



Projekt v rámci SIPVZ:

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

LINUX

Lekce 12

Základní orientace v systému - 1

Obsah lekce:

Cíle	1
Základní navigace v systému	1
Příkaz cd a ls	
Příkazy pro kontrolu procesů	3
Zobrazení připojených uživatelů	5
Příklad použití who	
Příklad použití w	
Použití w a who pro naleznutí informací o uživateli	
Otázky k opakování	7
Lab.	8

Cíle

Po skončení této lekce studenti budou schopni:

- Definovat a vykonat základní příkazy systému.
- Vyjmenovat dva příkazy používané pro navigaci a manipulaci se souborovým systémem.
- Umět zobrazit připojené uživatele

Základní navigace v systému

- Příkaz cd - používán pro navigaci souborovým systémem.
- Příkaz ls - používán pro zobrazení souborů v adresáři a vlastností adresáře
- Rozložení výstupu ls

Produktivita Linuxu, stejně jako ostatního softwaru, je limitována rozsahem znalostí a schopností uživatel daný program ovládat a využít jeho funkcí. Cílem této sekce je obeznámit uživatele se základními příkazy, utilitami a procedurami nezbytnými pro navigaci v systému a jeho správu. S těmito základními znalostmi uživatel bude moci dále pokračovat v objevování schopností OS Linux.

Příkazy cd a ls

Dvě nejvíce důležité funkce v každém operačním systému jsou přesun mezi adresáři a zobrazení seznamu souborů. V systému Linux slouží příkaz cd pro změnu adresáře a příkaz ls pro zobrazení výpisu souborů. Například příkazová řádka je nastavena tak, aby zobrazovala jméno aktivního uživatele, název pracovní stanice a název současného adresáře:

```
[name@holt home]$
```

Kde name je jméno uživatel, host je název počítače či domény, do které je počítač připojen a home je současný adresář. Dále předpokládejme, že máme adresář mail v adresáři home. To znamená, že je vlastně jeho podadresář a home jeho je jeho rodičovský adresář. Zadáním příkazu cd mail se přepneme z aktuálního adresáře do adresáře mail, takže se nám zobrazí toto:

```
[name@host home]$ cd mail  
[name@host mail]$
```

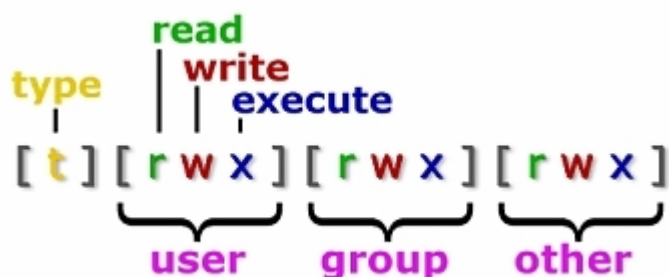
Příkaz ls zadáný bez argumentů zobrazí soubory včetně adresářů, které se nacházejí v aktuálním adresáři. S použitím argumentů může ls zobrazit rozličné typy souborů, informace o nich, přístupová práva a podobně. Například příkaz ls -l zobrazí dlouhou formu výpisu pro všechny soubory:

```
[name@host mail]$ ls -l *
-rw----- 1 jordan users 49 Jul 17 1988 clog
drwxr-x-- 2 jordan users 4096 Sep 27 1998 doclog
-rw-rw-r-- 1 jordan users 46 Jul 29 1991 edclog
-rwxr-xr-x 1 jordan users 19 Sep 29 1991 edolog
```

Z tohoto výstupu můžete poznat, že všechny soubory vlastní uživatel jménem jordan a ten patří do skupiny users. Jordan nemusí být členem této skupiny, ale soubory mohou této skupině náležet. Číslo vlevo od data vytvoření souboru zobrazuje jeho velikost v bytech.

Rozložení výstupu příkazu ls

Všimněte si prvního sloupce hned vlevo. Reprezentuje přístupová práva k danému souboru. Vlastník, skupina a ostatní, mají oddělená práva číst (r), zapisovat (w) a spouštět (x) pro každý soubor v adresáři stejně tak pro samotný adresář. Je zde deset znaků pro tyto nastavení. První indikuje typ, tj. zda se jedná o soubor či adresář. Jestliže je se jedná o soubor je první znak prázdný, pokud je t adresář je zde symbol d. Další tři znaky označují práva vlastníka. Další tři práva pro skupiny. A poslední tři přístupová práva pro všechny ostatní. tudíž jak bylo v našem příkladu, soubor clog má vlastní právo pro čtení a zápis, nikdo jiný nemůže se souborem nakládat. Soubor edclog může číst a zapisovat do něj vlastník a skupina a pro ostatní je k dispozici pouze pro čtení. soubor ekolog je přístupný pro čtení a spuštění všem, ale pouze vlastník do něj může zapisovat a tím měnit jeho obsah. Obrázek 8-1 zobrazuje tyto práva více rozkreslena.



