Wszystkie pliki systemów uniksowych posiadają swoje prawa dostępu dla zapisu, odczytu i wykonywania. Jeżeli dotychczas spotykałeś się z systemami Windows na partycjach FAT - możesz pomylić prawa dostępu z pewnymi atrybutami pliku. W Windowsie można było ustawić atrybuty: archiwalny, ukryty, systemowy, tylko do odczytu, ale każdy kto tego Windowsa używał mógł sobie je również dowolnie zmieniać. Ponadto nie można było przydzielić do pliku dowolnego użytkownika itp...

Używając Linuksa zauważyłeś już pewnie, że istnieje podział na przynajmniej dwóch użytkowników - root-a i normalnego. Tego drugiego możemy zrobić, kiedy nam się podoba, root jest tworzony podczas instalacji. Dla roota nie ma ograniczeń. On może ustawiać prawa jak mu się podoba. Ale musi ograniczać je dla pozostałych użytkowników... Dlaczego? Linux jest systemem przystosowanym do sieci. Posiada konta dla wielu użytkowników. Z komputerem można się łączyć zdalnie lub lokalnie. Konta też mogą być udostępniane zdalnie (shell - SSH). Jak widać swoboda użytkowania nie wchodzi tutaj w grę.
Linux umożliwa przydzielanie praw dostępu do plików użytkownikom i grupom. W systemie istnieje podział na właściciela pliku, grupy przypisanej do pliku i innych użytkowników (nie będących ani przydzieloną grupą, ani właścicielem pliku).
W jaki sposób dowiedzieć się jakie prawa ustawiono dla danego pliku? Tutaj nie ma problemu. Wpisz ls -l nazwa_pliku. Dla przykładu:

```
home:/temp/katalog# ls -l example
-rw-r--r--  1 root root 0 2006-04-29 17:27 example
```

Po wpisaniu tego polecenia, wyświetlono informacje dotyczące pliku example. Pierwsza informacja z lewej to właśnie prawa dostępu, druga do ilość powiązań, trzecia z kolei to użytkownik (właściciel), a czwarta to grupa przypisana do tego pliku. Na końcu widzimy jeszcze datę i czas modyfikacji oraz nazwę pliku.

Jak interpretować informacje dot. praw dostępu?
Weźmy powyższą informację: -rw-r--r--
Jest tutaj 10 znaków.
Pierwszy znak oznacza zawsze typ pliku (np. zwykły plik, albo katalog). Jak widzisz jest to myślnik ("-"). Przyjrzyj się poniższej tabeli:
Znak
Znaczenie
-
zwykły plik
b
specjalny plik blokowy (ang. Block)
c
specjalny plik znakowy (ang. Character)
d
katalog (ang. Directory)
l
małe ("L") - dowiązanie symboliczne (ang. Symbolic link)
p
nazwany potok
s
gniazdo
- w takim razie obiekt to zwykły plik.
Następnie widzimy oznaczenia typu r, w oraz "-". Oprócz tego może również wystąpić x. Litery te oznaczają:
Znak
Angielski
Plik
Katalog
r
Read
prawo do odczytu
prawo do przeszukania zawartości
w
Write

prawo do zapisu
prawo do zmiany zawartości
x
eXecute
prawo do wykonywania
prawo do wejścia do katalogu
- myślnik - oznacza brak danego prawa (występuje w miejscu odpowiedniej litery).
Bierzemy jeszcze raz przykład bez pierwszego znaku: rw-r--r--
Te 9 kolejnych znaków oznaczają po 3 prawa kolejno dla każdego z 3 użytkowników.
Od lewej pierwsze 3 symbole przeznaczone są dla właściciela pliku (ang. owner ale oznaczany jako ang. user - dalej dowiesz się dlaczego), drugie 3 symbole dla grupy przypisanej do tego obiektu, a ostatnie 3 dla innych użytkowników (nie będących właścicielami, ani grupą do której plik czy katalog został przydzielony).
Właściciel katalogu lub pliku to użytkownik, który stworzył ten plik lub któremu go przypisano. Grupa to określona grupa użytkowników (wg. grup można ustalać jakie kto ma prawa bez potrzeby ustalania ich konkretnemu użytkownikowi).
Czyli interpretacja ma się następująco:
"Plik jest zwykłym plikiem, właściciel ma prawo do odczytu i zapisu, grupa tylko do odczytu, pozostali - również tylko do odczytu".
Jak nadać użytkownikowi lub grupie jakiś plik?
Standardowo plik przydzielany jest do użytkownika (i jego grupy), który go stworzył. Jeżeli chcemy zaś właścicielem pliku uczynić innego użytkownika lub inną grupę to przydadzą nam się dwa programy: chown - do użytkownika i chgrp - do grupy. Użycie jest bardzo proste:

