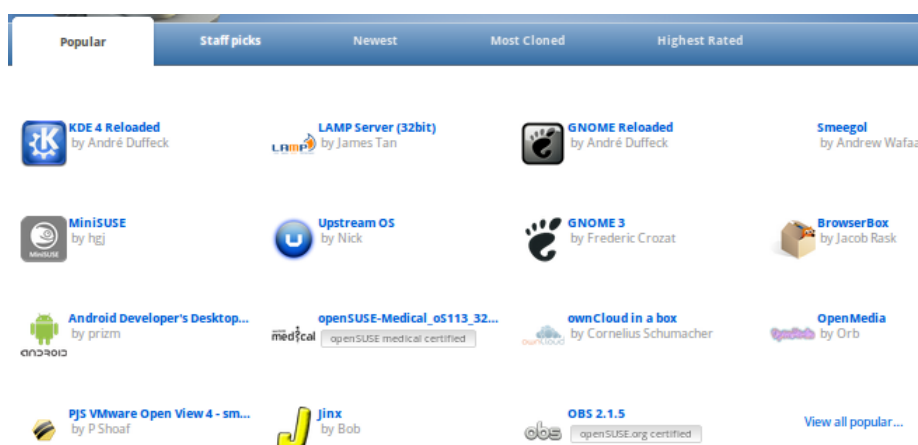
Revisor je aplikace pro vytvoření vlastního instalačního CD/DVD/USB založeném na distribuci Fedora. Má jak grafické rozhraní tak CLI. Další rozdíl je také to, že Revisor nevytváří instalační medium z existujícího obrazu, ale stahuje vše z internetu, takže rychlost vytvoření média je také závislá na rychlosti vašeho připojení.[15]



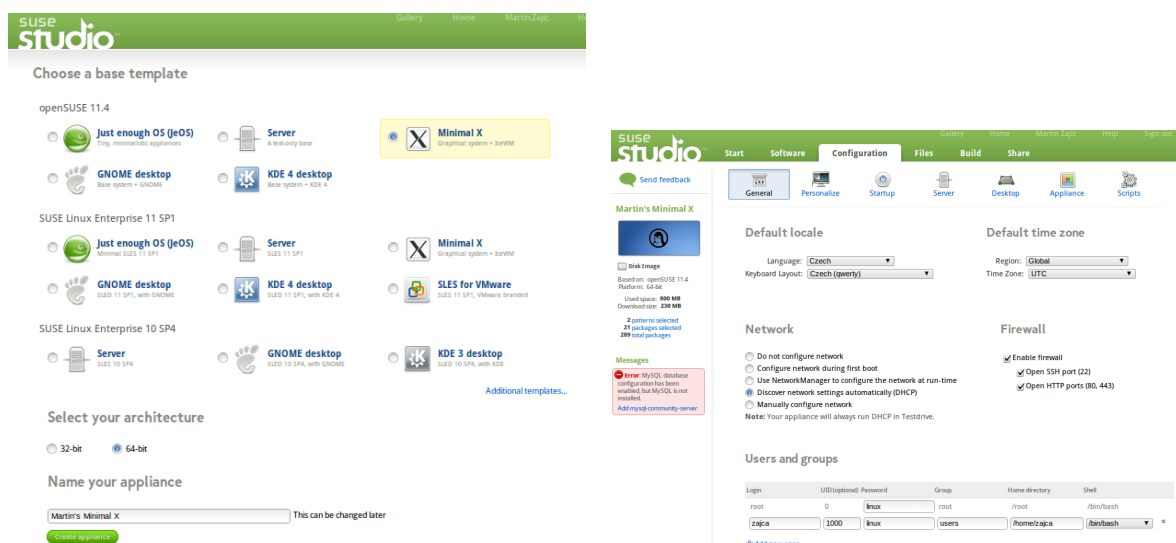
Obr. 3. Grafické rozhraní programu Revisor

## SUSE Studio

Asi nejzajímavějším programem je webová aplikace od společnosti Novell, SUSE Studio. Nejenomže umí nakonfigurovat velké množství nastavení. Ale velice dobrou vlastností je testování systému přímo v prohlížeči. Výborná je také galerie, která obsahuje obrazy vytvořené ostatními uživateli, lze tak bez práce najít už funkční obraz dle potřeb uživatele a případně si již vytvořený obraz upravit.



