

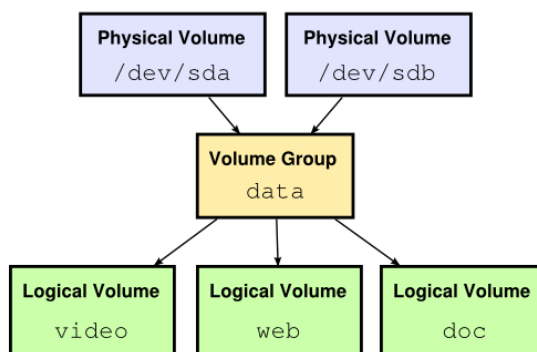
Svazky LVM vs. Standardní rozdělení disků

- Snadné přidávání a rušení nových oddílů
 - U klasické tabulky rozdělení disku jsou možné pouze 4 primární oddíly, potom je třeba použít rozšířené. U LVM je počet prakticky neomezený. LVM také transparentně řeší volné místo pro nové oddíly, u klasického rozdělení disku je třeba mít při vytváření oddílu volné místo v jedné spojitě oblasti.
- Změnu velikosti oddílů
 - LVM umožňuje libovolně zvětšovat a zmenšovat oddíl (s určitými podmínkami například u zrcadleného oddílu).
 - Možnost vytvořit zrcadlený (obdoba RAID-1), prokládaný oddíl (RAID-0) i kombinace zrcadlení a prokládání.
- Vkládat další pevné disky
 - A tím zvětšit dostupný prostor pro nové oddíly. LVM umožňuje i odebrat pevný disk bez ztráty dat, podmínkou je samozřejmě dostatek volného místa. LVM za běhu přesune oddíly na zbylé fyzické oddíly.
- Snapshoty (i zapisovatelné)
 - Snapshot je obraz oddílu vytvořený v učitým čase. Hodí pro např. pro zálohu, kdy zálohovací skript kopíruje data ze snapshotu, který se mu tak nemění pod rukama.

Hierarchie LVM

Koncept LVM lze rozdělit do tří vrstev.

- Fyzický oddíl (Physical Volume - PV) je místo na blokovém zařízení (typicky pevný disk, oddíl či celé diskové pole). Aby šlo využít místo na disku (nebo třeba jen diskovém oddílu) pro LVM, je potřeba na tomto disku/oddílu (obecně blokovém zařízení) vytvořit PV.
- Druhou vrstvou je skupina oddílů (Volume Group - VG). VG v sobě zahrnuje datový prostor ze všech fyzických oddílů do této skupiny přiřazených a toto místo distribuuje do jednotlivých logických oddílů (Logical Volume - LV).
- LV tvoří poslední vrstvu konceptu LVM. Na LV už lze vytvořit systém souborů nebo jej používat jako kterékoliv jiné blokové zařízení. Vše je patrné ze schématu:



LVM vlastně spojí místo z pevných disků či jiných úložných zařízení (PV) do jednoho celku (VG) a poté jej opět rozdělí podle potřeby na jednotlivé oddíly (LV).

Rozdělení disku pro instalaci LINUX

Linux je velmi variabilní systém a proto může běžet na různě rozděleném disku. Dokonce se obejde bez vlastního oddílu a spokojí se i s pouhým souborem na cizím souborovém systému. Nejjednodušší volbou je nic nedělit a nainstalovat celý systém pohromadě. Není to ovšem nejlepší volba – sice nemusíme řešit problém s místem (volné místo je společné), ale přicházíme o celou řadu výhod, které rozdělený disk přináší.

Základní výhody rozdělení disku jsou:

- bezpečnost – dojde-li k poškození některého souborového systému, ostatních se to nedotkne.
- snadná přeinstalace - je-li vše společně, musí se před přeinstalací veškerá uživatelská data někam zkopírovat a po instalaci zase přesunout zpět
- volba souborového systému – lze pro každý oddíl zvolit pro nejvhodnější souborový systém a optimálně ho nastavit. Není tedy nutné dělat zbytečné kompromisy mezi výkonem a bezpečností.
- lepší využití více disků – při využití více disků, lze definovat, kam se umístí určitou část hierarchie souborového systému to vede ke zvýšení rychlosti čtení a zápisu na disky
- možnost ochrany proti zápisu - může zabránit nechtěnému smazání dat, případně zmenšit dopady případného útoku na systém.