chown użytkownik nazwa_pliku - zmienia użytkownika, grupa pozostaje,
chown użytkownik: nazwa_pliku - zmienia użytkownika, natomiast grupę na główną,
chown użytkownik:grupa nazwa_pliku - zmienia użytkownika i grupę,
chown :grupa nazwa_pliku - zmienia tylko grupę,

chgrp grupa nazwa_pliku - zmienia grupę, j/w.
Zamiast dwukropka można wstawić kropkę. Jako grupę można podać albo nazwę grupy, albo identyfikator numeryczny. Pomiędzy użytkownikiem a dwukropkiem oraz dwukropkiem a grupą nie ma odstępów! Jeżeli istnieje potrzeba przydzielenia wszystkich plików do użytkownika i/lub grupy (w obrębie katalogu) można posłużyć się poleceniem chown -r...

chown - z ang. Change Owner - zmień właściciela,
chgrp - z ang. Change Group - zmień grupę

Nadanie praw właścicielowi, grupie i innym
Nadanie i zmianę praw dostępu dla użytkownika lub grupy możemy wykonać na kilka sposobów. Do tego posłuży nam program o nazwie chmod (ang. Change Mode).
Sposób pierwszy - za pomocą liter i znaków
Sposób ten określa ustawienia praw dostępu do pliku takich samych dla wszystkich - tzn. możemy ustalić takie same prawa dla wszystkich albo właścicielowi, grupie czy innym użytkownikom. W ten sposób nie ustawimy praw osobno dla właściciela, osobno dla grupy itd. (chyba, że użyjemy polecenia kilka razy).
Przykład:

chmod a+w nazwa_pliku
Dzieki temu możemy spowodować nadanie wszystkim prawa do zapisu dla pliku o podanej nazwie.
Wyjaśnienie składni tego polecenia:

1. Określamy kto ma dostac te prawa:
Litera
Znaczenie
a
(ang. all) - wszyscy (użytkownik, grupa, inni)

u
(ang. user) - użytkownik - właściciel pliku
g
(ang. group) - grupa, której przypisano plik
o
(ang. others) - inni

Właściciel określany jest jako user (użytkownik), a nie owner (właściciel), oto dlaczego - nie może być dwóch typów użytkowników zaczynających się na o (owner, others), więc jeden nich jest zastępowany przez user. Programiści wymyślili rozwiązanie, które i tak nie koliduje z prawidłowościami - (u)ser - użytkownik (któremu nadano plik - właściciel), (o)thers - inni użytkownicy w systemie - nie należą do grupy, do której przypisano plik i nie są właścicielami pliku.

2. Określamy znaczenie polecenia:
Znak
Znaczenie
+
nadanie prawa (dodanie)
-
odebranie prawa
=
zastąpienie prawa - kasuje poprzednie i zastępuje nowym

3. Ustalamy prawa:

r, w, x - wiadomo,
u, g, o - ustawie takie same prawa jak ma (u)ser, (g)roup lub (o)thers
t - oznacza, że usunąć katalog może tylko jego właściciel(e) (tzw. Lepki bit - ang. sticky bit); oznacza też plik tekstowy
l - obowiązujące zabezpieczenie (małe "L")
Argumenty możemy ze sobą łączyć, np. chmod ug+rwx nazwa_pliku - nada

właścicielowi i grupie prawo do czytania, zapisu i wykonywania, natomiast dla innych użytkowników (others) pozostaną one bez zmian.
Sposób drugi - za pomocą liczb oktalnych (ósemkowych)
- system oktalny polega na wykorzystywaniu cyfr od 0 - 7 (8 cyfr).
Z tym sposobem napewno zetknęli się już webmasterzy mający strony na serwerze uniksowym, obsługującym język PHP. Polecenie jest bardzo proste, wystarczy wpisać chmod 777 aby ustawić wszystkim prawa do odczytu, zapisu i wykonywania.
Tym sposobem ustalamy różne (lub takie same) prawa dla wszystkich użytkowników. Jak zwykle pierwsze kolejne cyfry od lewej to: właściciel-grupa-inni.
A oto rozkład cyfr:

Znaki
Cyfra
Znaczenie

0
brak praw
--x
1
tylko wykonywanie
-w-
2
tylko zapis
-wx
3
zapis i wykonywanie
r--
4
tylko odczyt
r-x
5

odczyt i wykonanie
rw-
6
pokazuje informacje na temat komendy
rwx
7
zapis, odczyt i wykonywanie

Oczywiście w przypadku katalogów znaczenie zmienia się odpowiednio. Jest jeszcze możliwość ustawienia czwartej cyfry. Ustawia się ją przed tymi trzema odnoszącymi się do u, g i o. A oto ich oznaczenia:

1 - to samo co prawo t - lepki (ang. sticky) bit,
2 - ustawianie ogólnego zabezpieczenia (lub ustawia identyfikator grupy podczas uruchomienia),
4 - ustawienie identyfikatora użytkownika podczas uruchamiania

Zatem polecenie z chmod ustawiające "lepki bit" i jakieś prawa może wyglądać tak:
chmod 1770
Maska
Gdy w systemie Linux jakiś plik jest tworzony musi mieć od razu ustalone prawa dostępu. Dla przykładu możesz utworzyć dowolny plik i sprawdzić jakie prawa posiada (polecenie ls -l nazwa_pliku) - w każdym razie jakieś ma.
Istnieje polecenie, które pozwala określić jakie prawa nie mają być nadawane domyślnie dla nowo utworzonego pliku - umask.
Przykład:
umask 444
Będzie oznaczać, że standardowo nie bedą przyznawane prawa do odczytu. Czyli jeśli stworzysz teraz nowy plik, to będzie on miał takie prawa: -w--w--w-. Zwykle maska ustawiana jest na 022 (rw-r--r--). Jak widać 0 ustawia zapis i odczyt (a raczej nie wyłącza go). Można zauważyć, że

prawa do wykonywania są pomijane!
Możliwe cyfry to:

Cyfra
Znaczenie
0
brak ograniczeń praw (zapis i odczyt)
2
wyłącza zapis (ustawia tylko odczyt)
4
wyłącza odczyt (ustawia tylko zapis)
6
wyłacza zapis i odczyt (brak praw do pliku)

Podane opcje są prawidłowymi, ale można też użyć standardowych 0-7 (zostanie wyłączona opcja zapis i/lub odczyt - wykonywanie i tak zawsze będzie wyłączone).
Tłumaczenie dla ambitnych
Zwykle standardowe prawa dostępu (mode) ustawione są na 666 (rw-rw-rw-). Poleceniem umask ustawiamy jakie prawa odejmujemy od standardowego, czyli od 666 - ustawiamy tzw. maskę. Zatem jeśli maska ustawiona jest na 444 (umask 444) to układamy równanie 666 - 444 = 222 [mode] co nam daje:

-w--w--w-.

Jeżeli w równaniu wyjdzie jakieś prawo wykonywania - zostanie ono ustawione jako brak (-). Jeśli przekroczymy zakres odejmowania, czyli odejmiemy większą liczbę od mniejszej to prawa ustawione będą ustawione na 000 (---------), czyli w normalnym wypadku wszelkie prawa będą odejmowane począwszy od mode 666.
Polecenie umask bez podanych parametrów wyświetli nam aktualny status

maski.
domena@home:~$ umask
0022
Jak można zauważyć mamy tutaj 4 cyfry.
Zamiast 0 mogą pojawiać się inne wartości:

Cyfra
Opis
1
Ustawiony lepki bit
2
Ustawiony bit SGID
4
Ustawiony bit SUID
Jeżeli chcesz na stałe ustawić wartość umask, zmień jego wartość w pliku /etc/profile (lub profiles); dla konkretnego użytkownika w systemie Linux - w .bashrc, natomiast Unix - .profile (w katalogu domowym użytkownika).
Ponadto poleceniem umask -S można sprawdzić jakie prawa będą przydzielane standardowo (nie podaje maski):
domena@home:~$ umask -S
u=rwx,g=rx,o=rx
Cały czas posługiwałem się określeniem pliku, ale jeśli chodzi o prawa odczytu, zapisu i wykonania to ich odpowiedniki dla katalogów nie ulegają zmianie (przeszukiwanie, zmiana zawartości, wejście).