



Přednáška 5

Identita uživatelů, procesů a souborů.

Přístupová práva a jejich nastavení.





Uživatel

- **Při přihlášení do systému musí uživatel:**
 - **identifikovat systém**, na který se chce přihlásit
 - fyzické umístění (lokální přihlášení)
 - jméno systému/ IP adresa (vzdálené přihlášení)
 - zadat **uživatelské jméno**
 - prokázat se odpovídajícím **heslem**

- Pro úspěšné přihlášení musí být na daném systému vytvořen příslušný **uživatelský účet**.



Uživatelský účet I

- Pro každý uživatelský účet musí být definováno:
- **uživatelské jméno (jméno)**
 - přidělováno administrátorem, max. 8 znaků
 - nesmí být tvořeno pouze velkými písmeny
 - různé účty by měly mít různé jména
- **identifikační číslo uživatele (UID)**
 - přidělováno administrátorem
 - celé číslo (dříve max. 65535, nyní i více ale nedoporučuje se)
 - UID=0 definuje tzv. privilegovaný účet (obvykle se jménem root)
 - pod tímto UID bude po přihlášení uživatele spuštěn jeho shell
 - různé účty by měly mít různá čísla
- **identifikační číslo primární skupiny (GID)**
 - přidělováno administrátorem
 - celé číslo (dříve max. 65535, nyní i více ale nedoporučuje se)
 - pod tímto GID bude po přihlášení uživatele spuštěn jeho shell

Uživatelský účet II

- **úplné jméno uživatele**
 - podrobnější popis uživatele
 - je posíláno jako úplné jméno v mailu
- **přihlašovací interpret (shell)**
 - absolutní cesta k příkazu, který je spuštěn po přihlášení uživatele z řádkového terminálu
 - obvykle shell, ale může být i jiný program
- **domovský adresář (adresář)**
 - adresář, ze kterého je spuštěn přihlašovací interpret
 - je na něj nastavena proměnná **HOME**
 - zašifrovaná podoba hesla
- **seznam sekundárních skupin**
 - uživatel může pomocí příkazu **newgrp** nastavit sekundární skupinu jako svou primární skupinu (tzn. bude mít právo skupinového přístupu k souborům této skupiny)

Uživatelský účet III

- **zašifrované heslo (heslo)**

- mění se příkazem **passwd**
- uživatel musí znát původní heslo, root ne
- uživatel musí dodržovat pravidla (délky, znaky) root ne

sůl
(2 znaky)

vstupní heslo
(max. 8 znaků)

crypt(3)

/etc/shadow

Jméno Zašifrované heslo
(2+11 znaků)

Jméno	Zašifrované heslo (2+11 znaků)	
novak	kgTd6j\$2f%Dxo	...



Databáze uživatelských účtů

- **/etc/nsswitch.conf**
 - definuje odkud se budou příslušné informace číst
- **Lokální databáze**
 - informace jsou uloženy v lokálních souborech
 - /etc/passwd
 - /etc/shadow
 - /etc/group
- **Centrální databáze (jmenné služby)**
 - umožňuje centrální správu informací (např. uživatelských účtů, ...)
 - NIS (příkazy ypcat, ypmatch, ...)
 - NIS+ (příkazy niscat, nisgrep, ...)
 - LDAP (příkazy ldaplist, ...)



- **soubor /etc/passwd**

- slouží k překladu UID na jméno a naopak a k uložení informací nutných pro přihlášení uživatele
- pro každý uživatelský účet je jedna řádka rozdělená znaky : na sedm položek

`jméno:x:UID:GID:úplné jméno uživatele:adresář:shell`

- **soubor /etc/shadow**

- obsahuje zašifrované heslo a parametry nastavení hesla
- pro každý uživatelský účet jedna řádka rozdělená znaky : na devět položek

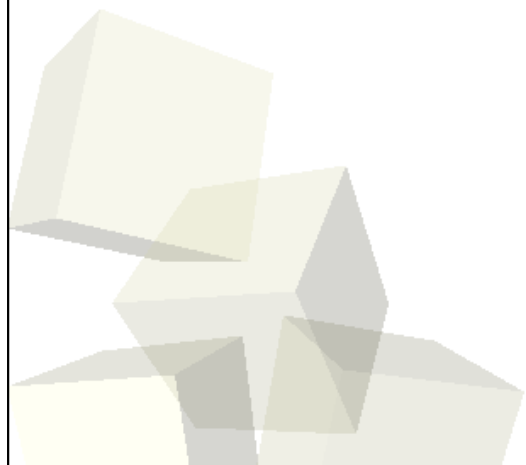
`jméno:heslo:lastchg:min:max:warn:inactive:expire:flag`