Obrázek 12-1 – Schéma přístupových práv v systému Linux

Právo spuštění (x) je interpretováno jako právo vyhledat pokud se jedná o adresář. Například zadání příkazu `ls -ld` zobrazí práva pro příslušný adresář:

```
[name@host mail]$ ls -ld
drwxr-xr-- 3 ptm html 8192 Jun 25 18:09
```

Pamatujte si, že znak nejvíce vlevo (d – directory) označuje adresář. Jako adresář, příznak x je změněn což značí, že vlastník a skupina mají právo hledat, ale ostatní ne. Jakože není přítomno x pro ostatní, příkaz `ls` nezobrazí soubory v adresáři pro ostatní než ve skupině html. Ovšem soubory mohou být stále přístupny přes jejich relativní cesty. Jinak řečeno, uživatelé mimo skupinu html nemohou vidět soubory clog, edclog a edolog, ale stále k nim mohou přistupovat, pokud znají jejich umístění a přesný název.

Příkazy pro kontrolu procesů

- ps – Příkaz stav procesu
 - Zobrazí běžící procesy
 - ps aux nebo ps eax vypíše více
- top – seřadí proces podle využití prostředků
- kill – ukončení procesu
 - Užitečné pro ochromené terminály, programy v X
- renice – mění prioritu procesů
 - Od HIGH až po LOW
 - Superuživatel může měnit prioritu všech procesů.

Příkaz ps

Uživatel přihlášený do systému vytváří procesy, aniž by třeba prováděl nějakou činnost. První proces je vytvořen systémem a může být zobrazen použitím příkazu ps -aux. Výstup hned po zalogování uživatel bude vypadat zhruba takto:

PID	TTY	TIME	CMD
11519	pts/1	00:00:00	bash
11531	pts/1	00:00:00	ps

Na začátku je identifikační číslo procesu (PID), dále vidíme, že uživatel je přihlášen z pseudo terminálu (pts/1) a systémový program je shell bash. Sloupec TTY zobrazuje jak je uživatel přihlášen, zda z konzole, přímo připojeného terminálu, nebo přes síť a podobně. V dalších verzích Linuxu uvidíme toto:

PID	TTY	STAT	TIME	COMMAND
397	3	S	00:00	-bash
11539	p0	S	00:00	-bash
11531	p0	R	00:00	ps

V této verzi je navíc kolonka STAT, která ukazuje stav procesu. Ty mohou být (S) jako odložený, (T) jako zastavený, (R) jako běžící, (Z) jako Zombie (mrtvý proces) a (D) jako nepřerušitelný.

Výstup příkazu ps zobrazuje aktuální běžící úlohy uživatele, ovšem pouze pokud je přihlášen. Další formou příkazu ps je použití přepínačů -eax, která zobrazí všechny procesy a více informací o nich. Pro získání všech přepínačů použitelných v příkazu ps se podívejte do příslušných stránek man. Výstup příkazu ps - eax by mohl vypadat zhruba takto:

PID	TTY	STAT	TIME	COMMAND
1	?	S	00:04	init
2	?	S	00:01	[kflush]
11534	pts/1	T	00:00	bash
11530	pts/1	R	00:00	ps -eax

Příkaz top zobrazí v podstatě ty samé informace, ale po pravidelných intervalech je aktualizuje.

Příkazy kill a renice

Jak již bylo předvedeno příkaz top dovoluje uživateli vidět jeho procesy a jejich systémové nároky. Příkazy kill a renice mu zase dovolují je plně kontrolovat. Příkaz kill odesílá procesu jeden z šestnácti možných signálů a může být použit pro jejich ukončení. Příkaz renice dokáže měnit prioritu procesů a upřednostnit tak ty, které potřebujeme provádět rychleji. V následujícím příkladu je použit příkaz top, pro zobrazení běžících procesů a jejich priority.

```
bash$ top
PID  USER  PRI   NI   SIZE  RSS  SHARE   STAT  LIB  %CPU%MEM  CMD
179  jjbritt  15   0   1296  100   56      R    0   87,9  0.0  wine
874  root    1    0  43328 40M  2716     S    0   4,7  32,0  x
868  sryburn0  0    0   5504  5504  3864     S    0   2,5  4,2  gnome
897  sryburn2  0    0   1016  1106   780     R    0   0,9  0,7  top
19035 sryburn1  0    0   2276  2200  1152     S    0   0,1  1,7  saw
```

Poznámka: Sloupec PRI zobrazuje stupeň priority procesu.

