

Polecenie `chmod` – jak *change mode*

Zmienia uprawnienia dostępu do pliku lub katalogu. W systemach Unix stosuje się uprawnienia do:

- Odczytu (Read)
- Zapisu (Write)
- Uruchomienia (eXecute)

Stąd w listingu katalogów (`ls -l`) oznaczenia **r**, **w**, **x**:

```
-rw-r--r-- 1 mcbx mcbx 18349 sty 11 2016 COPYING
drwxr-xr-x 3 mcbx mcbx 4096 lut 24 13:46 lib
drwxr-xr-x 2 mcbx mcbx 4096 lut 24 13:46 licenses
```

Pierwszy znak określa typ obiektu (- - plik, d – katalog, l – łącze, p – strumień itp.). Kolejne trójki znaków określają uprawnienia dla różnych grup użytkowników systemu (litera oznacza ustawienie uprawnienia, kreska - brak):

- Pierwsza trójka – właściciel pliku (**user**)
- Druga – grupa użytkowników, w której właściciel się znajduje (**group**)
- Trzecia – reszta użytkowników systemu (**other**)

Przy czym administrator – root – ma i tak technicznie dostęp do wszystkich danych.

Poleceniem `chmod` możemy przyznawać i odbierać uprawnienia w dwojaki sposób:

- **Notacja literowa:**

np.

```
chmod g+w plik.txt
```

Przyzna prawo do zapisu wszystkim członkom grupy. Prócz oznaczeń **u**, **g**, **o** Można również:

```
chmod a-x plik
```

Usunie prawo do uruchomienia wszystkim użytkownikom z właścicielem włącznie. Oczywiście właściciel będzie mógł sobie to później przyznać.

czyli:

- Pierwsza litera: Komu – **u** – user, **g** – group, **o** – other, **a** – all
- Drugi znak: Co - + - przyznanie uprawnienia, - - odebranie uprawnienia, = - przypisanie konkretnych uprawnień.
- Trzeci znak: Uprawnienie: **r**, **w**, **x**.

Przy czym znaki możemy łączyć, stąd polecenie:

```
chmod ug-wx
```

jest całkowicie prawidłowe i odbiera użytkownikowi oraz jego grupie prawa do zapisu i uruchomienia pliku.

- **Notacja ósemkowa:**

`chmod` przyjmuje wtedy 3 cyframi kolejno oznaczające uprawnienia dla właściciela, grupy i innych. Znaczenie cyfr jest następujące:

```
7  r w x
6  r w -
5  r - x
4  r - -
3  - w x
2  - w -
1  - - x
0  - - -
```

Przykładowo:

```
chmod 754 skrypt.sh
```

oznacza ustawienie pełnych (**rwx**) uprawnień dla właściciela pliku, prawa do odczytu i uruchomienia dla grupy i wyłącznie uprawnienia do odczytu dla pozostałych użytkowników.

W przypadku **katalogów** uprawnienia do odczytu i zapisu są interpretowane nieco inaczej.

- Uprawnienie do odczytu umożliwia odczytanie spisu obiektów w katalogu. Odebranie tego uprawnienia uniemożliwia pobranie listy plików (poleceniem ls), ale znając położenie pliku wciąż można z niego korzystać.
- Uprawnienie do zapisu umożliwia zapis w katalogu nowych plików i usuwanie istniejących.
- Uprawnienie do uruchamiania umożliwia w ogóle wejście do katalogu.

Atrybut +t – sticky bit

Zastosowany na katalogu powoduje, że jedynie właściciel katalogu lub root może zmieniać nazwę lub usuwać pliki wewnątrz, a nie każdy użytkownik mający tam prawo do zapisu i uruchamiania.

Program more

Program pokazuje na ekranie tekst, jaki napływa do niego przez standardowe wejście. Wyświetla go jednak ekran po ekranie, na kolejny ekran możemy przejść naciskając klawisz spacji, o linię przesuwamy klawiszem Return/Enter. Wyjście – klawisz q.

W zasadzie to program wykorzystywany przez polecenie **man**.

Przykład użycia:

```
cat dokument.txt | more
```

Operatorem | (pipe – z ang. potok) wysyłamy wyjście jednego programu jako wejście do drugiego.