- **soubor /etc/group**

- slouží k překladu GID na jméno skupiny a naopak a k definici tzv. sekundárních skupin
- pro každou skupinu jedna řádka rozdělena znaky : na čtyři pole

`jméno skupiny:x:GID:seznam uživatelů`





Přihlášení do systému

- **system vypíše** na příslušném zařízení **prompt**
- **uživatel vloží** uživatelské **jméno** a odpovídající **heslo**
- **system ověří vložené informace** proti databázi uživatelských účtů
- **system spustí přihlašovací shell** a nastaví pro tento proces:
 - **pracovní adresář** na domový adresář daného účtu
 - **reálné číslo uživatele RUID** = UID
 - **efektivní číslo uživatele EUID** = UID
 - **reálné číslo primární skupiny RGID** = GID
 - **efektivní číslo primární skupiny EGID** = GID

Identita procesu

- **Reálná identita procesu (RUID, RGID)**

- systém si pamatuje pod jakým uživatelským účtem jsme se původně přihlásili (např. lokálně nebo vzdáleně pomocí ssh,...)
- lze ji zobrazit např. pomocí následujících příkazů

```
who am i
```

```
ps -o ruid,rgid,comm
```

- **Efektivní (aktuální) identita procesu (EUID, EGID)**

- slouží k autorizaci procesu uvnitř systému (např. při vyhodnocování přístupu k souborům, ...)
- při přihlášení je reálná a efektivní identita totožná
- efektivní identitu lze změnit např. pomocí příkazu **su** nebo speciálních práv binárních souborů
- lze ji zobrazit např. pomocí následujících příkazů

```
id
```

```
ps -o uid,gid,comm
```

```
pcrd číslo-procesu
```



Změna identity procesu

- **Identitu procesu nastavuje kernel při startu** procesu nebo ji mění na žádost procesu.
- Obvykle jsou RUID a EUID resp. RGID a EGID stejná a **dědí se od rodičovského procesu.**
- **Ve zvláštních případech se nedědí**, ale nastavují se všechna nebo jen některá:
 - při přihlášení (pomocí procesů **login/dtlogin**)
 - pomocí příkazu **su**
 - u binárních programů s nastaveným **suid** bitem se mění **EUID**
 - u binárních programů s nastaveným **sgid** bitem se mění **EGID**



SU

`su [-] [uživatelské jméno]`

- Startuje nový shell pod novou identitou.
- Původní shell nekončí, po odhlášení z `su` se v něm pokračuje.
- Je-li `su` volán uživatelem, vyžaduje heslo, od roota ne.
- Je-li uveden přepínač `-`, provede přihlašovací skripty (nastaví prostředí).
- Je-li vynecháno přihlašovací jméno, doplní se jméno `root`.

`newgrp sekundární_skupina`

- Startuje nový shell s novou skupinovou identitou.

Příklad

```
$ id
```

```
uid=0(root) gid=1(other)
```

```
$ su - trdlicka
```

```
Sun Microsystems Inc. SunOS 5.10 Generic January 2005
```

```
You have new mail.
```

```
$ id -a
```

```
uid=4365(trdlicka) gid=1002(k336) groups=1002(k336),2003(y36uos)
```

```
$ newgrp y36uos
```

```
$ id
```

```
uid=4365(trdlicka) gid=2003(y36uos)
```

```
$ newgrp k336
```

```
$ id
```

```
uid=4365(trdlicka) gid=1002(k336)
```

