

UŽIVATEL, SKUPINA, PROCES

System bez uživatele je jedině Matrix?

Uživatelé se seskupují a řídí práci.



Group = skupina uživatelů

Trocha teorie:

LINUX je systémem **víceuživatelským**. Tzn. Že k dané instanci OS se může v reálném čase připojit současně **více** uživatelů, a to jak **lokálně** na daném PC, tak i **vzdáleně**.

Z toho důvodu musí mít OS informace o jednotlivých uživateli, o jejich přihlašovacích údajích, o jejich právech apod.

LINUX má uživatele sdruženy do **skupin**. Každý uživatel je členem minimálně **jedné** skupiny, může být však členem i **více** skupin.

Group = skupina uživatelů

Skupina je jednoznačně určena svým GID.

GID – Group Identifier – jednoznačné číslo určující skupiny

Informace o skupinách jsou uloženy v souboru `/etc/group`

```
gdm:x:120:  
nopasswdlogin:x:121:  
houda:x:1000:  
sambashare:x:122:houda  
!ost:x:1001:
```

jmeno_skupiny:x:GID:[clen1:clen2: ... clenX]

jmeno_skupiny - název skupiny

x - x – účet je aktivní, ! - skupina je zablokovaná

GID - group ID – jednoznačný číselný identifikátor skupiny

clen1 .. clenN - nepovinný seznam uživatelů, kteří jsou také členy skupiny

User = uživatel

Uživatel je jednoznačně určen svým UID.

UID – User Identifier – jednoznačné číslo určující uživatele

Informace o uživateli jsou uloženy v souborech **/etc/passwd** a **/etc/shadow**.

Soubor **/etc/passwd** v sobě uchovává kompletní seznam uživatelských **úctů**, a to jak reálných, tak systémových (virtuálních).

Soubor **/etc/shadow** obsahuje kryptovaná hesla k jednotlivým účtům.

User = uživatel

Zkrácený výpis /etc/passwd

username:x:UID:GID:GECOS:homedir:shell

```
root:x:0:0:root:/root:/bin/bash
```

```
houda:x:1000:1000:Pavel Houda,,,:/home/houda:/bin/bash  
host:x:1001:1001:./home/host:/bin/sh
```

username - uživatelské jméno

x – účet je aktivní, ! – účet je zablokován

UID - user ID – jednoznačný číselný identifikátor uživatele

GID - group ID – jednoznačný číselný identifikátor primární skupiny

GECOS - popis uživatele (např. jméno a příjmení)

homedir - domovský adresář

shell - shell, který bude spuštěn po přihlášení uživatele. Pro znemožnění přihlášení na konzoli je možno uvést například /bin/false

Vytvoření skupiny

Abychom mohli přidávat uživatele, je dobré mít pro ně skupiny.

Příkaz: **groupadd** – přidání skupiny

Syntaxe: **groupadd** [*prepinace*] *nazev_skupiny*

přepínač **-g** umožňuje založit skupinu s určeným GID

příklady: **groupadd 3V** založí skupinu 3V s automatickým GID

groupadd -g 2000 3V založí skupinu 3V s GID = 2000

Smazání skupiny

Příkaz: **groupdel** – smazání skupiny

Syntaxe: **groupdel** *název_skupiny*

Pozn.: K vytvoření a smazání skupiny musíte mít dostatečná práva, tj. musíte možnost upravit soubor /etc/group, vynucení rootovských oprávnění příkazem sudo nám pro náš výklad postačí.

Modifikace skupiny

Příkaz: **groupmod** – modifikace skupiny

Syntaxe: **groupmod [prepinace] nazev_skupiny**

přepínače: **-g XXX** nastaví skupině GID na XXX
 -n jmeno přejmenuje skupinu na jmeno

příklady: **groupmod -g 2000 3V** změní GID skupiny 3V na 2000
groupmod -n 4V 3V přejmenuje skupinu 3V na 4V

Vytvoření uživatele

Když už máme skupiny, je dobré nemít je prázdné.

