



Projekt v rámci SIPVZ:

**IMPLEMENTACE OPERAČNÍHO SYSTÉMU LINUX DO
VÝUKY INFORMAČNÍCH TECHNOLOGIÍ**

LINUX

Lekce 30

Vybrané příkazy - 3

Obsah lekce:

Cíle	1
Kompresní nástroje	1
Gzip, zip, bzip2 – komprese a rozbalení souborů a-29	
Souhrn a-30	
Tar – archivátor a-33	
Manipulace s textem / filtry	5
Kapell – kontrola pravopisu a-35	
Wc – počítadlo a-36	
Cat – slučování souborů a-37	
Head, tail – náhledy souborů a-38	
Grep – tisk řádků shodující se podle šablony a-39	
More, less – stránkovače a-40	
Sort – třídění souborů a-42	

Cíle

Po skončení této lekce studenti budou schopni:

- Ovládat vybrané příkazy

Kompresní nástroje

gzip, zip, bzip2 – Kompresce a rozbalení souborů

Přehled

gzip

Příkaz *gzip* může balit i rozbalovat soubory. Tato utilita změni příponu souboru na *.gz*, ale vlastnictví, přístup a čas modifikace zůstanou stejné. Utilita může být použita pouze na regulérní soubory a ignoruje symbolické odkazy, může být použita na jednotlivé soubory ne na celé adresáře. Další možnosti jsou popsány na konci této sekce.

zip

Podobně jako *gzip*, *zip* je používána pro balení a rozbalování souborů. Na rozdíl od *gzip*, může být *zip* použit pro zabalení celého adresáře do jednoho souboru. Soubory zabalené pomocí *zip* jsou příbuzné souborům *pkzip*, což jsou ekvivalenty v systému MS-DOS. Z tohoto důvodu je *unzip* používán pro rozbalení souborů ze systému MS Windows. Další možnosti jsou popsány v souhrnu na konci této sekce.

bzip2

Jako *gzip* a *zip* je *bzip2* používán pro kompresi a rozbalování souborů. Hlavním rozdílem mezi *bzip2* a ostatními je algoritmus, který připravuje kompresi. Algoritmus použitý v *bzip2* je více efektivnější než v programech *zip* a *gzip*. Stejně jako *gzip*, může být *bzip2* použit pouze na jednotlivé soubory a změni jejich příponu na *.bz2*, *.bz*, *.tbz2* nebo *.tbz*. Další možnosti jsou popsány v souhrnu na konci této sekce.

Scénář

Tento scénář může být ztížen mnoha podmínkami, záležících na druhu použité kompresní utility.

Příklady

1. gzip

Krok 1: Nejprve, *tar* ani *gzip* nemohou být použity na adresář. Tato nepříjemnost *taru* je blíže vysvětlena v sekci *tar* v tomto manuálu. Kompresce jednotlivých souborů není složitá. Příkaz vypadá takto:

\$ gzip filename

Někteří uživatelé si k tomu ještě přejí vytvořit záložní kopii původního souboru, aby zajistili, že pokud komprese selže, nebudou jejich data ztracena. Tuto kopii později smažou.

Krok 2: Nyní když je soubor zkomprimován, měl by mít příponu tar.gz. Existuje mnoho způsobů jak otestovat kompresi a konzistenci dat, což je ukázáno dále.

A. První metodou je využití přepínače -t. Ten otestuje archiv a ohlásí případné chyby uživateli.

\$ gzip -t filename

B. Druhou metodou testování komprese je použití příkazu tar v následujícím tvaru:

\$ tar tvfz file.tar.fz (for GNU tar)

\$ gunzip < file.tar.gz | tar tvf (pro jakýkoliv tar)

2. zip

Příkaz zip může být použit dvěma způsoby. První je komprese souborů do jednoho archivu v jendom kroku. A druhým je využití společně s gzip. Nejdřív použít tar na adresář a posléze použít zip na tar soubor. Druhá metoda dosahuje lepší komprese, ale první je lehčí.