Přístupová práva I

- **Každý soubor/adresář má v i-uzlu:**
 - vlastníka souboru (UID)
 - vlastnickou skupinu (GID)
 - přístupová práva čtení (**r**ead), zápis (**w**rite) a spuštění (**e**xecute) pro vlastníka (**u**ser), skupinu (**g**roup) a ostatní (**o**ther).
- Tyto informace můžeme vypsat např. pomocí příkazu **ls -l**:

přístupová práva pro
vlastníka, skupinu, ostatní

`-r-xr-xr-x 1 root bin 10260 Jan 23 2005 /usr/bin/cat`

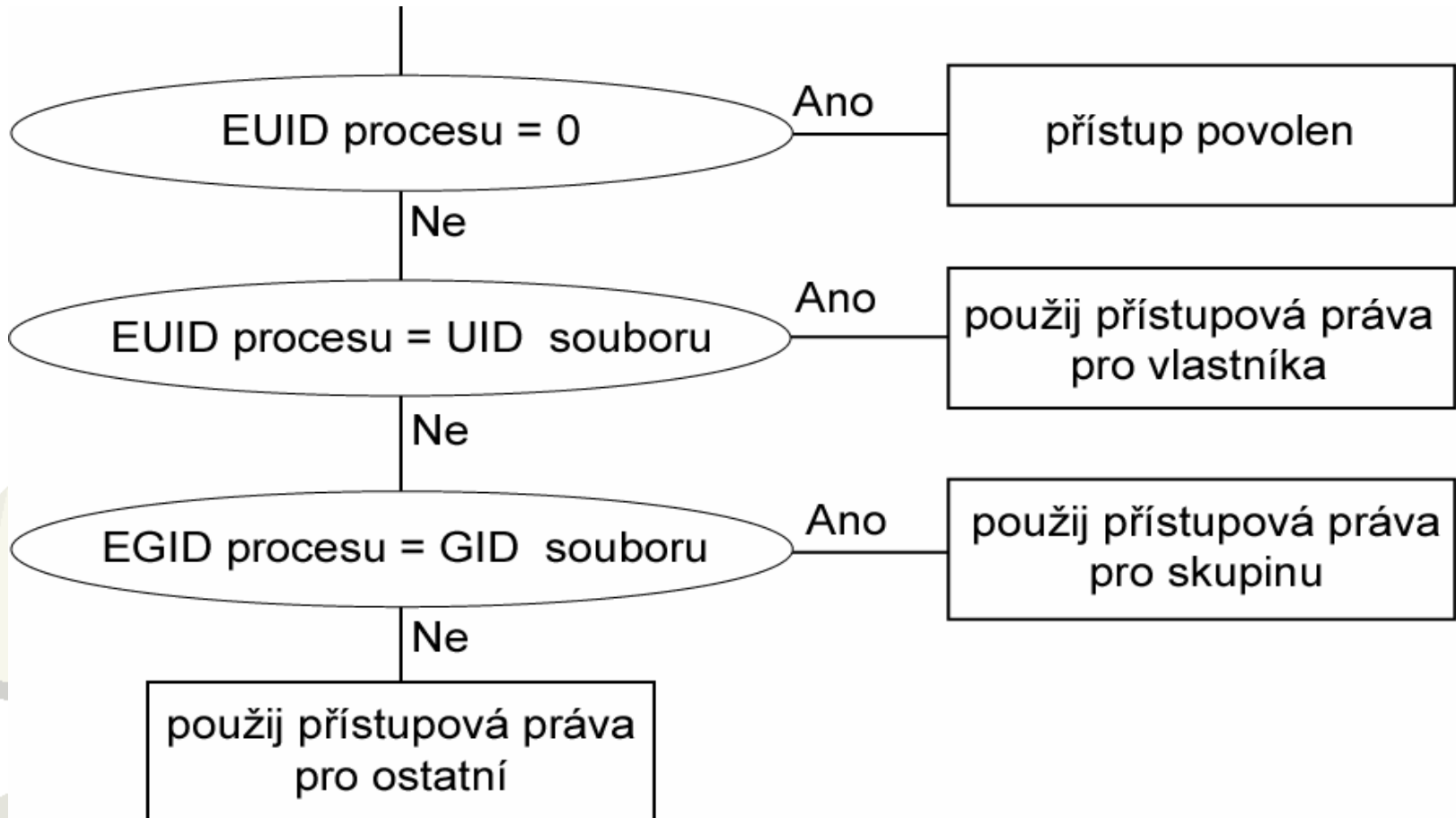
typ souboru

vlastník (UID) skupina (GID)

Přístupová práva II

Právo	Význam u souborů	Význam u adresářů
r	číst obsah souboru (cat)	vypisovat obsah adresáře (ls)
w	měnit obsah souborů (vi)	vytvářet a rušit soubory v adresáři (rm)
x	spouštět soubor jako program	nastavovat a procházet adresář (cd)

