

Wykład 2

System operacyjny **Linux** cd.

Skrypty powłoki

Prawa dostępu

Dowiązania

Edytor *vi*

Jeszcze raz przykłady operacji kopiowania

- cp p1 p2** - źródło p1 i kopia p2 w tym samym katalogu (bieżącym) – nowa nazwa
- cp ../p1 .** - źródło w katalogu nadrzędnym a cel w katalogu bieżącym – ta sama nazwa
- cp p1 ..** - źródło w katalogu bieżącym, a cel w nadrzędnym – ta sama nazwa p1
- cp ../p1 ./p2** - źródło w katalogu nadrzędnym a cel w katalogu bieżącym – nazwa p2
- cp KAT1/p1 KAT2/p2** - źródło p1 w katalogu podrzędnym KAT1, a cel w innym podrzędnym KAT2 - nowa nazwa p2
- cp ../KAT1/p1 ../KAT2/p2** – to samo co powyżej

find *katalog* *-typ* *wzorzec* – wyszukiwanie plików

```
find / -name 'p*'
```

szukaj w całym systemie plików
(poczynając od katalogu głównego /)
plików o nazwie na literę p

```
find ~ -name 'p*'
```

szukaj tylko w katalogu domowym i
strukturze folderów do niego podrzędnych

cmp – porównywanie plików

```
cmp p1 p2 - znajdzie 1-szą różnicę
```

```
cmp -l p1 p2 - znajdzie wszystkie różnice
```

Strumienie – przekierowanie wejścia/wyjścia (domyślnie wejście to **klawiatura** a wyjście to **ekran**):

ls -l > dir.txt

zawartość katalogu zapisz do pliku

ls -l >> dir.txt

dopisanie listy do pliku

sort < dir.txt

sort (program sortujący) pobiera dane z pliku

cat > plik

plik ze standardowego wejścia (klawiatura) do nowego pliku – już znamy

cat plik1 plik2>plik3

połączenie (*catenate*) plików

Potoki – efekty jednego polecenia przekazane dla innego polecenia, przykładowo:

ls -l | more

Można łączyć strumień i potok

sort <plik.txt | more

Prawa dostępu

Każdy plik (katalog) ma swojego właściciela indywidualnego (najczęściej jest to użytkownik, który utworzył plik) oraz właściciela zbiorowego (**grupę**).

Właściciel pliku jest automatycznie członkiem grupy (czasami jest to grupa jednoosobowa).

Użytkownik (user) i grupa (group) posiadają określone **prawa do pliku lub katalogu**

Pozostałym użytkownikom (**others**) mogą też być przypisane określone prawa do danego pliku (katalogu).

Wyjątkowym użytkownikiem jest **root**, który posiada wszystkie prawa dostępu do każdego pliku - może je zawsze zmienić, nawet gdy nie jest właścicielem pliku (katalogu).

W odniesieniu do konkretnego pliku określony użytkownik może posiadać dane prawo dostępu lub też nie.

W systemie wielodostępnym istotna jest organizacja, **selektywna ochrona dostępu** do danych.

W Linuksie (UNIX-ie) wyróżniamy trzy rodzaje praw dostępu do pliku:

- prawo **czytania** (r - **read**)
- prawo **pisania** (w - **write**)
- prawo **wykonywania** (x - **execute**)

Powyższe prawa w stosunku każdego pliku (katalogu) są niezależnie nadawane w trzech kategoriach:

- prawa **właściciela** pliku (user)
- prawa użytkowników należących do określonej **grupy**
- prawa **pozostałych** użytkowników (others)

Informacja o prawach

- **rwx** **rwx** **rwx**

właściciel grupa inni

- plik

d **rwx** **rwx** **rwx**

d to katalog

Nowotworzone elementy otrzymują prawa:

plik: **r w - r w - r - -**

katalog: **r w x r w x r - x**

Ustalanie praw: pierwszy sposób liczbowy

Litera	Znaczenie	Parametr liczbowy
r	prawo odczytu	4
w	prawo zapisu	2
x	prawo uruchomienia	1
-	brak praw dostępu	0

W poleceniu zmiany praw sumujemy liczby dla każdej kategorii użytkowników, przykładowo:

- chmod** 400 plik tylko użytkownik ma prawo czytania
- chmod** 760 plik użytkownik wszystkie prawa a grupa tylko r i w
- chmod** 000 plik całkowity brak praw dla wszystkich
- chmod** 777 plik wszystkie prawa
- chmod** 770 plik brak praw dla innych
- chmod** 774 plik brak praw wykonywania i zapisu dla innych (**rwxrwxr--**)

Drugi sposób ustalania praw – metoda symboliczna

u	użytkownik (user)
g	grupa (group)
o	inni (other)
a	wszyscy (all)
+	nadać prawo
-	zabrać prawo

Przykłady:

chmod o-w plik - odebranie innym prawa zapisu w pliku

chmod g+x plik - nadanie grupie prawa wykonania pliku

chmod a+rw plik - nadanie wszystkim prawa odczytu i modyfikacji treści pliku

Posiadanie prawa dla **pliku**:

- * **czytania** - umożliwia otwieranie pliku i przeglądanie jego zawartości
- * **pisania** - umożliwia otwieranie pliku i modyfikację jego zawartości
- * **wykonywania** - umożliwia wykonywanie programu (jeżeli jest to plik binarny) lub skryptu powłoki

Dla **katalogu**:

- prawo czytania umożliwia listowanie elementów zawartości (polecenie **ls**),
- prawo zapisu - umożliwia tworzenie, modyfikacje (treści, nazw), usuwanie. przenoszenie
- prawo wykonywania - dostęp do katalogu (polecenie **cd** nie działa jeśli nie mamy prawa)

Skrypt powłoki

Tworzymy plik, którego tekstem jest sensowny ciąg poleceń, np.:

```
cat > plik_skryptu  
    mkdir nowy_kat  
    ls -l  
^D
```

Musimy jeszcze **nadać plikowi prawo X** (wykonania) !!!

Wówczas uruchamiamy plik:

```
./plik_skryptu
```

Wykonywane są polecenia będące treścią pliku

UWAGA: w webminal: sh ./plik_skryptu

Dowiązania

Dowiązanie to inna nazwa tego samego pliku (pewna analogia do skrótów w Windows).

Dowiązania dzielimy na **sztywne** wykonywane poleceniem:

ln nazwa_pliku nazwa_dowiązania

i dowiązania **miękkie**, inaczej zwane **symbolicznymi** wykonywane poleceniem:

ln -s nazwa pliku nazwa_dowiązania

i-węzeł (ang. i-node)

Struktura, w której jądro systemu przechowuje informacje o pliku.

Każdy plik ma dokładnie jeden unikatowy i-węzeł - nazwę pliku skojarzoną wprost z **i-węzłem** nazywamy dowiązaniem sztywnym

Każdy plik ma co najmniej jedno sztywne dowiązanie, utworzenie nowego dowiązania **sztywnego** to niejako stworzenie **innej nazwy dla tego samego pliku** – ma ten sam i-węzeł

Dowiązanie **symboliczne** to skojarzenie nazwy pliku nie z jego i-węzłem, lecz z istniejącą już nazwą (ściślej: ścieżką dostępu) pliku – plik dowiązania symbolicznego jest mały i ma inny **i-węzeł** – to jest podobne do „skrótów” w Windows

Polecenie `ls -li` wyświetla dodatkowo numery i-węzłów

```
lxc01@MF102:~/xx
[~] lxc01@MF102 xx]$ cat>plik
jakis tekst
[~] lxc01@MF102 xx]$ ln -s plik pliksymb
[~] lxc01@MF102 xx]$ ln plik pliksz
[~] lxc01@MF102 xx]$ ls -li
total 8
301705 -rw-rw-r-- 2 lxc01 lxc01 12 Oct 14 16:47 plik
301707 lrwxrwxrwx 1 lxc01 lxc01 4 Oct 14 16:47 pliksymb -> plik
301705 -rw-rw-r-- 2 lxc01 lxc01 12 Oct 14 16:47 pliksz
[~] lxc01@MF102 xx]$ rm plik
[~] lxc01@MF102 xx]$ cat pliksz
jakis tekst
[~] lxc01@MF102 xx]$ cat pliksymb
cat: pliksymb: No such file or directory
[~] lxc01@MF102 xx]$ ls
pliksz pliksymb
[~] lxc01@MF102 xx]$
```