Obr. 4. Webové stránka s výběrem již existujících sestavení



Obr. 5. Webové rozhraní s nastavením systému, v levo výběr základního rozhraní, v pravo podrobnější nastavení

## Pungi

Pungi je nástrojem, který používají vývojáři Fedory pro sestavení oficiálních vydání. Jedná se o program bez grafického rozhraní napsaný v pythonu a stejně jako Revisor stahuje balíčky z internetu a sestavuje z nich instalační médium.[15]

## Builder

Builder je série scriptů napsaných v BASHi, které používají vývojáři derivátu distribuce Ubuntu "gNewSense" na sestavení jejich distribuce. Ke scriptům je také napsaný přehledný manuál jak je používat, který je dostupný online [15].

<http://www.gnewsense.org/Builder/HowToCreateYourOwnGNUlinuxDistribution>

## MySlax Creator

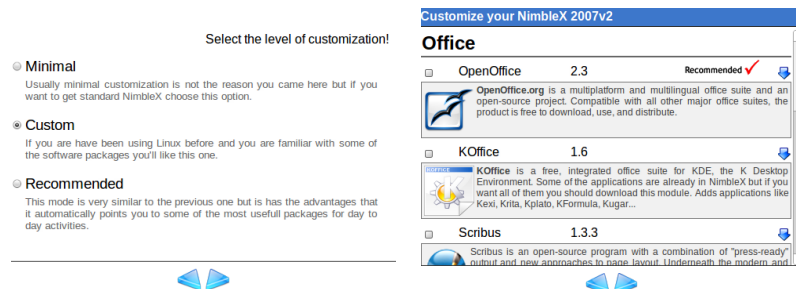
MySlax Creator jsou scripty určené k úpravě distribuce Slackware, především tedy k vytvoření vlastního obrazu odvozené distribuce SLAX. Jejich odlišnost je v tom že fungují přímo z Windows.

## NimbleX

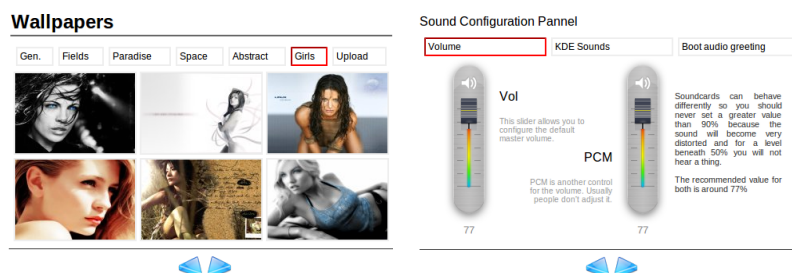
Dalším nástrojem je jednoduchá webová aplikace na vytvoření vlastního derivátu NimbleX. Umožňuje vytvořit vlastní obraz několik jednoduchými kroky, na výběr je stáhnutí minimální obrazu nebo sestavení vlastního obrazu. U vlastního obrazu si uživatel vybere aplikace přehledně rozdělné do kategorií, wallpaper, nastavení zvuku, nastavení uživatelů a jazyk systému. Pak už jen počká na vygenerování obrazu, který je následujících 12hodin ke stažení.

## Instalinux

Jednoduchým, ale mocným nástrojem je Instalinux, umožňuje uživateli vybrat z několika distribucí (Debian, Ubuntu, CentOS, Fedora, OpenSUSE, Scientific) a v dalších krocích základní nastavení systému a výběr aplikací. Výsledkem je ISO obraz, kterým nemusí mít ani 30MB. Pro podrobnější nastavení existuje Designer Pro, který umožňuje nastavit mnohem více. Výbornou vlastností je uložení profilu, který je možné použít