Vyhodnocování práv



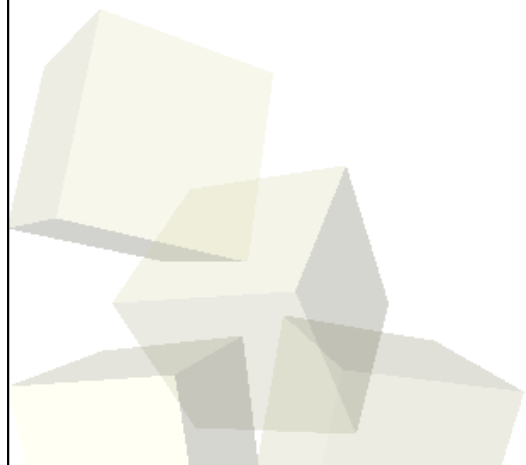


Změna přístupových práv

`chmod [-R] práva seznam_souborů`

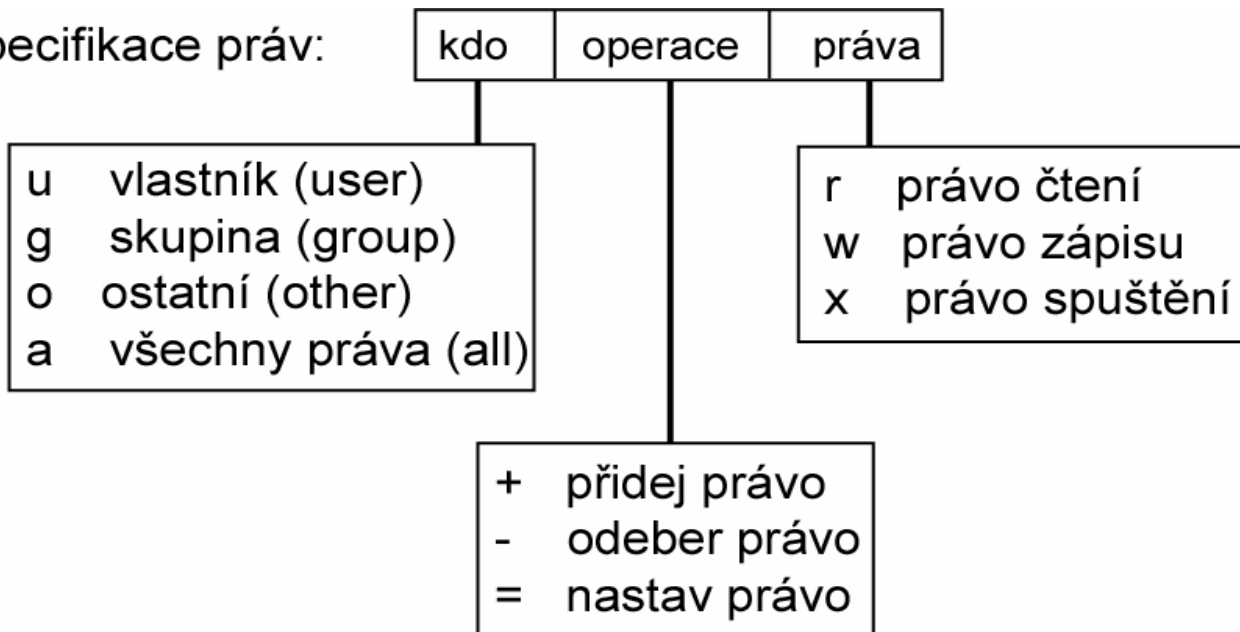
-R změnu práv se aplikuje na všechny soubory
a podadresáře v daném adresáři

práva můžeme zadat symbolicky nebo absolutně (oktalově)



Symbolický mód

Specifikace práv:



Příklad.

```
$ ls -l a.txt
```

```
-rw-r--r-- 1 trdlicka k336      1105 Oct 23 20:52 a.txt
```

