KAPITOLA 2

ÚVOD DO OPERAČNÉHO SYSTÉMU LINUX

Operačný systém Linux, podobne ako iné operačný systémy vychádzajúce z OS Unix™ poskytuje používateľovi okrem grafického používateľského rozhrania aj textové rozhranie, ktoré tvorí jeho základnú časť. Na rozdiel od monolitickej architektúry OS Windows je rozhranie OS Linux založené na malých, autonómnych programoch, ktoré je možné vzájomne kombinovať a tým dosiahnuť veľkú variabilitu pri práci s operačným systémom. V nasledujúcej kapitole popíšeme základy práce s operačným systémom tak, aby čitateľ mohol bez problémov pracovať v prostredí OS Linux.

2.1 Úvod do operačného systému Linux

Operačný systém Linux bol od svojho vzniku koncipovaný ako viacpoužívateľský operačný systém. Automaticky sa teda predpokladá súčasný prístup viacerých používateľov k jednému počítaču.

Používateľ pristupuje k počítaču prostredníctvom terminálu, ktorý je možné považovať za „neinteligentné zariadenie“ pozostávajúce z obrazovky a klávesnice. Režim činnosti počítača, kedy súčasne obsluhuje viacero používateľov zdieľajúcich rovnaký disk, tlačiareň a ďalšie zariadenia, vyžaduje vytvorenie virtuálneho prostredia, v ktorom má používateľ pocit, že nie je obmedzovaný žiadnym iným používateľom, s ktorým zdieľa ten istý počítač. V praxi to teda znamená, že môže napríklad používať tlačiareň bez toho, aby si nechával potvrdiť, že žiaden iný používateľ nebude na danej tlačiarni tlačiť.

Základnou filozofiou pri návrhu prvého pôvodné operačného systému Unix, z ktorého Linux vychádza, bola prenositeľnosť. Táto myšlienka ovplyvnila aj celkový návrh operačného systému a aplikácií, kde operačný systém poskytuje programátorovi programové celky, ktoré môže pri písaní aplikácií využívať. Z tohto pohľadu je možné rozdeliť operačný systém Linux na:

- jadro operačného systému
- základné podporné programy
- nástroje programátora

Medzi ďalšie črty operačného systému, ktoré sú prítomne už v operačných systémoch Unix sú:

- hierarchická štruktúra adresárov,
- práca s procesmi ako s elementmi zmien v dátovej základni,
- podpora volaní jadra pre oddelenie používateľa od technického vybavenia,
- textovo orientovaná interaktívna práca viacerých používateľov prostredníctvom terminálov.

OS Linux je interaktívny operačný systém. Každá interakcia používateľa a operačného systému je zvýhodnená oproti bežiacim úlohám. Pri interakcii sa používa určitý komunikačný jazyk, ktorý sa vo všeobecnosti nazýva **shell** resp. **príkazový interpreter**.

2.1.1 Používateľské účty

Na podporu mechanizmov definovaných v predchádzajúcej kapitole je použitý systém používateľských účtov. Správca (administrátor) systému vytvorí každému používateľovi jeho účet. Účet jednoznačne definuje používateľa a jeho pracovné prostredie. Väčšina informácii je uložených v súbore `/etc/passwd`. Používateľský účet obsahuje nasledujúce položky:

- **používateľské meno** (angl. login name) - je to skupina spravidla 3 až 8 znakov, ktorá jednoznačne identifikuje používateľa v systéme. Slúži na prihlásenie používateľa do systému,
- **heslo** (angl. password) - v počítači sa uchováva v zakódovanej podobe buď v súbore `/etc/passwd` alebo `/etc/shadow`. Druhý prípad je bezpečnejší, keďže údaje v súbore `/etc/shadow` môže čítať iba správca systému,
- **používateľské číslo UID** (angl. user identification) - ide o jednoznačný identifikátor používateľa pre operačný systém. Operačný systém využíva pre identifikáciu práve UID, nie používateľské meno,
- **číslo primárnej skupiny GID** (angl. primary group identification) - každý používateľ je členom aspoň jednej používateľskej skupiny. Pri používateľskom účte sa zaznamenáva iba primárna používateľská skupina (skupina, ktorá je nastavená po prihlásení do systému). Zoznam používateľských skupín je v súbore `/etc/group`,
- **doplňujúce používateľské údaje** - napr. meno, priezvisko, adresa a pod.,
- **domovský adresár** (angl. home directory) - absolútna cesta k adresáru, ktorý sa po prihlásení nastaví ako domovský adresár používateľa,
- **príkazový interpreter** (angl. shell) - cesta k súboru, ktorý sa spustí ako interpreter príkazov po prihlásení používateľa do systému.

2.1.2 Administrátor

V každom operačnom systéme unixového typu existuje práve jeden privilegovaný používateľ, ktorému sa nekontrolujú jeho prístupové práva. Nazýva sa superpoužívateľ a je identifikovaný používateľským menom **root** s používateľským číslom 0. Pod týmto používateľom vykonáva administrátor systému jeho správu. Účet by nemal byť používaný na bežnú používateľskú prácu, pretože aj malou, neúmyselnou chybou je možné nevratne poškodiť operačný systém.

Z dôvodu neobmedzených prístupových práv superpoužívateľa je nutná ochrana účtu pred neautorizovaným prístupom.

2.1.3 Prihlásenie do systému, odhlásenie zo systému

Základným predpokladom pre vstup používateľa do systému je jeho autorizácia. Autorizácia používateľa prebieha interaktívne na základe jeho používateľského mena a hesla. Výzva na zadanie používateľského mena je na obrazovke zobrazená v tvare

```
login as:
```

Po zadaní používateľského mena je používateľ požiadaný o zadanie používateľského hesla

```
Password:
```

a systém znova čaká na reakciu používateľa. Pokiaľ prebehne prihlásenie korektne, objaví sa na obrazovke identifikátor promptu:

```
zabovsky@frios:~$
```

Od tohto okamihu je používateľ korektne prihlásený do systému. Pri prihlásení do systému ako superpoužívateľ je znak výzvy pre komunikáciu $ nahradený znakom #.