Wyjście programu cat – zawartość pliku – zostało przesłane do programu more by ten wyświetlił plik na ekranie strona po stronie.

Użyteczne parametry:

- s – zastępuje następujące po sobie puste linie jedną.
- w – po ostatniej stronie nie zamyka się od razu, ale czeka na naciśnięcie klawisza pokazując w ostatniej linii tekst „[no more]”. Użyteczne w zastosowaniach zautomatyzowanych.

Skrócona forma używania more dla pliku:

```
more plik.txt
```

Warty do zapamiętania wniosek: **Wyjście każdego programu może być przekierowane do pliku albo na wejście innego programu.** Jest to często wykorzystywane tworząc „łańcuszki” filtrujące specyficzne dane z długich list, formatujące teksty czy manipulujące wynikami poleceń diagnostycznych tak, by je łatwo wizualizować.

Program grep – jak *Global Regular Expression Print*

Program wyszukuje w pliku linie, które zawierają tekst pasujący do dostarczonego do programu wyrażenia regularnego.

Wejściem dla programu może być zarówno plik, jak i polecenie.

Przykłady użycia:

```
ps -ax | grep ssh
```

W liście całości procesów działających na serwerze wyszukuje te, w których pojawia się nazwa ssh. I tak nie są to nasze procesy i nie mamy do nich uprawnień.

Dysponując plikiem nie musimy wydawać polecenia cat (cat plik | grep wyrażenie), grep może odczytać plik:

```
grep '[a-zA-Z0-9]@*.pl' mbox
```

Wylistuje linijki, w których znajdują się adresy e-mail z domeny .pl obecne w wiadomościach skrzynki pocztowej niezależnie od ich pozycji w nagłówku.

Użyteczne parametry:

```
grep -v wyrażenie – zwraca wszystkie linie niepasujące do wyrażenia.
```

Rozbudowane wyrażenia regularne będą omawiane przy okazji języka Perl, gdzie są podstawowym narzędziem do sprawnej edycji łańcuchów tekstowych.

Przykładowe wyrażenia regularne

* - dowolny ciąg znaków

[abc123] – jeden znak ze zbioru

[^abc] – jeden znak, ale nie a, b czy c.

[a-z] – jak w poprzednich zadaniach.

{11,12,13} – jeden z łańcuchów.

Większość tych wyrażeń można również używać do nazw plików, lecz w niektórych dystrybucjach można znaleźć któreś wyrażenie nie działające.

cut – narzędzie do rozdzielania kolumn

Użycie:

```
cut -c 4-10 dane.txt
```

uzyska z pliku dane.txt znaki (c – jak *characters*) na pozycji 4-10 i wyprowadzi je na standardowe wyjście.

Tekst dzielony (d - *delimiter*):

```
cut -d , -f 2 dane.txt
```

Uzyska drugą kolumnę z pliku, którego kolumny dzielone są znakami przecinka i wyprowadzi wynik na standardowe wyjście.

Jeżeli tekst dzielony jest jakimiś znakami specjalnymi, należy ująć je w apostrof lub cudzysłów:

```
cut -d ';' -f 2-3 dane2.txt
```

Definiowanie przedziałów:

Zarówno przy cięciu na znakach jak i kolumnach można uzyskać kilka kolumn korzystając ze znaku – (kreska) jako przedziału. Jeżeli liczbę po prawej stronie pozostawimy pustą (np. -f 3-), program drukuje kolejne kolumny do końca linii.

sort – narzędzie do sortowania linii.

Przyjmuje plik, sortuje linijki, wypisuje na standardowe wyjście.

Użycie:

```
sort dane.txt
```

Parametry:

-f – traktuje małe litery jak wielkie

-r – sortuje w odwrotnej kolejności (malejąco)

-n – próbuje sortować według wartości liczby na początku dokumentu.

Zadania:

W jednej linijce wyświetl listę plików w bieżącym katalogu posortowaną rosnąco według rozmiaru.

Podpowiedź: Poeksperymentuj z narzędziem cut i trybem znakowym -c dla uzyskania różnych kolumn startowych. Po uzyskaniu listy zaczynającej się od rozmiaru możliwe jest sortowanie z parametrem -n