Krok 1: Pro použití samotného příkazu zip:

\$ zip -r dir_name zip_name

dir_name je název baleného adresáře a zip_name je název kam se zabalí, přípona .zip se vytvoří automaticky.

Krok 2: Použití příkazu zip na tar soubor.

\$ zip filename zipname

Význam a fungování je podobné jako předešle.

Krok 3: Pro testování archivu, který byl právě vytvořen použijeme volbu -T.

\$ zip -T zipname.zip

To vrátí stav archivu zprávou OK nebo případnými chybami.

3. bzip2

Krok 1: Podobně jako gzip musí být použit ve spojení s archivátorem jako je například tar. Jakmile je vytvořen tar soubor pro adresář, vypadá příkaz pro jeho kompresi takto:

\$ bzip2 -T filename.tbz2

Jestliže během komprese nastanou nějaké problémy, apk se zde vypíší.

Souhrn

Název příkazu:	gzip
Osnova:	gzip [-acdfhlLnNrtv19] [-S sufix] [name...] gunzip [-acdfhlLnNrtv19] [-S sufix] [name...] zcat [-fhLV] [name..]
Popis	Tento příkaz je používán pro kompresi souborů algoritmem LZ77.
Volby:	-d Dekomprese. -f Rychlá komprese či dekomprese. -h Nápořveda, zobrazí další dostupné parametry. -l Vypíše seznam souborů, původní velikost, současnou velikost a poměr komprese. -q Ticho, neotravuje uživatele s chybami. -r Dovoluje používat adresářovou strukturu. -t Text, testuje integritu komprese.

Tabulka 30-1 – Popis příkazu gzip

Název příkazu:	zip
Osnova:	zip [-aABcdDeEfFghjklMnoqrRSTuvVwXyz!@\$] [-b path] [-n suffixes] [-t mmdyyy] [-tt mmdyyy] [zoofile [file1 file2 ...]] [-xi list] zipcloak [-dHl] [-b path] zoofile zipnote [-hwL] [-b path] zoofile zipsplit [-hiLpst] [-n size] [-b path] zipfile
Popis	Kompresní utilita pro Unix. Často používaná pro dekompresi souborů zip systému MS Windows.
Volby:	-r Umožňuje používat adresářovou strukturu. -x Rozbalí specifikované soubory. -T Testuje komprimovaný soubor.

Tabulka 30-2 – Popis příkazu zip

Název příkazu:	Bzip2
Osnova:	bzip2 [-cdfkqstvzVL123456789] [filenames...] bunzip2 [-fkvsVL] [filenames ...] bzcat [-s] [filenames ...] bzip2recover filename
Popis	Komprimuje soubory použitím algoritmu Burrows-Wheeler a kódování Hoffman. Ta je více efektivní než komprese LZ77.
Volby:	-d Dekomprese. -f Rychlá komprese či dekomprese. -q Ticho. Nezobrazuje chybové zprávy. -t Test, testuje integritu komprese. -z Zabalí rychlou kompresí.

Tabulka 30-3 – Popis příkazu bzip2

tar – "Páskový" archivátor

Přehled

Příkaz tar je hlavní archivační program systémů Linux. Program tar kombinuje mnoho souborů do jednoho, komprimuje soubory, jestli je to požadováno. Archivátor také udržuje kritické systémové informace jako vlastnictví souborů, nastavení přístupových práv, cestu souborů, stejně jako pevné a symbolické odkazy.

Scénář

Uživatel si přeje zálohovat kompletní systém, program tar mu to dovolí v klidu a bezpečně.

Příklady

1. Následující vypsání příklad popisuje proces tvorby archivu souborů uvnitř aktuálního adresáře, prohledání archivu a přesunu archivovaných souborů.