Heslo si používateľ mení príkazom `passwd`. Po zadaní príkazu musí používateľ najskôr zadať svoje pôvodné heslo, ktorým sa prihlásil do systému a následne zadá nové heslo (dvakrát kvôli kontrole, či nedošlo k „preklepu“).

Odhlásenie zo systému (ukončenie komunikácie so systémom) je možné pomocou príkazu `exit` alebo stlačením `Ctrl-d` (súčasné stlačenie `Ctrl` a `d`).

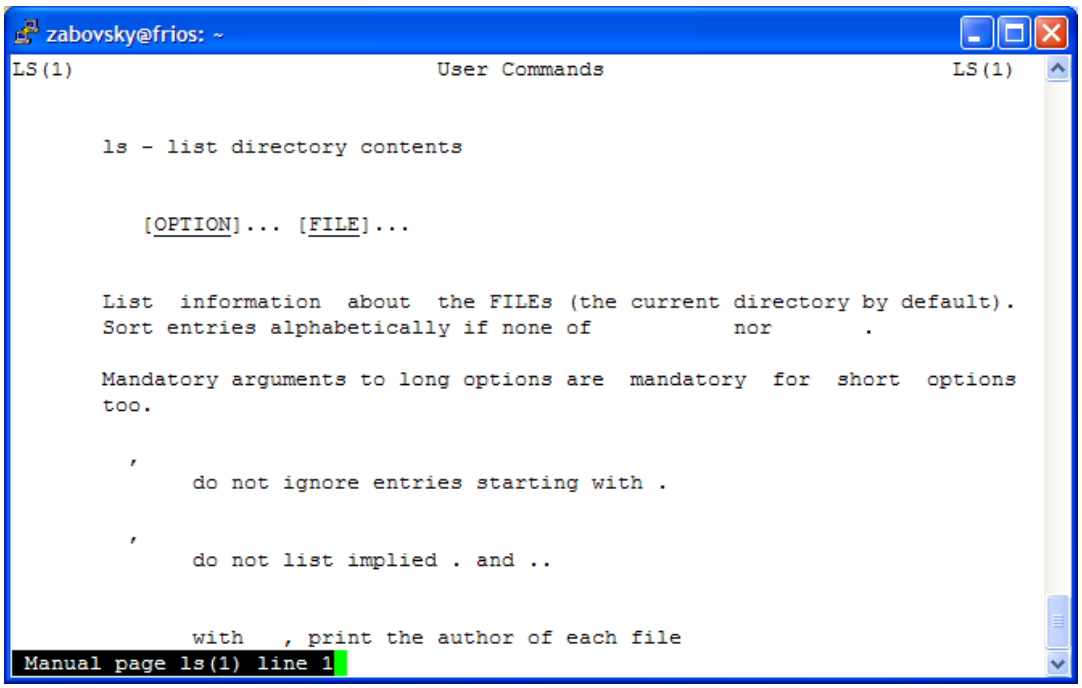
2.1.4 Dokumentácia

Dokumentácia pre systémy Linux je založená na sade tzv. manuálových stránok - referencií. Tieto sú rozdelené do 8 sekcií nasledovne:

1. príkazy používateľskej úrovne (angl. commands)
2. volania jadra (angl. system calls)
3. podprogramy (angl. subroutines)

4. formáty súborov (angl. file formats)
5. rôzne (angl. miscellaneous)
6. hry (angl. games)
7. špeciálne súbory - rozhrania pre technické vybavenie (angl. special files)
8. príkazy privilegovaného používateľa (angl. superuser commands)

Každý diel má referencie zoradené podľa abecedy. Referencie sú formátované špeciálnym spôsobom, zvýraznené sú časti NAME pre označenia názvu referencie, formát použitia SYNOPSIS, popis DESCRIPTION a ďalšie odkazy SEE ALSO (obrázok 2.1).



Obrázok 2.1: Manuálová stránka príkazu ls

2.1.5 Vstup a výstup príkazu

Každý proces (procesom je aj príkaz) má v Linuxe automaticky prístup k trom zvláštnym súborom:

- štandardnému vstupu
- štandardnému výstupu
- štandardnému chybovému súboru

Tieto súbory existujú preto, aby bolo možné vzájomne prepájať jednotlivé príkazy. Ak teda prvý príkaz posiela výstup na štandardný vstup druhého príkazu ide o prepojené príkazy. Prepojenie príkazov je možné pomocou znaku `|`.

Majme nasledujúci príklad: je potrebné vypísať informácie o všetkých súboroch v danom adresári, ktoré obsahujú v názve slovo profile. Riešenie: na vypísanie obsahu použijeme príkaz `ls`. Z daného výpisu potrebujeme vybrať práve tie súbory, ktoré obsahujú v názve slovo profile. Na vytvorenie filtra použijeme príkaz `grep`. Výsledok je potom volanie:

```
zabovsky@frios:~$ ls -lsa | grep profile
4 -rw-r--r-- 1 zabovsky zabovsky  414 Jul 13 13:15 .bash_profile
```

Ďalší príklad súvisí s príkazom `ls`. Príkaz `ls -lsa` vracia veľmi dlhý zoznam adresárov a súborov (viac ako jedna obrazovka). Je potrebné „zastaviť“ výpis vždy po vypísaní práve jednej obrazovky. Riešenie: na stránkovanie výpisu použijeme príkaz `more`, ktorý bude prepojený s príkazom `ls`.

```
zabovsky@frios:~$ ls -lsa /bin/ | more
total 3404
  4 drwxr-xr-x  2 root root   4096 Jul 11 12:35 .
  4 drwxr-xr-x 23 root root   4096 Jul 11 12:34 ..
  4 -rwxr-xr-x  1 root root   3248 Feb 21  2007 arch
668 -rwxr-xr-x  1 root root 677184 Dec 11  2006 bash
 28 -rwxr-xr-x  3 root root  25304 Aug 25  2006 bunzip2
 28 -rwxr-xr-x  3 root root  25304 Aug 25  2006 bzcat
  4 -rwxr-xr-x  2 root root   2105 Aug 25  2006 bzcmp
  4 -rwxr-xr-x  2 root root   2105 Aug 25  2006 bzdiff
  4 -rwxr-xr-x  3 root root   3642 Aug 25  2006 bzegrep
  8 -rwxr-xr-x  1 root root   4878 Aug 25  2006 bzexe
  4 -rwxr-xr-x  3 root root   3642 Aug 25  2006 bzfgrep
  4 -rwxr-xr-x  3 root root   3642 Aug 25  2006 bzgrep
 28 -rwxr-xr-x  3 root root  25304 Aug 25  2006 bzip2
 12 -rwxr-xr-x  1 root root   8308 Aug 25  2006 bzip2recover
  4 -rwxr-xr-x  2 root root   1297 Aug 25  2006 bzless
  4 -rwxr-xr-x  2 root root   1297 Aug 25  2006 bzmore
 20 -rwxr-xr-x  1 root root  17156 Jan 30  2007 cat
 36 -rwxr-xr-x  1 root root  33396 Jan 30  2007 chgrp
 32 -rwxr-xr-x  1 root root  30384 Jan 30  2007 chmod
 36 -rwxr-xr-x  1 root root  35356 Jan 30  2007 chown
 60 -rwxr-xr-x  1 root root  56412 Jan 30  2007 cp
 96 -rwxr-xr-x  1 root root  93944 Aug  4  2006 cpio
--More—
```

