

Unixové nástroje (oprávnění: chmod, chown, chgrp)

Jelikož jsou unixové systémy víceuživatelské, je zde nutnost nějakého systému přístupových práv, aby uživatelé nemohli přistupovat k citlivým datům jiných uživatelů, páchat škodu na systému a tak podobně. Standard POSIX definuje systém oprávnění, který používají všechny unixové systémy.

Základní unixová přístupová práva

[link](#)

Každý soubor či adresář má svá základní oprávnění. To zahrnuje vlastníka (UID) a skupinu (GID), a dále oprávnění ke čtení (**r**ead), zápisu (**w**rite) a spuštění (**e**xecute) pro vlastníka (**u**ser), skupinu (**g**roup) a ostatní uživatele (**o**thers). Význam tučných písmen je dobré si zapamatovat, nejen proto, že je budeme používat dále v článku.

Právo	Význam u souboru	Význam u adresáře
r	Čtení souboru.	Čtení názvů obsažených položek.
w	Zápis do souboru.	Vytváření souborů a adresářů.
x	Spuštění souboru.	Vstup do adresáře.

Samotný popis je pro nezasvěceného patrně složité pochopit, a proto to předvedu na ukázce. Již ve [druhém díle](#) jsme si představili [příkaz ls](#), který nám teď pomůže s vypsáním a pochopením práv. Vypíšeme si adresář s různými soubory, které mají různá práva.

```
$ ls -lF --color=auto
brw-rw---- 1 root root  7, 0 2009-11-08 14:41 blokov_e_zarizeni
-rw----- 1 dave home 16 2009-11-08 14:34 muj_tajny_soubor.txt
prw-r--r-- 1 dave home  0 2009-11-08 14:43 pojmenovana_roura|
drwxr-xr-x 1 dave home  0 2009-11-08 14:37 prazdny_adresar/
-rwxr-xr-x 1 dave home 5558272 2009-11-08 14:37 spustitelny_soubor*
lrwxrwxrwx 1 dave home  20 2009-11-08 14:35 symlink_na_muj_tajny_soubor -> muj_tajny_soubor.txt
srwxr-xr-x 1 dave home  0 2009-11-08 17:34 unixovy_socket=
crw-rw-r-- 1 root root  5, 2 2009-11-08 14:41 znakov_e_zarizeni
```

Zajímají nás teď především první tři sloupce. V prvním sloupci jsou vypsána oprávnění. První znak udává typ souboru (***** to si vysvětlíme později, s právy to nesouvisí). Další tři znaky znázorňují oprávnění pro vlastníka, následující tři pro skupinu a poslední tři pro ostatní uživatele. Ve druhém sloupci je uvedeno jméno vlastníka a ve třetím potom název skupiny.

Co vlastně jednotlivé znaky u oprávnění znamenají, vysvětlím třeba na adresáři nazvaném *prazdny_adresar*. Když ignorujeme první znak, máme **rw-r-xr-x**. To si můžeme představit rozdělené po třech znacích jako **rw**, **r-x**, **r-x**, tedy **vlastník, skupina, ostatní**. Znaky r, w, x mají již popsany význam a spojovník (-) znamená, že dané právo adresář nemá. Vlastník může s adresářem manipulovat, jak chce (rw), zatímco skupina a ostatní do něj mohou jen vstoupit a vypsát si jeho obsah (r-x).

Právo	Hodnota
r--	4
-w-	2
--x	1

Další důležitá věc je vyjádření práv v osmičkové číselné soustavě. Vizte tabulku. Například **rw-** lze zapsat jako 6, protože jednotlivá čísla se sčítají. Z toho vyplývá, že 3 je -wx, 5 je r-x a 7 je rwx.

Základní oprávnění souboru či adresáře lze tedy vyjádřit trojčíselným číslem, přičemž první číslo udává práva vlastníka, druhé skupiny a třetí ostatních uživatelů. Například pro spustitelné soubory se často používá 755, tedy **rw-r-xr-x** – vlastník má neomezená práva a ostatní mohou soubor jen číst nebo spustit.