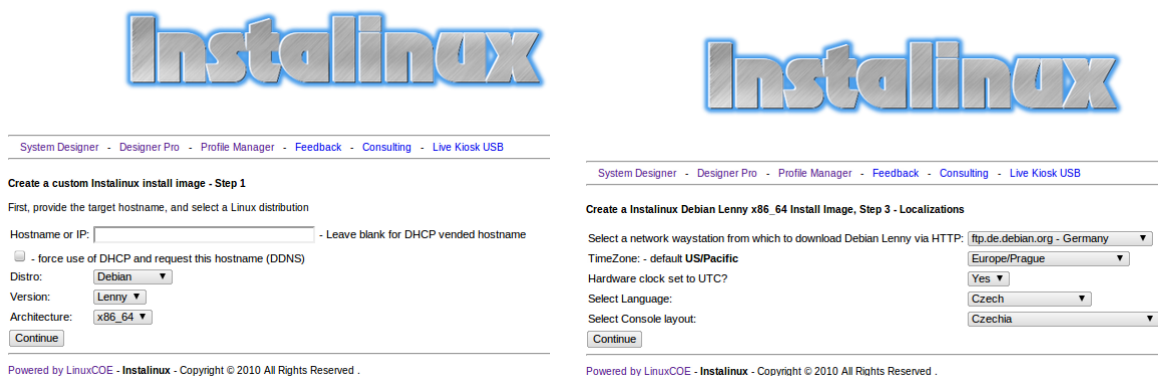


Obr. 6. Webové rozhraní s nastavením NimbleX, v levo výběr míry nastavení, v pravo nastavení programů v sekci office



Obr. 7. Webové rozhraní s nastavením NimbleX, v levo výběr wallpaperu, v pravo nastavení zvuku

znovu a upravit si jej.



Obr. 8. Webové rozhraní Instalinux, v levo výběr distribuce, v pravo nastavení jazyka a časové zóny

Takto vytvořená distribuce může mít 8MB!

System Designer - Designer Pro - Profile Manager - Feedback - Consulting - Live Kiosk USB

Create a Instalinux Debian Lenny x86\_64 Install Image, Step 4 - Pick Debian Lenny Bundles

**Software package selection**

Select Debian Lenny software bundles:

Select All Unselect All Invert Selections

Individual Debian Lenny debs:  
e.g. "openssh", not "openssh-2.1-386.rpm|deb"  
openssh, cherokee, ruby

☒ database-server  
☐ desktop  
☐ dns-server  
☒ file-server  
☐ laptop  
☐ mail-server  
☐ print-server  
☒ web-server

**Choose disk partitioning method additional options**

☐ Autopartition - just swap and / on this drive  
☐ Desktop Machine (as defined by Debian installer)  
☐ Multi-user workstation (as defined by Debian installer)  
☒ None - I want to do this interactively during install - good for multi-boot/pre-existing OS

Continue

Powered by LinuxCOE - Instalinux - Copyright © 2010 All Rights Reserved .

System Designer - Designer Pro - Profile Manager - Feedback - Consulting - Live Kiosk USB

Create a Instalinux Debian Lenny x86\_64 Install Image, Step 5 - Confirm your intentions

Here is your data as entered, please confirm:

**General**

Distro - Arch - Profile	Debian Lenny - x86_64 - Custom
Method	HTTP
Path	ftp.de.debian.org/debian/

**Network**

**ALWAYS USE DHCP**

TimeZone: EuropePrague

**Set root's post-installation password - REQUIRED!**  
Best practice is to choose a temporary password and change it after install.

Enter root password: \*\*\*\*\* - 8 characters MAX!

Confirm root password: \*\*\*\*\*

**Configure a mortal user account- REQUIRED or you will be prompted interactively!!**

Username/Login: zajca

Real Name: Martin Zajic

Enter user's password: \*\*\*\*\* - 8 characters MAX!

Confirm user's password: \*\*\*\*\*

Go For It! - Looking good, create my custom install image

Powered by LinuxCOE - Instalinux - Copyright © 2010 All Rights Reserved .