Příkaz: **useradd** – přidání uživatele

Syntaxe: **useradd [přepínač] jmeno_uzivatele**

přepínač	popis
-G groups	zařadí uživatele do dalších skupin daných seznamem skupin (oddělujeme je čárkou)
-d homedir	domovský adresář uživatele
-s shell	shell pro uživatele (nebo /bin/false)
-m	domovský adresář bude vytvořen pokud neexistuje
-M	domovský adresář nebude vytvořen
-u UID	založí uživatele s daným UID

příklad: `useradd -g 3V -d /home/pepa -m -s /bin/bash pepa`

Smazání uživatele

Příkaz: ***userdel*** – *smazání skupiny*

Syntaxe: ***userdel jmeno_uzivatele***

přepínač ***-r*** zároveň s účtem smaže i soubory v domovském adresáři a poštu

Modifikace uživatele

Příkaz: ***usermod*** – *modifikace skupiny*

Syntaxe: ***usermod [prepinace] jmeno_uzivatele***

Přepínače ***-g, -G, -s, -d, -u*** mají stejný význam jako v případě příkazu ***useradd***.

přepínače: ***-l new_username*** změní uživatelské jméno uživatele při zachování UID a dalších údajů
 -L zablokuje (LOCK) účet
 -U odblokuje (UNLOCK) účet

Pozn.: K vytvoření a smazání uživatele musíte mít dostatečná práva, tj. musíte možnost upravit soubor ***/etc/passwd***, vynucení rootovských oprávnění příkazem ***sudo*** nám pro náš výklad postačí.

Výpis členů skupiny

Příkaz: **groups** – výpis skupin uživatele

Syntaxe: **groups jméno_uživatele**

Pokud použijeme příkaz *groups* bez jména uživatele, vypíše se skupiny aktuálně přihlášeného uživatele.

Skupiny uživatele

Příkaz: **members** – výpis členů skupiny

Syntaxe: **members jméno_skupiny**

Program *members* není v ubuntu standardně nainstalován a je třeba jej přidat:
sudo apt-get install members

Procesy

- každý program v Linuxu se po spuštění stává procesem
- o správu procesů se stará jádro
- jádro přidělí na základě výše jeho priority část paměťového prostoru a případně další systémové prostředky, je-li to vyžadováno
- jeden program může vytvořit více procesů

ps – příkaz na vypisování běžících procesů

Syntaxe: `ps [parametry]`

`ps` bez parametrů vypíše procesy spuštěné uživatelem

```
houda@houda-Lynx:~$ ps
  PID TTY          TIME CMD
 1237 pts/0        00:00:00 bash
 1253 pts/0        00:00:00 ps
houda@houda-Lynx:~$
```

Sloupce, pid číslo procesu, identifikace terminálu, na kterém běží, využití procesorového času, jméno procesu.

***ps aux ...* výpis všech procesů**

```
houda@houda-Lynx:~$ ps aux
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.3  0.3   2804   1636 ?        Ss   13:54   0:00 /sbin/init
root         2  0.0  0.0     0     0 ?        S    13:54   0:00 [kthreadd]

houda    1229  0.0  1.4  18968   7228 ?        S    13:54   0:00 /usr/lib/gnome-
houda    1232  0.7  2.5  44972  12748 ?        Sl   13:54   0:00 gnome-terminal
houda    1233  0.0  0.1   1988    708 ?        S    13:54   0:00 gnome-pty-helpe
houda    1234  0.1  0.6   5840   3204 pts/0    Ss   13:54   0:00 bash
houda    1251  0.1  2.9  31544  15240 ?        S    13:55   0:00 python /usr/sha
houda    1252  0.1  2.1  34664  11036 ?        Sl   13:55   0:00 /usr/lib/evolut
houda    1260  0.2  2.2  35264  11492 ?        S    13:55   0:00 update-notifier
root     1264  0.1  1.5  13688   8036 ?        S    13:55   0:00 /usr/bin/python
root     1266  0.2  2.1  16420  11100 ?        S    13:55   0:00 /usr/bin/python
houda    1275  0.0  0.2   2716   1056 pts/0    R+   13:57   0:00 ps aux
```

Pozn. výpis byl zkrácen

Sloupec STAT identifikuje stav procesu – pro podrobnější pochopení navštivte *man ps*.