Pred vykonaním príkazu je možné určiť, čo sa má urobiť so vstupom a výstupom príkazu. Ako príklad bolo uvedené spojenie príkazov. Operátor presmerovania slúži potom na čítanie vstupu zo súboru, zápis výsledku do súboru a pod. V ďalšom texte uvedieme príklad presmerovanie vstupu a výstupu procesu.

Presmerovanie vstupu má vo všeobecnosti tvar `[n] < subor`. Operátor presmerovania < spôsobí, že súbor `subor` sa otvorí pre čítanie popisovačom súboru `n` (angl. file descriptor). Ak nie je číslo `n` uvedené, bude sa čítať štandardný vstup (descriptor 0). Súbor `subor` musí existovať

Presmerovanie výstupu má tvar `[n] > subor`. Operátor presmerovania < spôsobí, že súbor `subor` sa otvorí pre zápis popisovačom súboru `n`. Ak nie je číslo `n` uvedené, bude sa zapisovať na štandardný výstup (descriptor 1). Ak súbor `subor` neexistuje, vytvorí sa, ak existuje, prepíše sa.

Princíp fungovania presmerovania výstupu si opäť ukážeme na príklade. Predstavme si, že chceme uložiť výsledok príkazu ls -lsa do súboru ls_vystup pre ďalšie spracovanie. Jednoduchou cestou je práve presmerovanie výstupu príkazu `ls` tak, ako je to uvedené v nasledujúcom výpise:

```
zabovsky@frios:~$ ls -lsa > ls_vystup
zabovsky@frios:~$ cat ls_vystup
total 48
4 drwxr-x---   5 zabovsky zabovsky 4096 Sep  2 15:10 .
8 drwxrwsr-x 353 root     staff    8192 Jul 13 13:15 ..
8 -rw-------   1 zabovsky zabovsky 5078 Sep  2 14:35
.bash_history
4 -rw-r--r--   1 zabovsky zabovsky  227 Jul 13 13:25
.bash_logout
4 -rw-r--r--   1 zabovsky zabovsky  414 Jul 13 13:15
.bash_profile
4 -rw-r--r--   1 zabovsky zabovsky 2250 Aug 30 17:17 .bashrc
4 -rw-------   1 zabovsky zabovsky   35 Sep  2 13:27 .lesshst
4 drwxr-xr-x   3 zabovsky zabovsky 4096 Sep  2 13:34 .mc
4 drwx------   2 zabovsky zabovsky 4096 Aug 31 10:55 .ssh
0 -rw-r--r--   1 zabovsky zabovsky    0 Sep  2 15:10 ls_vystup
4 drwxr-xr-x   3 zabovsky zabovsky 4096 Aug 24 15:29 priklady
```

Podobne môžeme postupovať pri riešení ďalšieho problému. Máme dva adresáre `dir1` a `dir2`. Prvý adresár neexistuje, druhý existuje. Chceme vypísať obsah adresárov do súboru `obsah` a prípadné chyby do súboru `chyby.` Skúsime použiť príkaz `ls` pre dva adresáre:

```
zabovsky@frios:~$ ls dir1 dir2
ls: dir1: No such file or directory
dir2:
subor1  subor2
```

Vidíme, že prvá správa je chybová, druhá je výpis obsahu adresára. Presmerujeme teda výstup príkazu (descriptor 1) do súboru `obsah` a chybovú správu (descriptor 2) do súboru `chyby`:

```
zabovsky@frios:~$ ls dir1 dir2 > obsah 2> chyby
```

```
zabovsky@frios:~$ cat obsah
dir2:
subor1
subor2
zabovsky@frios:~$ cat chyby
ls: dir1: No such file or directory
```

Vidíme, že obsah súboru `obsah` zodpovedá obsahu adresára a súbor `chyby` obsahuje chybové správy celého procesu. Ak chceme ignorovať prípadné chyby, najjednoduchší spôsob je zaslať ich zariadeniu `/dev/null`, ktorého funkciou je „ničenie“ vstupných dát:

```
zabovsky@frios:~$ ls dir1 dir2 > obsah 2> /dev/null
zabovsky@frios:~$ cat obsah
dir2:
subor1
subor2
```

V niektorých prípadoch chceme výstup pripojiť k súboru, ktorý už existuje. V takomto prípade musíme pri presmerovaní použiť zdvojený znak `>` v tvare `>>`. Príklad použitia potom vyzerá napríklad takto:

```
zabovsky@frios:~$ echo "subor1" > subor1
zabovsky@frios:~$ echo "subor2" > subor2
zabovsky@frios:~$ cat subor1
subor1
zabovsky@frios:~$ cat subor2
subor2
zabovsky@frios:~$ cat subor2 >> subor1
zabovsky@frios:~$ cat subor1
subor1
subor2
zabovsky@frios:~$ cat subor2
subor2
```

2.2 Procesy

Základnou vlastnosťou operačných systémov Linux je schopnosť spúšťať viacero programov súčasne. Implementácia celého mechanizmu vychádza z oddelenia programu (spustiteľná podoba uložená na disku) od konkrétnej, bežiacej inštancie daného programu - **procesu**. Z toho vyplýva, že nie je možné pomenovať procesy (napr. pomocou mena programu), ale je potrebné ich očíslovať.