Obr. 9. Webové rozhraní Instalinux, v levo výběr aplikací, v pravo nastavení uživatelů

## 6 ZABEZPEČENÍ LINUXOVÉHO SYSTÉMU

### 6.1 Access Control

#### 6.1.1 Discretionary Access Control (DAC)

#### 6.1.2 Mandatory Access Control (MAC)

#### 6.1.3 Role-based Access Control (RBAC)

#### 6.1.4 Práva souborů a uživatelů v Linuxu

#### 6.1.5 Rozšíření zabezpečení

### 6.2 Internetové zabezpečení

#### 6.2.1 Firewall

#### 6.2.2 Antivirus

#### 6.2.3 další

### 6.3 Hypotetické bezpečnostní hrozby v linuxu

## 7 TVORBA LIVECD

### 7.1 Výhody/nevýhody LiveCD

### 7.2 Možnosti tvorby LiveCD

#### 7.2.1 Obecný postup

#### 7.2.2 Linux-Live

Linux-Live je sada scriptů, která není závislá na distribuci, ale lze ji použít na jakémkoliv linuxu. Prakticky stačí stáhnout scripty ze stránek linux-live a zkompilevat si kernel s podporou aufs a squashfs nebo si stáhnout již zkompilevaný kernel ze stránek linux-live. Scripty už pak samy vytvoří Live system, který je možné nabootovat z CD nebo USB.

## II. PROJEKTOVÁ ČÁST



## 8 LFS BY BASH SCRIPTS

### 8.1 Struktura a použití

### 8.2 Návod pro sestavení

## ZÁVĚR

text

## SEZNAM POUŽITÉ LITERATURY

- [1] COOPER, Mendel. *Advanced Bash-Scripting Guide* [online]. Revision 6.2. [s.l.] : [s.n.], 17 March 2010, Dostupné z WWW: <http://www.tldp.org/LDP/abs/html/index.html>. [e-kniha]
- [2] MILAR, Bohdan. *Bash očima Bohdana Milara* [online]. [s.l.] : LinuxExpres, 2008. Dostupné z WWW: <http://www.root.cz/knihy/bash-ocima-bohdana-milara/>. [e-kniha]
- [3] BEEKMANS, Gerard. *Linux From Scratch* [online]. 6.7. [s.l.] : [s.n.], 1999, 2010. Dostupné z WWW: <http://www.linuxfromscratch.org/lfs/view/stable/>. [e-kniha]
- [4] KROAH-HARTMAN, Greg. *Linux kernel in a nutshell* [online]. [s.l.] : O Reilly Media, Inc., 2006. Dostupné z WWW: <http://www.root.cz/knihy/linux-kernel-in-a-nutshell/>. [e-kniha]
- [5] KOLEKTIV, Autorů, et al. *Linux: Dokumentační projekt* [online]. 4. vyd. Brno : Computer Press, 2008. Dostupné z WWW: <http://www.root.cz/knihy/linux-dokumentacni-projekt-4-vydani/>. ISBN 978-80-251-1525-1. [e-kniha]
- [6] LITERÁK, Leoš. *Co je to Linux. AbcLinuxu.cz* [online]. 17.1.2006, [cit. 2011-03-30]. Dostupné z WWW: <http://www.abclinuxu.cz/ucebnice/uvod/co-je-to-linux>.
- [7] BEEKMANS, Gerard. *Linux From Scratch* [online]. 1998 [cit. 2011-02-04]. Linux From Scratch. Dostupné z WWW: <http://www.linuxfromscratch.org/lfs/>.
- [8] MARTINEK, David. *Fakulta Informačních Technologií VUT* [online]. 23. září 2010 [cit. 2011-03-30]. Překlad programu. Dostupné z WWW: <http://www.fit.vutbr.cz/~martinek/clang/gcc.html>.
- [9] *Nette Framework* [online]. 2011 [cit. 2011-03-30]. Český překlad nové BSD licence. Dostupné z WWW: <http://nette.org/cs/licence/newbsd>.
- [10] *Linuxové distribuce. In Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, [cit. 2011-02-23]. Dostupné z WWW: [http://cs.wikipedia.org/wiki/Linuxové\\_distribuce](http://cs.wikipedia.org/wiki/Linuxové_distribuce).
- [11] *The Debian GNU/Linux FAQ* [online]. 2009 [cit. 2011-03-30]. Dostupné z WWW: <http://www.debian.org/doc/FAQ/ch-pkgtools.en.html>.
- [12] *Dokumentace Gentoo Linuxu* [online]. 2006 [cit. 2011-03-30]. Dostupné z WWW: <http://www.gentoo.org/doc/cs/handbook/handbook-x86.xml?part=2&chap=1>.
- [13] *ABCLinuxu* [online]. 13.11.2010 [cit. 2011-04-29]. Bash. Dostupné z WWW: <http://www.abclinuxu.cz/software/programovani/jazyky/bash>.
- [14] FUCHS, Jan. *ABCLinuxu* [online]. 2003 [cit. 2011-04-30]. Seriál: BASH. Dostupné z WWW: <http://www.abclinuxu.cz/serialy/bash>.
- [15] SHARMA, Mayank. *TechRadar UK* [online]. 2010-1-24 [cit. 2011-05-01]. 10 scripts to create your own Linux distribution. Dostupné z WWW: <http://www.techradar.com/news/software/operating-systems/10-scripts-to-create-your-own-linux-distribution-665247>.