Krok 1: Pro vytvoření archivu souborů uvnitř adresáře zadejte následující:

```
$ tar czf myarchive *
```

Krok 2: Pro shlédnutí obsahu archivu můžeme zadat následující příkaz:

```
$ tar tvf myarchive
```

2. Pro archivaci velkého množství diskového prostoru a přesunu souborů z jedné části systému do jiné, můžeme použít pár příkazů tar:

Krok 1: Zde je skvělý příklad multitaskingové práce z příkazové řádky.

```
$ tar cf - . | (cd /newplace; tar xvf -)
```

První instance příkazu tar začne číst současný adresář (.), archivovat soubory a generovat archiv (díky přepínači c neboli create) do standardního výstupu (kombinace přepínače -f). Standardní výstup je nyní směrován do dalšího programu oddělené operátorem (|). Druhý program vykoná dva příkazy v sekvenci, první příkaz cd a druhý příkaz tar. Příkaz cd přepne z aktuálního adresáře do adresáře newplace, kde jsou uchovány soubory, zatímco druhý příkaz tar extrahuje (přepínač x) archiv ze standardního vstupu a přemístí individuálně soubory do aktuálního adresáře newplace.

Souhrn

Název příkazu:	tar
Osnova:	tar c t x [-vkmx] [f archive] files...
Popis	Hlavní utilita pro archivování v systémech Linux. Umí komprimovat velké množství souborů do jednoho velkého souboru. Zachovává adresářovou strukturu. Názvy a cesta k souborům je limitována na 100 znaků.
Volby:	-f archive Název souboru nebo zařízení k archivaci. -k Nepřepíše novější soubory. -m Zachová čas souborů. -v Vypíše všechny informace o balených souborech. -z Komprimuje nebo dekomprimuje soubory. Files Seznam souborů pro archivaci. Jestliže byl zadán adresář, tar je zabalí také, ale ne v adresáři.

Tabulka 30-4 – Popis příkazu tar

Manipulace s textem / filtry

ispell – Kontrola pravopisu

Přehled

Tento program kontroluje pravopis slov v jednom či více souborech oproti slovům v jeho slovníku. Jakmile nalezne chybu, zobrazí tyto chyby a nabídne možné opravy.

Scénář

Když uživatel napíše textový soubor či více, může být ispell použit ke kontrole pravopisu jednotlivých slov v těchto souborech. Jako přídavek, může být program vyvolán s přepínačem, která uloží původní nezměněné soubory.

Příklad

Použijeme přepínač `-b` pro vytvoření záložní kopie před kontrolou souboru. Protože jsme použili `-b` vytvoříme záložní kopii `letter_draft`, který je obsahově stejný jako původní soubor.

Krok 1: Použijeme následující syntaxi pro vyvolání programu ispell pro vytvoření záložní kopie `letter_draft`. Jakmile jednou program běží, zobrazí špatně napsaná slova následované možnými opravami. Na spodku obrazovky jsou možnosti nastavení voleb programu.

```
$ ispell -b letter_template
      gretings          File: letter_draft
gretings
0: grating
1: greetings
[SP] <number> R)eply A)ccept I)nsert L)ookup U)nap Q)uit e(X)i tor ? for help.
```

Krok 2: Jakmile byly provedeny potřebné změny, uživatel může použít příkaz ls pro shlédnutí záložního souboru, který program vytvořil. Normálně má tento soubor jméno jako původní jen je připojen znak vlnovky (~).

```
$ ls
Desktop/  bhaljohn.jpgfimit/nsmail/letter-draft
Mail/     autosave/fmdictionarymail/letter_draft~
```

Souhrn

Název příkazu:	ispell
Osnova:	ispell [-bw] files
Popis	Kontroluje pravopis textu v souborech a nabízí možné opravy.
Volby:	-b Vytvoří záložní kopii původního souboru. -w [chars] Označí že [chars] jsou správné.

Tabulka 30-5 – Popis příkazu ispell

wc – Počítadlo

Přehled

Tento příkaz je používán pro zobrazení souhrnných informací o textu v souboru. Zobrazí informace o počtu slov, řádků a bytů specifického souboru.