Každý proces je jednoznačne identifikovaný číslom procesu (PID – angl. process ID). Proces s PID 0 je vytvorený pri zavádzaní systému a nič nevykonáva. Proces s PID 1 sa nazýva `init` a je predkom všetkých ďalších vytvorených procesov v systéme. Je teda zrejmé, že pre procesy existuje hierarchická štruktúra.

Ku každému procesu existuje **reálne UID** identifikujúce používateľa, ktorý je zodpovedný za bežiaci proces a **efektívne UID**, ktoré sa používa pre priradenie vlastníctva novo vytváraným súborom, kontrole prístupových práv a pod.

2.3 Systém súborov

Vo svete operačných systémov Unix/Linux existuje veľké množstvo fyzických spôsobov ukladania dát na diskoch. Pre BSD systémy je to súborový systém 4.2, pre UNIXové systémy firmy Sun ufs pre Linux je to najmä ext2fs prípadne ďalšie žurnálové súborové systémy. V ďalšom texte sa budeme zaoberať prácou so súbormi a adresármi, konkrétnu fyzickú implementáciu, pokiaľ to bude možné, zanedbáme.

Operačné systémy typu Linux pre urýchlenie práce s diskovými zariadeniami nezapisujú dáta okamžite, ale využívajú systém vyrovnávacej diskovej pamäte (cache), ktorá sa zapisuje na disk spravidla každých 30 sekúnd. Preto je potrebné pamätať na synchronizáciu v prípade odpájania súborových systémov.

2.3.1 Súbory a adresáre

Rozdiely pri práci so súbormi a adresármi sú oproti systémom typu DOS/Windows nasledovné:

- rozlišuje sa medzi malými a veľkými písmenami
- meno môže obsahovať ľubovoľné znaky okrem znaku NULL (binárna nula)
- súbory začínajúce bodkou sú obdobou skrytých súborov (napríklad .profile)
- adresáre sa v zápise cesty oddeľujú obyčajným lomítkom (znak `/`)
- absolútnu cestu definujeme od koreňového adresára (napríklad `/home/uzivatel/dir1/file1.c`), relatívna cestu je obdobou zápisu vo Windows (`dir1/file1.c`)
- každý adresár obsahuje automaticky dve položky - `.` predstavuje aktuálny adresár, `..` nadradený adresár (prechod do nadradeného adresáru sa teda vykoná príkazom `cd ..`)

Najbežnejšie príkazy pre prácu s adresármi a súbormi sú:

- `pwd` – vypíše cestu k aktuálnemu adresáru – pozíciu v súborovom systéme
- `cd <adresar>` - zmenní aktuálny pracovný adresár
- `mkdir <adresar>` - vytvorí adresár
- `rmdir <adresar>` - zmaže adresár
- `touch <subor>` - vytvorí nový súbor

- `rm <subor>` - vymaže súbor
- `cp <zdroj> <ciel>` - skopíruje súbor
- `mv <zdroj> <ciel>` - presunie súbor

2.3.2 Odkazy

Systém odkazov (angl. links) využíva zvláštny spôsob oddeleného uchovávania obsahu a názvov súborov. Znamená to teda, že na rovnaký súbor (fyzické dáta) môže odkazovať viacero mien. Rozlišujeme medzi **tvrdými odkazmi** a **symbolickými (mäkkými) odkazmi**. Pre tvrdé odkazy platí:

- je ich možné vytvárať iba v rámci jedného fyzického systému súborov
- novo vytvorený tvrdý odkaz patrí vlastníkovi súboru bez ohľadu na používateľa, ktorý tvrdý odkaz vytváral
- nie je možné vytvárať tvrdé odkazy na adresáre (nebolo by možné rozlíšiť, kam má ukazovať súbor `..`)
- na vytvorenie tvrdého odkazu sa používa príkaz `ln <subor> <odkaz>`

Symbolické odkazy sa používajú omnoho častejšie ako tvrdé odkazy. Výrazne zjednodušujú a sprehľadňujú prácu so súbormi, pričom vytváranie symbolických odkazov je povolené ľubovoľnému používateľovi a tiež nie je nutné, aby bol odkazovaný objekt fyzicky dostupný. Samotná kontrola prebehne až v okamihu použitia. Komplexný príklad vidieť na nasledujúcom výpise:

```
zabovsky@frios:~$ mkdir pokus
zabovsky@frios:~$ cd pokus
zabovsky@frios:~/pokus$ echo "subor" > subor
zabovsky@frios:~/pokus$ cat subor
subor
zabovsky@frios:~/pokus$ ls -la
total 12
drwxr-xr-x 2 zabovsky zabovsky 4096 Sep  2 16:15 .
drwxr-x--- 6 zabovsky zabovsky 4096 Sep  2 16:15 ..
-rw-r--r-- 1 zabovsky zabovsky    6 Sep  2 16:15 subor
zabovsky@frios:~/pokus$ ln subor hard_link
zabovsky@frios:~/pokus$ ls -la
total 16
drwxr-xr-x 2 zabovsky zabovsky 4096 Sep  2 16:16 .
drwxr-x--- 6 zabovsky zabovsky 4096 Sep  2 16:15 ..
-rw-r--r-- 2 zabovsky zabovsky    6 Sep  2 16:15 hard_link
-rw-r--r-- 2 zabovsky zabovsky    6 Sep  2 16:15 subor
zabovsky@frios:~/pokus$ ln -s subor soft_link
zabovsky@frios:~/pokus$ ls -la
total 16
drwxr-xr-x 2 zabovsky zabovsky 4096 Sep  2 16:16 .
drwxr-x--- 6 zabovsky zabovsky 4096 Sep  2 16:15 ..
```

```
-rw-r--r-- 2 zabovsky zabovsky    6 Sep  2 16:15 hard_link
lrwxrwxrwx 1 zabovsky zabovsky    5 Sep  2 16:16 soft_link ->
subor
-rw-r--r-- 2 zabovsky zabovsky    6 Sep  2 16:15 subor
zabovsky@frios:~/pokus$ cat hard_link
subor
zabovsky@frios:~/pokus$ cat soft_link
subor
zabovsky@frios:~/pokus$ rm hard_link
zabovsky@frios:~/pokus$ ls -la
total 12
drwxr-xr-x 2 zabovsky zabovsky 4096 Sep  2 16:18 .
drwxr-x--- 6 zabovsky zabovsky 4096 Sep  2 16:15 ..
lrwxrwxrwx 1 zabovsky zabovsky    5 Sep  2 16:16 soft_link ->
subor
-rw-r--r-- 1 zabovsky zabovsky    6 Sep  2 16:15 subor
zabovsky@frios:~/pokus$ rm soft_link
zabovsky@frios:~/pokus$ ls -la
total 12
drwxr-xr-x 2 zabovsky zabovsky 4096 Sep  2 16:18 .
drwxr-x--- 6 zabovsky zabovsky 4096 Sep  2 16:15 ..
-rw-r--r-- 1 zabovsky zabovsky    6 Sep  2 16:15 subor
```