Část	Přípojný bod (označení)	Velikost	Poznámka
První část disku pro systém	/ (kořen)	8-15 GB	Dobře uvažte velikost, do této části se kromě samotného systému instalují i programy.
Bootovací část	/boot	dostačující 200MB, doporučeno 500MB	Soubory pro bootování systému.
Odkládací oddíl, "náhradní RAM"	swap	Do 4 GB, obvykle 2×množství RAM. Od 4 GB je dostačující stejná velikost jako RAM.	Důležité také pro uspávání počítače na disk.
Domovské adresáře uživatelů	/home	zbytek diskového oddílu	Prostor pro uživatelská data.

Disk lze sice ještě rozdělit na více oddílů, ale vzhledem ke získaným výhodám je větší rozdělení zbytečné. Plně dostačující je popsáno ve výše uvedené tabulce.

Balíčkovací systémy v GNU/Linux

Úvod

Pojem software je čtenáři z předchozího výkladu dostatečně objasněn a nepřekvapí ho tak nutnost doplnit hardware počítače spolu s operačním systémem i uživatelskými programy. Bez nich by sice počítač fungoval, však toliko pouze jako spotřebič elektřiny a nepříliš kvalitní přímotop.

Jsou to právě uživatelské programy (aplikační programové vybavení), které mění nevzhlednou krabici s přístroj nám užitečný, zjednodušující práci a přinášející i zábavu.

V dřevních dobách výpočetní techniky měl uživatel počítače situaci velmi jednoduchou – pokud chtěl počítač používat, musel po jeho zapnutí napsat program, který následně počítač prováděl. Postupem času se objevily možnosti zapsané programy i uložit do trvanlivějších pamětí, ba dokonce pamětí výměnných. Tím vznikla druhá možnost – program nepsat sám, ale získat ho již hotový. Pro účely této kapitoly nebudeme řešit otázku licencí a plateb za užívání software.

Pro lepší pochopení dalšího textu je také nutné připomenout formy, ve kterých se program může vyskytovat – tedy již samotná myšlenka v hlavě programátora (či lépe analytika), algoritmus zapsaný formálně technickými prostředky (vývojový diagram, struktogram, slovní popis...), zdrojový kód v určitém programovacím jazyce a nebo přeložený sled instrukcí pro konkrétní procesor (možno se setkat i s termínem „platforma“). V dalším výkladu budeme pracovat programy v posledních dvou formách.

Software v GNU/Linux

Rodina operačních systémů založená na jádře GNU/Linux si zakládá především na filosofii Open source – kdy je veškerý software (systémový i aplikační) k dispozici především ve formě zdrojových kódů (nejčastěji v jazyce ANSI C, někdy Perl a podobně). Právým opakem je pak „komerční“ přesněji proprietární software, který je distribuován výhradně v přeložené verzi a samotné zdrojové kódy jsou úzkostlivě utajovány (typicky produkty Microsoft®).

Přístup Open-Source umožňuje (tedy nevyžaduje) zájemcům takový software libovolně upravovat, vytvářet na jeho základě programy vlastní a dále je distribuovat. Uživatel GNU/Linux si může z internetu stáhnout zdrojové kódy například textového editoru a na svém počítači je zkompilovat právě optimálně pro svůj procesor, kapacitu paměti atd. - úplně stejně jako by kompiloval vlastní právě napsaný Hello World!.

Protože komunita svobodného software nevnučuje své myšlenky a postupy nikomu násilím, je mnohdy k dispozici i možnost stažení již přeložené (zkompilované) verze programu pro určitou platformu (zjednodušeně procesor). Takovou verzi často označujeme jako „binární“ či „binárka“. Mnohdy ji stačí stáhnout, spustit a program funguje. Třeba ne s využitím všech možností které právě Váš procesor nabízí, o několik % pomaleji než pokud by byl kompilován přímo pro Váš počítač, ale vcelku bez práce.