Scénář

Uživatel potřebuje jednoduše zjistit kolik souborů nebo adresářů je v jeho domovském adresáři. Příkaz wc je tou nejjednodušší cestou.

Příklady

V následujícím příkladu příkaz ls vygeneruje seznam souborů, který je odeslán do programu grep, který zobrazí pouze jednotlivé řádky, které začínají na d (pro adresáře). Výstup je odeslán do programu wc, který sečte řádky a ohlásí počet adresářů.

Krok 1: Zadáme toto do příkazové řádky:

```
$ ls -l | grep d | wc
    25   225  1617
```

Výstupní seznam má tři pole, zleva doprava to jsou počet podadresářů (25), počet slov (225) a počítadlo bytů (1617).

Souhrn

Název příkazu:	wc
Osnova:	wc [-clw] [--bytes] [--chars] [--lines] [--words] [--help] [--version] [file...]
Popis	Wc spočte počet bytů, počet slov oddělených mezerou, počet řádek v souboru, pokud se nazdá soubor, provede toto pro všechny soubory, pokud se zadá více souborů, wc zobrazí jejich souhrnou statistiku či průměr, počítadla jsou v pořadí, řádky, slova, byty.
Volby:	-c, --bytes, --chars Vypíše pouze počítadlo bytů.. -w, --words Vypíše pouze počítadlo slov. -l, --lines Vypíše pouze počítadlo řádek. --help Vypíše info o užití a status indikující úspěšné ukončení.

Tabulka 30-6 – Příkaz wc

cat – Slučování souborů

Přehled

Tento příkaz může být použit pro tvorbu jednoduchých textových souborů nebo pro základní formátování výstupu. Užitečná funkce tohoto příkazu je spojování textových souborů do jednoho.

Scénář

Použití příkazu cat společně se symbolem přesměrování, je jednoduchá cesta pro tvorbu textových souborů. Také je vhodný pro spojení obsahu více textových souborů do jednoho.

Příklad

Použitím příkazu cat společně se symbolem přesměrování, uživatel může vytvořit nespočet souborů s textem a spojit je do jednoho souboru. Kromě toho může uživatele limitovat volný prostor pro nový soubor.

Krok 1: Použitím příkazu cat společně se symbolem přesměrování můžeme vytvořit nespočet textových souborů. Soubor se jmenuje file_1. Jakmile zmáčkne <ENTER> může uživatel zapisovat do tohoto souboru.

```
$ cat > file_1
```

Krok 2: Spojení výstupu ze dvou a více oddělených souborů může být provedeno pomocí příkazu cat následovaným názvy souborů oddělených mezerou. Volba -s nám zajistí eliminaci prázdného místa v textu mezi jednotlivými soubory.

```
$ cat -s file_1 file_2 file_3
```

Souhrn

Název příkazu:	cat
Osnova:	cat [-ns] file
Popis	Edituje či spojuje soubory.
Volby:	-n, --numbered Očísluje všechny řádky ve výsledném souboru. -s, --squeeze-blank Komprimuje výsledný soubor odstraněním prázdného místa mezi soubory.

Tabulka 30-7 – Popis příkazu cat

head a tail – Prohlížeče souborů

Přehled

Příkazy head a tail jsou nápomocné pro rychlé shlédnutí log souborů nebo jiných typů souborů. Příkaz head zobrazuje implicitně prvních 10 řádek souboru, zatímco příkaz tail zobrazuje posledních 10 řádek souboru.

Scénář

Jestliže uživatel potřebuje prohledat velké množství souborů s mnoha názvy, jsou příkazy tail a head jednoduchou cestou pro zjištění obsahu souborů.

Příklady

Pro zobrazení více řádek, musíme použít číselný argument.

Krok 1: Pro shlédnutí posledních 50 řádek log souboru stačí zadat následující:

```
$ tail -50 /var/log/messages | less
```

V tomto příkladu, je posledních 50 řádků odesláno do řádkovače less, který zobrazí obsah postupně po jedné stránce. Pamatujte si, že k souborům v /var/log může přistupovat pouze root.