- 
- [16] *InternetNews* [online]. March 19, 2001 [cit. 2011-05-08]. German Army No Longer Uses Microsoft Programs. Dostupné z WWW: <http://www.internetnews.com/business/article.php/716871/Report-German-Army-No-Longer-Uses-Microsoft-Programs.htm>.
- [17] HOCHMUTH, Phil. *Network world* [online]. January 19, 2004 [cit. 2011-05-08]. Linux delivers for U.S. Postal Service. Dostupné z WWW: <http://www.networkworld.com/techinsider/2004/0119linuxcase.html>.
- [18] Free Software Foundation. *GNU Operating System* [online]. 1996 [cit. 2011-05-09]. Dostupné z WWW: <http://www.gnu.org/>.
- [19] *ABCLinuxu* [online]. 2004, 2009 [cit. 2011-05-09]. Gnu - výkladový slovník. Dostupné z WWW: <http://www.abclinuxu.cz/slovník/gnu>.
- [20] KRČMÁŘ, Petr. *Root.cz* [online]. 2011-05-09 [cit. 2011-05-09]. Proč není Hurd ani po dvaceti letech hotový?. Dostupné z WWW: <http://www.root.cz/clanky/proc-neni-hurd-ani-po-dvaceti-letech-hotovy/>.

**SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK**

LFS	Linux From Scratch
OS	Operační systém
HW	Hardware
SW	Software
GPL	General Public License
BSD	Berkeley Software Distribution
EULA	End-User-License-Agreement
RHEL	Red Hat Enterprise Linux
SLED	SUSE Linux Enterprise Desktop
Např	například
APT	Advanced Packaging Tool
DPKG	Debian package management system
GUI	Graphic user interface
RPM	Red Hat Package Manager
GNU	GNU is Not Unix

**SEZNAM OBRÁZKŮ**

Obr. 1. logo .....	2
Obr. 2. logo .....	3
Obr. 1. Typický obrázek tučnáka spojený s linuxem .....	12
Obr. 1. Logo Linux From Scratch .....	16
Obr. 1. Grafické rozhraní programu Remastersys.....	26
Obr. 2. Webové rozhraní Reconstructoru.....	27
Obr. 3. Grafické rozhraní programu Revisor.....	28
Obr. 4. Webové stránka s výběrem již existujících sestavení.....	28
Obr. 5. Webové rozhraní s nastavením systému, v levo výběr základního rozhraní, v pravo podrobnější nastavení .....	29
Obr. 6. Webové rozhraní s nastavením NimbleX, v levo výběr míry nastavení, v pravo nastavení programů v sekci office.....	30
Obr. 7. Webové rozhraní s nastavením NimbleX, v levo výběr wallpaperu, v pravo nastavení zvuku .....	30
Obr. 8. Webové rozhraní Instalinux, v levo výběr distribuce, v pravo nastavení jazyka a časové zóny .....	30
Obr. 9. Webové rozhraní Instalinux, v levo výběr aplikací, v pravo nastavení uživatelů .....	31

## SEZNAM TABULEK

## SEZNAM PŘÍLOH

P I.      Název přílohy

**PŘÍLOHA P I. NÁZEV PŘÍLOHY**