Problém ale může nastat, pokud programy obsahují vnitřní závislosti jeden na druhém. Typicky zmíněný textový editor potřebuje pro svoji funkčnost program pro zobrazování znaků, správce tisku, slovník pravopisu a podobně. Řada těchto programů se navíc vyskytuje v mnoha verzích – na rozdíl od uzavřeného software, probíhá vývoj a vylepšování open-source neustále a nové verze programů jsou vydávány i několikrát denně. Zde může být samostatný přístup komplikovaný, kdy zájemce o program musí ručně dohledávat jaké další programy jsou potřebné (tzv. závislosti) a v jakých verzích. Autoři distribucí GNU/Linux tento problém řeší zavedením balíčkovacích systémů.

Balíčkovací systém

Jako balíček označujeme množinu dat (nejčastěji několik souborů zkomprimovaných do jednoho) která v určité distribuci GNU/Linux reprezentuje určitý program. Balíčkovacích systémů je celá řada – historickým vývojem se uživatel nejčastěji setká s RPM (RedHat package Manager, dnes používaný zejména u distribuce Fedora) a DEB (u distribucí Debian, Ubuntu atd.).

V součástí konkrétní distribuce GNU/Linux je pak správce těchto balíčků, který zná jejich vnitřní strukturu a dokáže jimi manipulovat a vzájemně je synchronizovat. Řada základních balíčků je pak i přímo na instalačním médiu určité distribuce (případně si uživatel při instalaci vybírá které chce nainstalovat ihned), všechny ostatní balíčky (zejména v mnoha i historických verzích) pak autor distribuce umísťuje na internet do tzv. repozitářů. Všechny hlavní distribuce pak těchto archivů spravují několik, nejčastěji rozdělených podle své aktuálnosti a otestovanosti (označované například jako „stable“ a „testing“ nebo „released“ a „release-candidate“). Uživatel si pak může zvolit zda chce získávat balíčky absolutně nejnovější (staré většinou jednotky hodin), nebo raději starší verze (dny) u kterých již jiní prověřili jejich bezchybnost.

Zároveň si uživatel může zvolit, zda má zájem o balíčky již předkompilované (pro obecnou platformu jako x86, alpha a podobně), nebo balíčky ve formě zdrojových kódů – které si následně zkompile na svém počítači přesně dle svého počítače.

Zásadní výhodou balíčkovacího systému je, že prakticky veškerý běžně známý software je autory distribucí do balíčků zařazován a uživatel tak nemusí na internetu složitě hledat odkud tu kterou aplikaci stáhnout – jednoduše nahlédne do repozitáře právě své distribuce a stáhne si balíček s požadovaným programem, ve verzi odzkoušené přímo s jeho distribucí. Tedy postup znějící pro uživatele Windows™ jako sci-fi. Odborný přístup je nyní (jako pro mnohé novinka) aplikován u software pro Apple (Apple store) a OS Android (Android market).

Každý stabilní RPM balíček, který je publikován v rámci Fedora projektu je podepsán GPG klíčem. GPG je svobodná alternativa kryptografického software, který je vydaný pod GNU General Public License a využívá asymetrické šifry i symetrické šifry. Implicitně yum a jiné grafické nástroje pro update ověří tyto podpisy a odmítnou nainstalovat balíček, který není podepsán, nebo má špatný podpis. Vždy by jste měli ověřit podpis balíčku dříve, než jej nainstalujete. Tyto podpisy zaručují, že balíček, který se chystáte nainstalovat byl vytvořen v rámci Fedora projektu, a že nebyl změněn (náhodně nebo záměrně).

Balíčkový manager

Aby se instalace software uživatelům GNU/Linux ještě více zjednodušila, jsou všechny moderní distribuce vybaveny i některým typem balíčkového manageru. Ten dovádí problematiku získání a instalace software téměř k dokonalosti, neboť on sám prohledává repozitáře, analyzuje balíčky, dohledává chybějící závislosti a především má kompletní přehled o software, který již v počítači máte nainstalován. Nejčastějšími balíčkovými managery jsou apt, yum nebo yast.

Nehrozí tedy (na rozdíl od Windows®) situace kdy dva programy sdílí stejnou pomocnou knihovnu a později nainstalovaný přepíše již existující novější verzi svou verzí starší, nebo je ta stejná knihovna nainstalována v systému hned několikrát a v několika odlišných verzích. Je právě úkolem správce balíčků zjistit jako jsou závislostní požadavky nově instalovaného programu, ověřit zda jsou v systému přítomny všechny pomocné programy a případně je z repozitáře doinstalovat.