Souhrn

Název příkazu:	head
Osnova:	head [#b #k #m -n#] file
Popis	Zobrazí počátek souboru.
Volby:	#b Zobrazí prvních # bajtů. #k Zobrazí prvních # kilobajtů. #m Zobrazí prvních # megabajtů. -n# Zobrazí daný počet řádek udaný znakem #.

Tabulka 30-8 – Popis příkazu head

Název příkazu:	tail
Osnova:	tail [#b #k #m -n#] file
Popis	Zobrazí konec souboru.
Volby:	#b Zobrazí posledních # bajtů. #k Zobrazí posledních # kilobajtů. #m Zobrazí posledních # megabajtů. -n# Zobrazí daný počet řádek udaný znakem #.

Tabulka 30-9 – Popis příkazu tail

grep – Vytiskne řádky shodné se vzorem

Přehled

Je to program pro hledání záznamů v souborech shodné se zadaným vzorem. Program zobrazí řádky, na kterých je shoda obsažena.

Scénář

Jestliže někdo upravují více souborů (dopisy, zprávy, poznámky, deníky a podobně) a potřebuje se vrátit k určité myšlence či části, tento příkaz mu dovoluje vyhledat tyto informace.

Příklad

Zadáním následujícího příkazu se pokusíme vyhledat řetězec Friday:

Zadáme příkaz grep s přepínačem -i abychom vynechali hledání v podadresářích.

```
$ grep -i Friday * | less
```

```
grep: Desktop: Is a directory
MIScontract.doc:\par Re-installation services preformed outsider of standard operating
hours (8:00 a.m. to 5:00 p.m. Monday through Friday) $500 per location.
clet1:revisions will be completed Friday of this week.
clet2:revisions will be completed Friday of this week.
clet3:revisions will be completed Friday of this week.
jnk:06:00 PM Central Time, Monday through Friday.
jnk2:06:00 PM Central Time, Monday through Friday.
grep: wp: Is a directory
```

Když grep nedokáže zpracovat soubor vraátá grep. Jinak hlásí název souboru a řádku, která obsahuje hledaný řetězec.

Souhrn

Název příkazu:	grep
Osnova:	grep [-f -c -s] pattern file
Popis	Hledá v souborech zadaný řetězec.
Volby:	-f name_file Použije název souboru pro prozkoumání více souborů. -c Potlačí normální výstup, zobrazí počet shodných záznamů v souboru. Při použití -v zobrazí převrácenou formu, tudíž počet neodpovídajících řádek. -s Neohlašuje chybové zprávy o neexistujících nebo nedostupných souborech.

Tabulka 30-10 – Popis příkazu **grep**

more a less – Stránkovače

Přehled

Existují dva odlišné příkazy, které jsou používány pro navigaci v ASCII dokumentech. Tyto příkazy jsou `more` a `less`. Příkaz `more` je starší příkaz s menšími možnostmi. Jestliže použijeme klávesy <UP> a <DOWN> kláves pohybujeme se po dokumentu po jednotlivých řádkách. S `less` klávesy <PAGE UP> a <PAGE DOWN> pohybujeme po celých stránkách.

Scénář

Uživatel uchovává soubor, který obsahuje velmi dlouhý a obsáhlý seznam uživatelských jmen jejich poštovních adres. Vlastník tohoto souboru si přeje hledat jméno, které bylo zadáno do souboru před několika týdny. Jak by mu mohli pomoci příkazy `less` a `more` pro efektivní prohledání souboru?

Příklady

1. S příkazem `less` mohou být díky parametru `-p` vyhledány určité fráze podle vzoru; `less` prochází soubory a označí specifické klávesy.

Krok 1: Pro hledání potřebné fráze, zadejte `less` s potřebným parametrem.

```
$ less -p linux filename
```

Tento příkaz vyhledá všechna slova `linux` v souboru `filename` a zvýrazní slova `linux` jakmile se objeví v textu.