Jak widzimy dowiązanie sztywne ma ten sam numer i-węzła co **plik**

Po skasowaniu pliku dowiązanie symboliczne jest nieaktywne, a dowiązanie sztywne pod inną nazwą identyfikuje skasowany plik

Wnioski praktyczne

Jeśli usuniemy plik oryginalny:

- dowiązanie sztywne jako inna nazwa nadal istnieje i mamy dostęp do pliku (katalogu),
- dowiązanie symboliczne (jak skrót w Windows) nie odnajdzie pliku (katalogu)

Edytor VI

Uruchomienie edytora z wiersza poleceń: - dla utworzenia nowego dokumentu:

vi

- z wczytaniem do edytora istniejącego dokumentu:

vi *nazwa_pliku*

Edytor **vi** posiada tryby:

- tryb wprowadzania tekstu (INSERT)
- tryb wydawania poleceń (uruchamiany klawiszem **ESC**)
- tryb specjalny ex (: z trybu ESC)

Klawiszem ESC przechodzimy do trybu wydawania poleceń (aktualny tryb widoczny na dole ekranu) - odtąd każdy wciśnięty klawisz będzie poleceniem **vi**.

Powrót do trybu pisania - polecenie klawiszem "i" (lub innymi – np. "a" , "A" , "O" , "o")

Składnia ogólna poleceń (tryb ESC)

[bufor] [licznik] [operator] obiekt

Bufor, licznik i operator są opcjonalne.

Obiekty:

- początek wiersza - **0**
- koniec wiersza - **\$**
- słowo - **w**
- wiersz (nie ma oznaczenia; powtarza się dwukrotnie operator)
- znacznik - **'#** (gdzie # to dowolny znak od a do z)

Ruchy kursora - **h**(lewo), **j**(dół), **k**(góra), **l**(prawo)

Operatory w trybie ESC

d - kasuj (do bufora roboczego)

c - zmień

y - zapamiętaj (yank)

G – skok na koniec pliku

b - skocz do poprzedniego słowa

o - utwórz nowy wiersz pod kursorem (przechodzi do trybu wstawiania)

x - usuń znak na który wskazuje kursor

p – wklej zawartość bufora

u – cofnij zmianę

i inne

Przykłady operacji:

dw - skasowanie do końca słowa (operator d obiekt w) - wycina do bufora (jak schowka Windows)

yw - zapamiętaj do końca słowa (do bufora)

yy - zapamiętaj aktualny wiersz

p – wkleja uprzednio zapamiętany w buforze tekst

4dd – skasuj cztery wiersze (dodatkowo licznik)

ma - utwórz znacznik (marker) o nazwie **a** znaczniki mogą mieć oznaczenia od a do z)

d'a - skasuj wszystko od bieżącej pozycji kursora do znacznika **a** - wytnij do bufora

's - skocz do znacznika **s** (uprzednio utworzonego)

Wykorzystanie buforów roboczych

Bufory robocze oznaczane są literami od a do z, a ich wykorzystanie określa znak cudzysłowu (").

nazwa bufora ("x) na początku, a potem operacja.

Przykłady:

"f17yy - powoduje skopiowanie 17 wierszy do bufora roboczego f, bez ich kasowania,

"h6dw - kopiuje do bufora h 6 słów, kasując je z pliku.

Do odtworzenia zawartości tych buforów roboczych w innym miejscu pliku (lub w innym pliku) należy użyć :

"fp

"hp

Polecenia w trybie **ex**

Zmiana trybu - polecenie **:** (lub **/**)

Przykłady:

/tekst znajduje w dokumencie *tekst*
(kolejne/poprzednie wystąpienia można odnaleźć przez
n lub N)

s/tekst1/tekst2/ zamienia w wierszu pierwsze
wystąpienie tekst1 na tekst2

W nazwa zapisuje zmiany do pliku *nazwa*

q! quit - wychodzi z vi bez zapisywania zmian

wq write - zapisuje zmiany w pliku o bieżącej
nazwie i wychodzi