2.3.3 Špeciálne súbory

Špeciálny súbor predstavuje rozhranie ovládača zariadení. Jeho prostredníctvom sa zjednocuje obsluha súborov s obsluhou zariadení na rovnakú množinu operácií. Špeciálne súbory je možné nájsť v adresári `/dev/`:

```
zabovsky@frios:~$ ls -la /dev/ | more
total 100
drwxr-xr-x 11 root root       24576 Sep  2 06:47 .
drwxr-xr-x 23 root root        4096 Jul 11 12:34 ..
lrwxrwxrwx  1 root root          13 Jul  3  2006 MAKEDEV ->
/sbin/MAKEDEV
crw-rw----  1 root video    10, 175 Jul  3  2006 agpgart
drwxr-xr-x  2 root root        4096 Jul  3  2006 ataraid
crw-------  1 root root     10,   3 Feb 26  2005 atibm
crw-rw----  1 root audio    14,   4 Feb 26  2005 audio
crw-rw----  1 root audio    14,  20 Feb 26  2005 audio1
crw-rw----  1 root audio    14,  36 Feb 26  2005 audio2
crw-rw----  1 root audio    14,  52 Feb 26  2005 audio3
crw-rw----  1 root audio    14,   7 Feb 26  2005 audioctl
brw-rw----  1 root cdrom    29,   0 Feb 26  2005 aztcd0
brw-rw----  1 root cdrom    41,   0 Feb 26  2005 bpcd
drwxr-xr-x  2 root root       12288 Jul  3  2006 cciss
lrwxrwxrwx  1 root root           8 Jul  3  2006 cdrom ->
/dev/hdc
brw-rw----  1 root cdrom    24,   0 Feb 26  2005 cdu535
brw-rw----  1 root cdrom    30,   0 Feb 26  2005 cm206cd0
crw-------  1 root tty       5,   1 Jul 12 18:58 console
```

```
lrwxrwxrwx  1 root root          11 May 25 13:31 core ->
/proc/kcore
crw-rw----  1 root disk     10, 252 Feb 26  2005 dac960_gam
crw-rw----  1 root disk    151,   0 Feb 26  2005 dpti1
crw-rw----  1 root disk    151,   1 Feb 26  2005 dpti2
--More—
```

2.3.4 Štruktúra adresárového stromu

Pri práci s operačným systémom Linux je potrebné sa vedieť dobre orientovať v adresárovom strome. Základná štruktúra, ktorá vychádza z adresára `/` je štandardná. Jednotlivé verzie operačného systému sa môžu líšiť ďalšou štruktúrou, ktorá závisí na distribúcii operačného systému. Základné adresáre však nájdete vo väčšine distribúcií, rovnako ako aj v iných operačných systémoch typu Unix. Základné adresáre potom najú nasledovné funkcie:

/bin - adresár so základnými príkazmi

/boot - adresár s programami nevyhnutne potrebnými pre zavedenie systému

/dev - adresár špeciálnych súborov zariadení

/etc - adresár konfiguračných súborov systému

/home - adresár domovských adresárov používateľov

/lib - adresár zdieľaných knižníc a modulov jadra

/mnt - pomocný adresár pre dočasné pripájanie iných súborových systémov

/proc - virtuálny systém súborov určený na čítanie dátových štruktúr jadra systému a obrazov procesov

/sbin - adresár so systémovými súbormi potrebnými pre zavedenie a základnú údržbu systému

/tmp - verejný adresár pre pomocné a dočasné súbory

/usr - adresár so súbormi, ktoré sú typicky nemenné, nie sú potrebné v okamihu zavedenia systému a je ich možné zdieľať s inými počítačmi s rovnakým systémom

/var - adresár pracovných, administratívnych a iných súborov systému

2.3.5 Prístupové práva

Operačné systémy typu UNIX/Linux regulujú prístupové práva k adresárom a súborom prostredníctvom mechanizmu prístupových práv. Tieto je možné zvlášť špecifikovať pre vlastníka, skupinu a ostatných používateľov.

Práva je možné skontrolovať v prvom stĺpci získanom pri výpise príkazom `ls –la`. Pre vlastníka, skupinu a ostatných používateľov sú špecifikované trojice s písmenami rwx

(angl. read, write, execute). Tretí a štvrtý súbor definujú používateľa a skupinu, ktorým súbor patrí. Súbor `subor` uvedený vo výpise má potom nastavené práva pre čítanie a zápis pre vlastníka, právo čítania pre skupinu a ostatných používateľov.

```
zabovsky@frios:~/pokus$ ls -la
total 12
drwxr-xr-x 2 zabovsky zabovsky 4096 Sep  2 16:18 .
drwxr-x--- 6 zabovsky zabovsky 4096 Sep  2 16:15 ..
-rw-r--r-- 1 zabovsky zabovsky    6 Sep  2 16:15 subor
```

Pre súbor sa operácie definujú nasledovne:

- **r** súbor je možné čítať
- **w** do súboru je možné zapisovať
- **x** súbor je možné spustiť (vykonať)

Pre adresár je význam nasledovný:

- **r** pre adresár je možné vypísať obsah
- **w** do adresára je povolené zapisovať
- **x** do adresára je možné vstúpiť

Implicitné prístupové práva pre vytvárané súbory je možné definovať príkazom `umask`. Zmena prístupových práv je možná príkazom `chmod`, zmena vlastníka a skupiny príkazom `chown`.