Velmi užitečný je balíčkovací manager i při odinstalaci programu – tedy operaci u komerčních operačních systémů nikdy 100% úspěšné. Tím že balíčkovací manager udržuje vlastní evidenci všech balíčků nainstalovaných na konkrétním počítači, dokáže bez problémů na pokyn uživatele kompletně odstranit nejen požadovaný program, ale i všechny ostatní závislosti, které již nejsou potřeba.

Samozřejmostí je pak možnost automatické aktualizace všech nainstalovaných programů na nejnovější vydávané verze – nejčastěji v nočních hodinách kdy není počítač používán. Uživatel tak ráno přichází ke své stanici, která pracuje více, lépe a rychleji než když ji předchozího dne opouštěl. Na rozdíl od jiných operačních systémů se u GNU/Linux prakticky

jakékoliv aktualizace obejdou bez nutnosti restartu. Nejsou výjimkou stanice a servery pracující bez restartu stovky a tisíce dnů (srovnejte s běžnou stanicí s Windows™ XP).

U všech moderních distribucí jsou pak balíčkové managery provedeny i s grafickým rozhraním („klikací“), kdy uživatel instaluje software pouze tím, že ho vybere ze seznamu a potvrdí své přání právě tento program v počítači mít. Do několika minut mu přibude v nabídce a to rovnou jeho nejaktuálnější verze.

Repozitáře

Studijní materiál: <https://wiki.mojefedora.cz/doku.php?id=navody:prirucka:repozitare>

Instalace SW v GNU/Linux (Fedora)

rpm -Uvh [http://adresa_nebo_cesta_k_balicku](#) – přidání repozitáře

rpm --import [http://adresa_nebo_cesta_klice](#) – import klíče

Aktualizace systému

Aktualizace systému je provedena na základě porovnání aktuálního stavu RPM databáze a stavu v repozitářích tzn. když jsou v repozitářích k dispozici novější verze balíčků, jsou navrženy k aktualizaci.

dnf update

- Automaticky se stáhnou staženy řídicí soubory z repozitářů a podle nich se rozhodne, zda je potřeba nahrát z některých nová metadata.
- Po zpracování metadat sdělí, zda jsou k dispozici nějaké aktualizace.
- Uživatel má možnost s výzvou souhlasit nebo nesouhlasit. Po odsouhlasení aktualizací dojde k výpočtu závislostí a ke stažení balíčků nutných pro aktualizaci.
- Poté je provedena zkušební transakce a je-li vše v pořádku, dojde k samotné aktualizaci tzn.:
 - instalace balíčků
 - z RPM databáze odstraněny záznamy o starších aktualizacích právě nahrazených balíčky.

Hledání balíčku

Při hledání jsou prohledávány jak názvy balíčků, tak jejich popisky. Většina popisů je však aktuálně jen v angličtině.

např. hledání balíčků obsahující klíčové slovo gimp

dnf search gimp

Výstupem je seznam nalezených balíčků a jejich popis.

Hledání balíčku podle názvu

Při hledání balíčku podle názvu jsou prohledávány jak nainstalované, tak dostupné balíčky. Výsledný seznam je předložen uživateli:

```
dnf list gimp
```

Při hledání lze použít hvězdičku, která zastupuje libovolnou posloupnost znaků. Zkuste například hledat *gimp* jen podle prvních dvou písmen (zpětné lomítko ruší speciální význam hvězdičky pro shell, v některých shellech ho není nutné zadávat):

```
dnf list gi\*
```

Instalace balíčku

Při instalaci balíčku je zadán jako parametr jeho název. Pokud balíček potřebuje ke své činnosti jiné balíčky (tzv. závislosti), jsou zahrnuty do instalace.

```
dnf install gimp
```

Odinstalace balíčku

Pokud chcete balíček odinstalovat, jsou zároveň s ním odinstalovány i balíčky, které by přestaly fungovat, kdyby byl vámi požadovaný balíček ze systému odebrán. V některých případech to může vést k tomu, že systém může přestat fungovat (například pokud je odinstalován balíček *glibc* se základními knihovnami). Pokud vám tedy yum nabídne, že odinstaluje další balíčky, pečlivě si to rozmyslete.

```
yum remove firefox
```