Příkaz *ps* poskytuje pouze statický výpis procesů v paměti.

Příkaz top – dynamický výpis

Syntaxe: `top -p-parametry`

```
top - 22:56:13 up 44 min, 2 users, load average: 0.21, 0.28, 0.26
Tasks: 128 total, 1 running, 127 sleeping, 0 stopped, 0 zombie
Cpu(s): 0.7%us, 2.3%sy, 0.0%ni, 97.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Mem: 509264k total, 485632k used, 23632k free, 56464k buffers
Swap: 407544k total, 0k used, 407544k free, 254656k cached
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
916	root	20	0	40792	18m	8044	S	1.0	3.7	0:19.60	Xorg
1187	houda	20	0	38660	12m	10m	S	1.0	2.6	0:09.35	wnck-applet
1549	houda	20	0	2548	1200	920	R	0.7	0.2	0:00.21	top
1033	root	20	0	3616	1320	1144	S	0.3	0.3	0:05.34	hald-addon-stor
1235	houda	20	0	44332	12m	9.9m	S	0.3	2.6	0:10.21	gnome-terminal
1	root	20	0	2804	1648	1196	S	0.0	0.3	0:00.66	init
2	root	20	0	0	0	0	S	0.0	0.0	0:00.01	kthreadd
3	root	RT	0	0	0	0	S	0.0	0.0	0:00.00	migration/0
4	root	20	0	0	0	0	S	0.0	0.0	0:00.05	ksoftirqd/0
5	root	RT	0	0	0	0	S	0.0	0.0	0:00.00	watchdog/0
6	root	20	0	0	0	0	S	0.0	0.0	0:00.50	events/0
7	root	20	0	0	0	0	S	0.0	0.0	0:00.00	cpuset
8	root	20	0	0	0	0	S	0.0	0.0	0:00.00	khelper
9	root	20	0	0	0	0	S	0.0	0.0	0:00.00	async/mgr
10	root	20	0	0	0	0	S	0.0	0.0	0:00.00	pm
11	root	20	0	0	0	0	S	0.0	0.0	0:00.01	sync_supers
12	root	20	0	0	0	0	S	0.0	0.0	0:00.01	bdi-default

Základní parametry příkazu top:

- **d ...** prodleva v s
- **n ...** počet načtení
- **p ...** sledování procesu s daným pid
- **u...** sledování procesů daného uživatele

[Odkaz na české manuálové stránky](#)

Ukončení procesu

Procesy lze ukončit pomocí tzv. signálů. Podrobný rozbor signálů je v tuto chvíli zbytečný. Uvedeme si pár jednoduchých příkladů.

Proces běžící v popředí lze ukončit tím, že mu zašleme signál SIGINT, to provedeme stisknutím kláves CTRL+C.

Proces běžící na pozadí ukončujeme příkazem kill.

Kill – ukončení procesu

Syntaxe: `kill -jméno_signálu pid`

Pokud jméno signálu vynecháme, příkaz kill pošle procesu signál TERM, což je v podstatě žádost o „legální“ ukončení procesu. Proces jej může z různých důvodů blokovat.

„Kousnutý“ proces lze ukončit „hrubou silou“ nebo-li „vraždou“:
`kill -kill pid` – tento signál nelze blokovat

[Odkaz na české manuálové stránky](#)

Problematika příkazu kill

Příkaz kill toho umí mnohem víc, než jen ukončovat procesu. Obecně může příkaz kill zaslat libovolný signál procesu, například signál, který znovu spustí pozastavený proces.

více zjistíte: `man kill`