Najlepšie je možné ukázať prácu s prístupovými právami na príklade. Majme nasledujúci výpis:

```
zabovsky@frios:~/pokus$ l
total 16
drwxr-xr-x 3 zabovsky zabovsky 4096 Sep  2 17:15 .
drwxr-x--- 6 zabovsky zabovsky 4096 Sep  2 16:15 ..
drwxr-xr-x 2 zabovsky zabovsky 4096 Sep  2 17:15 adresar
-rw-r--r-- 1 zabovsky zabovsky    6 Sep  2 16:15 subor
```

Chceme pridať právo zápisu (`+w` – plus write) používateľom v skupine a ostaným používateľom (`go` – group, others). Príkaz na zmenu bude vyzerať:

```
zabovsky@frios:~/pokus$ chmod go+w subor
zabovsky@frios:~/pokus$ ls -la
total 16
drwxr-xr-x 3 zabovsky zabovsky 4096 Sep  2 17:15 .
drwxr-x--- 6 zabovsky zabovsky 4096 Sep  2 16:15 ..
```

```
drwxr-xr-x 2 zabovsky zabovsky 4096 Sep  2 17:15 adresar
-rw-rw-rw- 1 zabovsky zabovsky    6 Sep  2 16:15 subor
```

Odobratie práva zápisu prebehne obdobne:

```
zabovsky@frios:~/pokus$ chmod go-w subor
zabovsky@frios:~/pokus$ ls -la
total 16
drwxr-xr-x 3 zabovsky zabovsky 4096 Sep  2 17:15 .
drwxr-x--- 6 zabovsky zabovsky 4096 Sep  2 16:15 ..
drwxr-xr-x 2 zabovsky zabovsky 4096 Sep  2 17:15 adresar
-rw-r--r-- 1 zabovsky zabovsky    6 Sep  2 16:15 subor
```

Prístupové práva je možné definovať aj číselne, kedy číslo vyjadruje hodnotu špecifikovanú každý typ vlastníka trojicou bitov:

```
zabovsky@frios:~/pokus$ chmod 755 subor
zabovsky@frios:~/pokus$ ls -la
total 16
drwxr-xr-x 3 zabovsky zabovsky 4096 Sep  2 17:15 .
drwxr-x--- 6 zabovsky zabovsky 4096 Sep  2 16:15 ..
drwxr-xr-x 2 zabovsky zabovsky 4096 Sep  2 17:15 adresar
-rwxr-xr-x 1 zabovsky zabovsky    6 Sep  2 16:15 subor
```

Zmena vlastníka a skupiny vyzerá nasledovne:

```
zabovsky@frios:~/pokus$ chown zabovsky.KI subor
zabovsky@frios:~/pokus$ l
total 16
drwxr-xr-x 3 zabovsky zabovsky 4096 Sep  2 17:15 .
drwxr-x--- 6 zabovsky zabovsky 4096 Sep  2 16:15 ..
drwxr-xr-x 2 zabovsky zabovsky 4096 Sep  2 17:15 adresar
-rwxr-xr-x 1 zabovsky KI          6 Sep  2 16:15 subor
```

Podrobnejšie informácie o príkazoch `chmod` a `chown` môžete nájsť v manuálových stránkach. Používanie príkazov potom bude viac-menej závisieť na vašich zvyklostiach, je však dobré poznať aj iné spôsoby použitia pre ľahšiu orientáciu v problematike.

2.3.6 Prehľad príkazov pre prácu so súbormi

Pre ľahšiu orientáciu pri práci so súbormi a adresármi uvedieme jednoduchý prehľad najpoužívanejších príkazov zoradených podľa účelu použitia:

- príkaz pre čítanie obsahu adresára
  - **ls** - výpis adresára
- príkazy pre kopírovanie, presun, rušenie súborov a pod.
  - **cp** - kopírovanie súborov a adresárov

- **mv** - presun resp. premenovanie súborov a adresárov
- **ln** - tvorba odkazov (liniek)
- **rm** - rušenie súborov a adresárov
- **touch** - zmena časovej informácie súboru (sekundárny efekt je vytvorenie súboru)

- príkazy pre zmenu vlastníctva a prístupových práv
  - **chgrp** - zmena skupiny vlastníkov
  - **chown** - zmena vlastníka a skupiny vlastníkov
  - **chmod** - zmena prístupových práv
- príkazy pre vytvorenie, rušenie a zistenie veľkosti adresárov
  - **mkdir** - vytvorenie adresára
  - **rmdir** - zrušenie prázdneho adresára
  - **du** - zistenie veľkosti adresára
- príkazy pre porovnávanie súborov
  - **cmp** - porovná dva súbory
  - **diff** - hľadá rozdielne a zhodné časti súborov
  - **patch** - inverzný príkaz k diff – pomocou súboru zmien z príkazu diff vytvorí nový súbor
  - **diff3** - porovnáva tri textové súbory
  - **sdiif** - interaktívne zlučuje dva súbory do jedného
- príkazy pre prehľadávanie adresárov
  - **find** - prechádza stromovú štruktúru adresárov a s vybratými položkami vykonáva špecifikovanú akciu
- ďalšie príkazy
  - **df** - zistenie voľnej kapacity systému súborov
  - **mkfifo** - vytvorenie rúry (angl. pipe)
  - **mknod** - vytvorenie špeciálneho súboru
  - **sync** - uloženie vyrovnávacej pamäte na disk

2.4 Ďalšie príkazy shelu

Okrem príkazov spomenutých v časti o súborovom systéme je potrebné poznať ešte aspoň niekoľko ďalších príkazov príkazového interpretra. Príkazy opäť uvedieme zoradené podľa účelu použitia tak, ako to bolo možné vidieť v predchádzajúcej kapitole:

- príkazy pre výstup textu
  - **echo** - vypíše argumenty príkazu na štandardný výstup
  - **printf** - sformátované údaje vypíše na štandardný výstup
- príkaz pre presmerovanie
  - **tee** - presmeruje vstup do viacerých súborov
- príkazy pre manipuláciu s menami súborov
  - **basename** - z cesty vypíše iba meno
  - **dirname** - z cesty vypíše iba adresáre
- príkazy pre výpis a nastavenie pracovného prostredia
  - **pwd** - vypíše aktuálny pracovný adresár
  - **stty** – vypíše, prípadne zmení charakteristiky terminálu
  - **printenv** - vypíše aktuálne premenné prostredia
  - **tty** - vypíše meno terminálu
- informácie o používateľoch
  - **id** - vypíše UID a GID
  - **logname** - vypíše prihlasovacie meno používateľa
  - **whoami** - vypíše informácie o aktuálnom používateľovi
  - **groups** - vypíše skupiny, do ktorých patrí používateľ
  - **users** - vypíše mená prihlásených používateľov
  - **who** - vypíše informácie o prihlásených používateľoch
- systémové informácie
  - **date** - vypíše aktuálny dátum a čas
  - **uname** - vypíše informácie o systéme
  - **hostname** - vypíše alebo nastaví meno počítača
- iné príkazy na spustenie procesov
  - **env** - spustí proces v zmenenom prostredí

- **nice** - spustí proces so zmenenou prioritou
- **nohup** - imunizuje proces pred signálom SIGHUP
- **su** - spustí proces pod právami iného používateľa
- **sleep** - počká zadaný počet sekúnd

2.5 Nástroje pre archiváciu a kompresiu dát

Základným nástrojom používaním na archiváciu v operačných systémoch Unix/Linux je program `tar`. Príkaz `tar` (angl. Tape ARchiver) je archivačným programom, ktorý sa v praxi používa nielen na archiváciu súborov. Archivuje jednotlivé súbory alebo celé adresárové stromy do jedného súboru - archívu. Archív je možné vytvárať priamo na magnetickej páske (pre účely zálohovania), častejšie sa však archív vytvára ako klasický súbor.

Vytvorenie archívu so špecifikovaným menom s obsahom aktuálneho adresára vykonáme nasledujúcim príkazom:

```
zabovsky@frios:~/pokus$ ls -la
total 12
drwxr-xr-x 3 zabovsky zabovsky 4096 Sep  2 20:11 .
drwxr-x--- 6 zabovsky zabovsky 4096 Sep  2 16:15 ..
drwxr-xr-x 2 zabovsky zabovsky 4096 Sep  2 17:15 adresar
-rw-r--r-- 1 zabovsky zabovsky    0 Sep  2 20:11 novy
-rw-r--r-- 1 zabovsky zabovsky    0 Sep  2 20:11 stary
zabovsky@frios:~/pokus$ tar cvf archiv.tar *
adresar/
novy
stary
zabovsky@frios:~/pokus$ ls -la
total 24
drwxr-xr-x 3 zabovsky zabovsky  4096 Sep  2 20:33 .
drwxr-x--- 6 zabovsky zabovsky  4096 Sep  2 16:15 ..
drwxr-xr-x 2 zabovsky zabovsky  4096 Sep  2 17:15 adresar
-rw-r--r-- 1 zabovsky zabovsky 10240 Sep  2 20:33 archiv.tar
-rw-r--r-- 1 zabovsky zabovsky     0 Sep  2 20:11 novy
-rw-r--r-- 1 zabovsky zabovsky     0 Sep  2 20:11 stary
```

Vidíme, že máme vytvorený archív s názvom `archiv.tar`. Ak chceme zistiť, čo je uložené v archíve, použijeme príkaz:

```
zabovsky@frios:~/pokus$ tar tvf archiv.tar
drwxr-xr-x zabovsky/zabovsky 0 2007-09-02 17:15 adresar/
-rw-r--r-- zabovsky/zabovsky 0 2007-09-02 20:11 novy
-rw-r--r-- zabovsky/zabovsky 0 2007-09-02 20:11 stary
```

Pre rozbalenie archívu do adresára `novy_adresar` použijeme nasledujúci postup:

```
zabovsky@frios:~/pokus$ mkdir novy_adresar
zabovsky@frios:~/pokus$ mv archiv.tar novy_adresar/
zabovsky@frios:~/pokus$ cd novy_adresar/
zabovsky@frios:~/pokus/novy_adresar$ ls -la
total 20
drwxr-xr-x 2 zabovsky zabovsky  4096 Sep  2 20:42 .
drwxr-xr-x 4 zabovsky zabovsky  4096 Sep  2 20:42 ..
-rw-r--r-- 1 zabovsky zabovsky 10240 Sep  2 20:33 archiv.tar
zabovsky@frios:~/pokus/novy_adresar$ tar xvf archiv.tar
adresar/
novy
stary
zabovsky@frios:~/pokus/novy_adresar$ ls -la
total 24
drwxr-xr-x 3 zabovsky zabovsky  4096 Sep  2 20:42 .
drwxr-xr-x 4 zabovsky zabovsky  4096 Sep  2 20:42 ..
drwxr-xr-x 2 zabovsky zabovsky  4096 Sep  2 17:15 adresar
-rw-r--r-- 1 zabovsky zabovsky 10240 Sep  2 20:33 archiv.tar
-rw-r--r-- 1 zabovsky zabovsky     0 Sep  2 20:11 novy
-rw-r--r-- 1 zabovsky zabovsky     0 Sep  2 20:11 stary
```

Ak chceme z archívu rozbaliť iba súbor `novy` a `stary` použijeme:

```
zabovsky@frios:~/pokus/novy_adresar$ rm novy stary
zabovsky@frios:~/pokus/novy_adresar$ rmdir adresar
zabovsky@frios:~/pokus/novy_adresar$ ls -la
total 20
drwxr-xr-x 2 zabovsky zabovsky  4096 Sep  2 20:44 .
drwxr-xr-x 4 zabovsky zabovsky  4096 Sep  2 20:42 ..
-rw-r--r-- 1 zabovsky zabovsky 10240 Sep  2 20:33 archiv.tar
zabovsky@frios:~/pokus/novy_adresar$ tar xvf archiv.tar novy
stary
novy
stary
zabovsky@frios:~/pokus/novy_adresar$ ls -la
total 20
drwxr-xr-x 2 zabovsky zabovsky  4096 Sep  2 20:44 .
drwxr-xr-x 4 zabovsky zabovsky  4096 Sep  2 20:42 ..
-rw-r--r-- 1 zabovsky zabovsky 10240 Sep  2 20:33 archiv.tar
-rw-r--r-- 1 zabovsky zabovsky     0 Sep  2 20:11 novy
-rw-r--r-- 1 zabovsky zabovsky     0 Sep  2 20:11 stary
```

Sada príkazov `compress`, `uncompress` a `zcat` slúži na kompresiu, dekompresiu a výpis komprimovaných súborov. Vyššiu účinnosť pri kompresii dosahuje príkaz `gzip`, príkaz `compress` však na rozdiel od príkazu `gzip` nájdeme vo všetkých implementáciách systému Unix. Súbory s príponou `.Z` sú výsledkom kompresie pomocou príkazu `compress`.