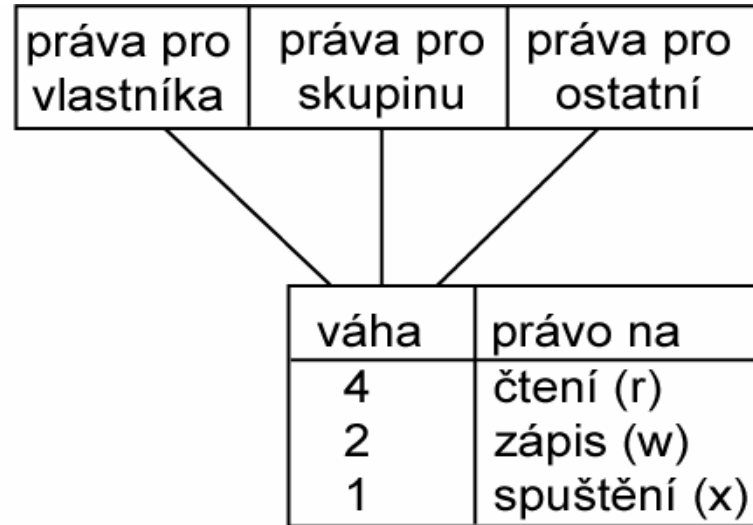
```
$ chmod u+x,g-r,o+w a.txt
```

```
$ ls -l a.txt
```

```
-rwx---rw- 1 trdlicka k336      1105 Oct 23 20:52 a.txt
```

Absolutní (oktalový) mód

Specifikace práv:



Příklad:

```
$ ls -l a.txt
```

```
-rw-r--r-- 1 trdlicka k336      1105 Oct 23 20:52 a.txt
```

```
$ chmod 706 a.txt
```

```
$ ls -l a.txt
```

```
-rwx---rw- 1 trdlicka k336      1105 Oct 23 20:52 a.txt
```

Maska přístupových práv

- **Definuje přístupová práva nově zakládaných souborů/adresářů.**
- Hodnota masky je součástí procesu (podobně jako EUID,EGID,...) a je dědičná.
- Lze ji vypsat a měnit příkazem **umask**.
- Přístupová práva vzniknou množinovým rozdílem výchozí hodnoty a masky.
- **Výchozí hodnota je 666 pro soubory a 777 pro adresáře.**

maska	soubor	adresář	poznámka
000	666	777	odpovídá výchozí hodnotě, Nebezpečné
022	644	755	obvyklé nastavení
027	640	750	vyšší bezpečnost
077	600	700	největší restrikce
066	600	711	kompromisní řešení

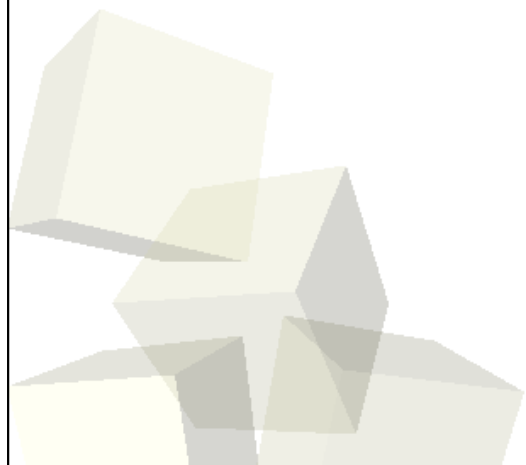


Změna vlastnictví souboru

- **Může měnit pouze root** (dříve i vlastník, to je ale bezpečnostní problém).
- Vlastnictví (i skupinové) lze měnit příkazem **chown**, skupinové příkazem **chgrp**.

```
chown [-R] vlastník [:skupina] seznam_souborů
```

```
chgrp [-R] skupina seznam_souborů
```



Speciální přístupová práva

Právo	Nastavení	Význam u souboru	Význam u adresáře
s(uid)	4000 u+s	binární program má po spuštění EUID vlastníka souboru	žádný
s(gid)	2000 g+s	binární program má po spuštění EGID vlastnické skupiny (je-li x pro skupinu) soubor se povinně zamyká (není-li x pro skupinu)	nové soubory v adresáři dědí GID z adresáře nikoliv z procesu toto právo nelze nastavit absolutním způsobem
l(ock)			
s(t)icky s(T)icky	1000 o+t	žádný není-li nastaveno x pro ostatní, souboru není měněn čas přístupu (swap)	soubory v adresáři s právy rwxrwxrwt smí zakládat každý, ale rušit smí pouze vlastník, root a uživatel s právem zápisu do souboru