Příkazem kill můžeme ukončit jakýkoliv běžící proces. Například, běželi gnome-terminal, který se zaseknul a chce ho uživatel ukončit, aniž by musel restartovat relaci, tak zadá následující příkaz pro ukončení procesu s PID 19868 a způsobí jeho ukončení.

```
kill – KILL 19868
```

nebo

```
kill –9 19868
```

Pokud není potřeba nebo nechceme ukončit proces, můžeme využít příkaz renice pro změnu priority jednotlivých procesů. Rozsah sahá od -20 do 19, kde -20 je největší priorita a 19 je nejmenší. Implicitní hodnota je 0. Pokud chce prioritu změnit uživatel, může pouze měnit prioritu svých procesů a to od hodnot 0 do 19. Pouze superuživatel může měnit priority všech procesů a také v celé škále úrovní. V následujícím příkladu snížíme prioritu procesu ze standardní úrovně nula na 15.

```
$bash renice 15 19868
```

Poznámka: Příkaz kill -9 0 vypne celý systém pokud byude zadán super uživatelem.

Zobrazení připojených uživatelů

- who – zobrazí aktuálně přihlášené uživatele
 - včetně použitého terminálu
- w – zobrazí mnohem více
 - souhrn stavu systému
 - host, ze kterého je uživatel připojen
 - uživatelův terminál
 - příkaz ,který má uživatel aktuálně spuštěn
- Zkusíte spustit `who | wc -l` pro součet přihlášených uživatelů.

Pro zobrazení přihlášených uživatelů, nebo zobrazení běžících virtuálních terminálů se používají příkazy `who` nebo `w`. Příkaz `who` zobrazí výpis uživatelů, kteří jsou aktuálně přihlášeni do systému a který používají terminál. Příkaz `w` poskytuje mnohem více informací o systému. Zobrazí souhrn zatížení systémových prostředků, na kterém terminálu, odkud jsou přihlášeni uživatelé a který příkaz právě provádějí.

Příklad použití who:

```
hostname:~$ who
bbroen      tty1Apr208:58
bbroen      tty0Apr21:01 (host1.yourcompany.com)
bbroen      tty1Apr307:59 (host1.yourcompany.com)
bbroen      tty2aprd1113:53 (sales.offsite.com)
```

Uživatel `bbroen` je přihlášen celkem čtyřikrát, jednou na virtuální systémové konzoli (`tty1`), dvakrát pře síť z počítače `host1` a jednou z počítače jménem `sales`.

Příklad použití w:

```
hostname:~$ w
11:54am up 2 days, 14:16, 4 users, load average 0.00, 0.00, 0.00
USER      TTY  FROM      LOGIN@    IDLE  JCPU  PCPU  WHAT
bbrown    tty1                Fri 8am   2.26  0.91s 0.91s -bash
bbrown    tty0  host1.com  Fri 11am  1:00s  2.91s 0.30s  w
bbrown    tty1  host2.com  7:59am   1:38m  2.89s 1.81s  bash
bbrown    tty2  sales.off  Thu 1pm   44:21m 0.88s 0.88s -bash
```

Příkaz `w` zobrazuje ty samé informace, ovšem přidá též využití systémových prostředků.

Použití w a who pro zobrazení informací o uživateli

Jako vytížený systémový administrátor chcete pouze vědět počet aktuálně přihlášených uživatelů. Můžete přesměrovat výstup příkazu w do wc utility a ta již sečte jejich celkový počet.

Zobrazení počtu uživatelů v systému

```
hostname:/$ who | wc -l
```

Nebo můžete najít specifického uživatele na systému, který jich má hodně, jednoduchým zadáním w nebo who a znaku | přičemž je odešlete do příkazu grep, který hledá řetězec, jenž zadáte.

Nalezení uživatele na vytíženém stroji

```
hostname:/$ who | grep root
```

Je zde mnoho dalších zajímavých parametrů, zkuste si vyzkoušet a uložit jako oblíbené pro pozdější potřeby

Otázky k opakování

Lab