Krok 2: Program `less` také může hledat určité heslo zadáním následujícího příkazu:

```
<ESC> then / then searchkeyword
```

`Less` poté zvýrazní všechny výskyty nového hledaného výrazu.

2. Příkaz `less` také může být použit pro prohlížení několika souborů za sebou. Příkaz bude potom vypadat takto:

`$ less filename1 filename2 filename3`

Takto se nahraje více souborů najednou. Zatímco prohlíží jeden soubor pomocí `less`, uživatel se může pohybovat mezi soubory použitím dalších příkazů:

`:n` <to go to the next file>

`:p` <to go to the previous file>

Aneb `n` pro další soubor a `p` pro předchozí.

Souhrn

Název příkazu:	Less
Osnova:	<code>Less [-p pattern] [-N] [-f] [-C] <filename></code>
Popis	Program podobný programu <code>more</code> , ale dovoluje i zpětné prohlížení souborů, před otevřením souboru ho implicitně načte a tudíž ho načítá rychleji.
Volby:	Argumenty příkazové řádky: -N Proměnná nastavující počet řádek k zobrazení. -f Neregulární soubory nebudou otevřeny. -C Vyčistí obrazovku před zobrazením výstupu. Příkazy použitelné po spuštění příkazu: SPACE Sroluje celou obrazovku. z Sroluje celou obrazovku pokud nebyl zadán přepínač -N. ? Indikuje použití příkazu na konec souboru. -I Následován kterým z parametrů příkazové řádky, změní nastavení této proměnné a zobrazí zprávů co provedená změna znamená. - Následován některým z parametrů příkazové řádky zobrazí zprávu popisující její aktuální nastavení. Nastavení nezmění.

Tabulka 30-11 – popis příkazu `less`

Název příkazu:	more
Osnova:	<code>more [options] [+/- pattern] [+linenum] filename</code>
Popis	Prochází text po stránkách.
Volby:	-num Proměnná určující velikost obrazovky. -d Zobrazí zprávy. -p Neroluje, vyčistí obrazovku a zobrazí text. -s Spojí prázdné řádky do jedné. -u Potlačí podtrhnutí. +/- Udává vyhledávaný výraz. SPACE Zobrazí další stránku textu. RETURN Zobrazí předchozí stránku textu.

Tabulka 30-12 – Popis příkazu `more`

sort – Seřazení textového souboru

Přehled

Příkaz sort je velmi užitečný, dovoluje uživateli organizovat nebo seřadit specifický soubor nebo soubory. Příkaz sort přináší tři specifické úlohy pro správu souborů: seřazení, sloučení a kontrolu zda je seřazen.

Scénář

Uživatel má seznam adres a chce ho mít abecedně seřazen, nebo dlouhý seznam čísel numericky.

Příklad

Jednoduché použití příkazu sort na textový soubor zobrazuje následující příklad:

Krok 1: Zadejte příkaz sort následovaný přepínačem -n pro numerické a abecední seřazení obsahu daného souboru. Prozkoumejte udaný příklad před a po provedení příkazu sort.

Soubor list

```
cog  
golem  
eliza  
hal  
tyler  
3  
4  
1  
8
```

```
$ sort -n list  
Result of sorted list:
```

```
cog  
eliza  
golem  
hal  
tyler  
1  
3  
4  
8
```

Krok 2: A nyní můžeme jednoduše seřadit obsah dvou souborů do třetího.

```
$ sort -n file1 file2 > file3
```

Souhrn

Název příkazu:	Sort
Osnova:	Sort [options] [string]
Popis	Poskytuje seřazení obsahu souboru.
Volby:	<ul style="list-style-type: none">-b Vynechává prázdné řádky.-c Neřadí soubor, pokud je již seřazen.-d Dovoluje řadit pouze 36 znaků (A-Z, 0-9).-m Spojuje seřazené soubory, podobné k příkazu cat.-n Dovoluje seřazení čísel.-g Dovoluje seřazení čísel s desetinou čárkou.

Tabulka 30-13 – Popis příkazu **sort**