Je důležité uvědomit si, že vlastník své soubory může smazat, i když k nim nemá právo zápisu (w). Třeba příkaz **rm** se sice standardně při pokusu o smazání zeptá, nicméně lze to a jiné programy se ptát nemusí. Další výjimka je ta, že superuživatel (root) má ke všem souborům právo minimálně rw- (6), a to i v případě, kdy logicky spadá do skupiny s právy --- (0).

Tolik tedy k právům. Co se týče příkazu **ls** a záhadného prvního písmena v prvním sloupci na začátku každého řádku jeho výpisu (****), jednotlivé znaky mají následující význam:

Znak	Význam	Čeho je znak zkratkou
-	obyčejný soubor	-
b	soubor blokového zařízení	b lock device
c	soubor znakového zařízení	c haracter device
d	adresář	d irectory
l	symbolický odkaz	symbolic l ink
p	pojmenovaná roura	named p ipe
s	unixový socket	unix s ocket

chmod

[link](#)

Program **chmod** (zkratka angl. **change mode**, tedy *změnit práva*) slouží ke změně oprávnění souborů a adresářů.

```
# nastaví práva souboru "skript.sh" na 755 (rwxr-xr-x)
chmod 755 skript.sh
```

Při vysvětlování práv výše jsem psal, že je dobré pamatovat si tučné zkratky, tak si v případě potřeby osvěžte, co znamená **u**ser, **g**roup a **o**thers. Jiný způsob, jak u souboru změnit práva, totiž využívá právě těchto zkratk a kombinace písmen r, w, x. Následuje ukázka, na které tento způsob předvedu.

```
chmod u+x soubor.run
```

Písmeno **u** signalizuje, že měnit se budou práva vlastníka. Znak **+** znamená, že práva se budou přidávat (- práva odebírá a = přiřazuje). Další znaky (v tomto případě jen **x**) symbolizují práva, která se budou měnit. Příkaz tedy přidává vlastníkovu právo ke spuštění souboru s názvem *soubor.run*.

```
# nastaví skupině rwx (7) pro „soubor.txt“
chmod g+rwx soubor.txt

# pokud měl soubor práva 777, po tomto příkazu budou 640
chmod u-x,g-wx,o-rwx "soubor s mezerami v názvu"

# nastaví práva 764 souboru „filename.ext“
chmod u=rwx,g=rw,o=r filename.ext
```

Jediným standardizovaným prepínačem příkazu **chmod** je -R, který slouží k rekurzivní změně práv.

```
# nastaví adresáři „adresar“ a veškerému jeho obsahu práva 755
chmod -R 755 adresar/

# odebere vlastníci skupině právo k zápisu do adresáře „x“ a jeho obsahu
chmod -R g-w "x"
```

Zde se hodí upozornit, že **chmod** s prepínačem -R dovede (především spuštěný pod rootem) velice snadno znehodnotit systém, a to takovým způsobem, že je pak podstatně jednodušší systém smazat a nainstalovat jej znova (či v lepším případě obnovit jej ze zálohy), než se pokoušet katastrofu napravit. Pozor si dávejte zvláště při zadávání absolutních cest, protože největší škodu by příkaz napáchal logicky tehdy, dostal-li by kořenový adresář (/) jako argument.

GNU verze příkazu má oproti standardu navíc několik dalších praktických prepínačů. Dva ovlivňující výpis programu:

```
# -v vypíše informace o každém zpracovaném souboru
chmod -v 755 a b c
mode of `a' changed to 0755 (rwxr-xr-x)
mode of `b' changed to 0755 (rwxr-xr-x)
```

```
mode of `c' changed to 0755 (rwxr-xr-x)
```

```
# -c je jako -v, ale vypisuje info jen tehdy, změní-li se práva a  
# následující příkaz tedy nevypíše nic (práva všech souborů již jsou 755)  
chmod -c 755 a b c
```

Dále je zde přepínač `--reference`, díky kterému můžeme místo konkrétních práv zadat cestu k souboru, jehož práva budou použita

```
# nastaví adresáři „dir“ práva kořenového adresáře  
chmod --reference=/ "dir"
```