Podobne, ako pri príkaze `compress` slúžia príkazy `gzip`, `gunzip` a `zcat` na kompresiu, dekompresiu a výpis komprimovaných súborov. Príkaz `gzip` pridáva

komprimovaným súborom implicitne príponu `.gz`. Keďže príkaz `gzip` je preferovaný v systémoch Linux, ukážeme aspoň základný príklad použitia tohto príkazu:

```
zabovsky@frios:~/pokus/novy_adresar$ rm novy stary
zabovsky@frios:~/pokus/novy_adresar$ gzip archiv.tar
zabovsky@frios:~/pokus/novy_adresar$ ls -la
total 12
drwxr-xr-x 2 zabovsky zabovsky 4096 Sep  2 20:45 .
drwxr-xr-x 4 zabovsky zabovsky 4096 Sep  2 20:42 ..
-rw-r--r-- 1 zabovsky zabovsky  173 Sep  2 20:33 archiv.tar.gz
```

2.6 Spracovanie textu

Nástroje pre spracovanie textu tvoria veľmi silnú skupinu nástrojov často favorizujúcou operačné systémy Linux pre operačnými systémami Windows. Príkazy pre spracovanie textu opäť uvádzame v jednoduchom prehľade.

- Nástroje pre manipuláciu s textom:
  - **grep** - príkaz hľadá v zadanom súbore (alebo štandardnom vstupe) riadky, ktoré vyhovujú zadanému vzoru. Tieto riadky potom implicitne vypisuje na štandardný výstup. Modifikáciou príkazu sú príkazy **egrep** a **fgrep**,
  - **awk** - programovací jazyk pre textové manipulácie. Program awk číta riadky buď zo zadaných súborov alebo zo štandardného vstupu,
  - **more** - ide o filter pre prezeranie textových súborov na obrazovke. Príkaz sa najčastejšie používa na prezeranie tých textových súborov, o ktorých predpokladáme, že budú väčšie ako jedna obrazovka,
  - **less** - program, ktorý je podobný príkazu more, rozširuje však jeho vlastnosti.
- Výstup kompletných súborov:
  - **cat** - spojenie viacero súborov a zápis na štandardný výstup,
  - **nl** - očíslovanie riadkov a zápis na štandardný výstup,
  - **od** - zápis súboru v osmičkovom alebo inom textovom formáte.
- Formátovanie obsahu súboru:
  - **fmt** - formátovanie odseku textu,
  - **pr** - formátovanie do stránok a stĺpcov,
  - **fold** - rozdelenie dlhých riadkov na viac kratších.
- Výstup časti súborov:

- **head** - výpis začiatku súboru,
- **tail** - výpis konca súboru,
- **split** - rozdelenie súboru do rovnako veľkých častí,
- **csplit** - rozdelenie súboru podľa kontextu.

- Sumarizácia obsahu súboru:
  - **wc** - vráti počet bytov, slov, riadkov súboru,
  - **sum** - vráti kontrolný súčet obsahu súboru,
  - **cksum** - vráti CRC súčet obsahu súboru.
- Triedenie obsahu súboru:
  - **sort** - zoradí obsah súboru,
  - **uniq** - vráti unikátne riadky obsahu súboru,
  - **comm** - porovná dva utriedené súbory.
- Spracovanie polí na riadku:
  - **cut** - na výstup sa odovzdajú iba vybrané časti riadku,
  - **paste** - zlúčia sa riadky z viacerých súborov,
  - **join** - zlúčia sa riadky na základe spoločných polí.
- Spracovanie jednotlivých znakov:
  - **tr** - transformuje, maže a ruší opakujúce sa znaky,
  - **expand** - transformuje tabulátory na medzery,
  - **unexpand** - transformuje medzery na tabulátory.

2.7 Záver

V tejto kapitole sme ukázali základné príkazy a definovali základné pojmy pre prácu s operačným systémom Linux. Keďže systémy Unix a Linux sú veľmi podobné, môže poslúžiť táto kapitola aj pre prácu s operačným systémom Unix. Napriek množstvu príkazov, ktoré sme v tejto kapitole popísali, nie je výpočet úplny a určite sa stretnete s inými príkazmi. Pre pokročilého používateľa by však nemal byť problém použiť dokumentáciu v manuálových stránkach.

2.8 Literatúra

[1] Brandejs M., UNIX - LINUX, GRADA Publishing, 1996
[2] Skočovský L., Principy a problémy operačního systému UNIX, Science, 1993

Kapitola 1 1
1.1 Úvod do operačného systému Linux 1
1.1.1 Používateľské účty 2
1.1.2 Administrátor 3
1.1.3 Prihlásenie do systému, odhlásenie zo systému 3
1.1.4 Dokumentácia 3
1.1.5 Vstup a výstup príkazu 4
1.2 Procesy 7
1.3 Systém súborov 8
1.3.1 Súbory a adresáre 8
1.3.2 Odkazy 9
1.3.3 Špeciálne súbory 10
1.3.4 Štruktúra adresárového stromu 11
1.3.5 Prístupové práva 11
1.3.6 Prehľad príkazov pre prácu so súbormi 13
1.4 Ďalšie príkazy shelu 15
1.5 Nástroje pre archiváciu a kompresiu dát 16
1.6 Spracovanie textu 18
1.7 Záver 19
1.8 Literatúra 19