Přepínač `--preserve-root` nedovolí spustit příkaz v rekurzivním režimu na kořenový adresář. Pozor, na rozdíl např. od `rm` je u `chmod` u výchozí chování takové, jako by byl zadán přepínač `--no-preserve-root`.

chown

[link](#)

Příkaz `chown` slouží ke změně vlastníka a skupiny souborů a adresářů. Název je zkratkou anglického **change owner**, tedy *změnit vlastníka*.

```
# změní vlastníka souboru „soubor.txt“ na „jmeno_uzivatele“  
chown jmeno_uzivatele soubor.txt  
  
# změní vlastníka souboru „soubor.txt“ na „dave“ a skupinu na „home“,  
# oddělovač vlastníka a skupiny může být znak : nebo .  
chown dave:home "soubor.txt"
```

Místo jména vlastníka a skupiny lze zadat také číslo vlastníka (UID) či skupiny (GID).

Existují ještě další dvě zvláštní použití.

```
# změní skupinu souboru „soubor.txt“ na „www“  
chown :www soubor.txt  
  
# změní vlastníka souboru „x“ na "admin" a  
# změní skupinu na implicitní skupinu uživatele „admin“  
chown admin: x  
  
# toto nefunguje, protože tento formát zápisu je platný  
# jen tehdy, když zadáte jméno uživatele (a ne UID)  
chown 1000: x  
chown: invalid spec: `1000:'
```

Co se týče přepínačů, je na tom `chown` podobně jako výše popisovaný `chmod`, tedy `-R` je standardní přepínač aktivující rekurzivní režim a společná GNU rozšíření jsou také stejná. Příkaz `chown` má ale ještě

další standardní i GNU přepínače.

```
# nastaví adresářům „a“, „b“, „c“ a jejich obsahu  
# vlastníka „dave“ a skupinu „home“ a navíc vypíše  
# informaci o každém změněném souboru či adresáři  
chown -c -R dave:home a b c
```

Standardní přepínač -h zakáže následování symbolických odkazů (symlinků) a mění přímo jejich vlastnictví (na systémech, kde to je podporováno – obvykle to smí provádět pouze root). Dále jsou zde standardní přepínače ovlivňující chování rekurzivního režimu (-R), a těmi jsou -H, -L a -P. Přepínač -H způsobí následování symlinku zadaného na příkazovém řádku. Je-li zadán přepínač -L a program narazí na symlink odkazující na adresář, spustí rekurzi na odkazovaný adresář. Přepínač -P neprochází žádné symlinky. Standard neudává, které chování se má použít jako výchozí, ale GNU používá -P.

Užitečný při rekurzivní změně vlastnictví může být také GNU přepínač --from, jemuž jako argument zadáte vlastníka a skupinu a **chown** pak mění vlastnictví jen u těch souborů, které odpovídají hodnotám zadaným přepínači --from.

```
# rekurzivně změni vlastnictví všech položek v adresáři „/home/petr“,  
# které mají vlastníka „dave“ a skupinu „home“ na vlastníka „petr“ a skupinu „users“  
chown -R --from=dave:home petr.users /home/petr  
  
# rekurzivně změni vlastnictví všech položek v adresáři „/home/dave“,  
# které mají vlastníka „dave“ na „petr“ s jeho implicitní skupinou  
chown -R --from=dave petr: /home/petr/docs
```

chgrp

[link](#)

Příkaz **chgrp** slouží ke změně vlastnické skupiny souborů a adresářů. Název vychází z anglického výrazu **change group** – *změnit skupinu*.

```
# nastaví souboru „x“ skupinu „home“  
chgrp home x  
# téhož lze docílit pomocí:  
chown :home x
```

Program má (až na absenci --from) stejné standardní i GNU přepínače jako výše popisovaný **chown**.

```
# změni skupinu aktuálního adresáře a jeho obsahu na „users“  
chgrp -R users .  
  
# totéž jako předchozí, ale navíc informuje o provedených změnách  
chgrp -cR users .
```