ssh (np. unixlab)

finger

wall (write to all)

ls -a

ls -l

ls -la

ls -l /dev

ls -ld /dev .bashrc

-

d

l

c

b

p

\*

?

[abc]

[0-9]

[^xyz]

ls .\*

ls -ld [N]\*

ls -d [A-Z]\* zwraca wszystkie nazwy zaczynające się od wielkiej litery

ls -ld ??? zwraca wszystkie nazwy mające 3 litery

ls -ld \*[0-9]\* zwraca wszystkie nazwy mające co najmniej jedną liczbę

mkdir katalog (tworzy nowy katalog)

touch - "dotyka plik", a jeśli pliku nie ma, to go tworzy. Aktualizuje jego modification time bez otwierania pliku

cd

cd .. bezpośredni nadkatalog

. katalog bieżący

stat

rmdir katalog – usuwa katalog

ls > pl1.txt – utworzenie pliku pl1.txt z zawartością wyniku polecenia ls

cp – kopiowanie plików

mv – move

ln – link

rm – remove

Argumenty:

1. Źródło(na których plikach wykonujemy operacje) 2. Cel (przeznaczenie tej operacji, co chcemy z tym

plikiem zrobić), np. cp pl1.txt kopia\_pl1.txt

cmp – compare

cp pl1.txt PK – skopiowanie pliku pl1.txt do podkatalogu PK. Plik będzie miał tę samą nazwę. Dla

pewności można zrobić: cd pl1.txt PK/

Ponieważ gdyby nie istniał taki katalog, to zostałby utworzony plik o nazwie PK. Przy użyciu ukośnika

nie trzeba się o to martwić.

mkdir PK/PPK/PPPK – tak nie można robić

mkdir PK

mkdir PK/PPK

Trzeba tworzyć je po kolei

cp pl1.txt PK/kopia\_pl1.txt – tak można

cp pl1.txt kopia\_pl1.txt pl2.txt PK – kopiowanie wielu plików naraz do katalogu PK. Warunek jest taki:

ostatnie słowo ma być nazwą katalogu

cp \*pl1.txt PK – Shell zinterpretuje tę gwiazdkę (wyrażenie regularne). Wszystkie pliki pasujące zostaną

skopiowane do katalogu PK. To interpretuje Shell, a nie polecenie cp.

mv pl1.txt kopia\_pl1.txt – to jest po prostu zamiana nazwy

mv pl1.txt PK/

mv pl1.txt PK/kopia\_pl1.txt

mv pl1.txt kopia\_pl1.txt PK – przeniesienie tych plików do podkatalogu PK?.

mv \*pl1.txt PK

mv \*pl1.txt

ls -i informacje o plikach, jest także numer i-węzła. Po przeniesieniu do innego katalogu i-węzeł się nie

zmienił. Natomiast po przeniesieniu do katalogu /tmp/ zmienił się.

ln pl1.txt kopia\_pl1.txt – utworzenie połączenia twartego? Oba pliki mają ten sam i-węzeł. Licznik

dowiązań mówi nam, ile nazw odnosi się do jednego i-węzła.

ln pl1.txt /tmp/kopia

ln PK linkPK

ln -s xyz slink - dowiązanie

ls -i wyświetla nr i-węzła

Polecenie rm. Ma tylko jeden rodzaj argumentu. Usuwa dowiązanie do pliku. Tam, gdzie licznik dowiązań jest większy niż jeden (są twarde dowiązania), to plik dalej istnieje pod nazwą, która nie została usunięta. Zatem rm usuwa dowiązanie. Jeżeli licznik dowiązań jest równy jeden, to można uznać, że usunęliśmy plik

i-węzeł = i-node

rm z opcją -r wytnie całe poddrzewa katalogów

rmdir - polecenie do usuwania PUSTYCH katalogów

rm -ri - to samo co z -r, ale z opcją potwierdzenia usunięcia każdego pliku

prawo do odczytu implikuje prawo do kopiowania

1. Właściciel pliku (tylko on ma prawo zmiany dostępu) 2. Grupa 3. Wszyscy pozostali

rw-r--r--

CHMOD (change mode) - polecenie do zmiany praw dostępu. Można to określić albo numerycznie, albo symbolicznie.

CHMOD

u - user

g - group

o - others

a - all

chmod u+w,g-w //tutaj nie może być żadnej spacji (separator oznacza, że dalej są pliki)

sprawdzian do procesów - 28 kwietnia. Dostęp do plików, wyrażenia regularne

$ - oznacza, że chcę wyświetlić wartość tej zmiennej

echo $?

cat

tty

#słowo - coś w rodzaju komentarza? po napisaniu # reszta jest ignorowana

cat > plik.txt - następnie to, co wpiszemy, będzie zapisane w pliku (nie nadpisane tylko zapisane. Potem Ctrl+C

cat >> plik.txt - tak samo jak powyżej, tylko nadpisywanie - dodawanie

cat << plik.txt - "czytaj swoje standardowe wejście do momentu, aż natrafisz na linię plik.txt". Cokolwiek wpiszę to wypisuje, aż do momentu, gdy wpiszę plik.txt i wtedy przerywa. Rozróżnia wielkość liter

cat plik.txt - odczytanie tego

Deskryptor

Uruchamianie w tle -

ls && ps - jednoczesne uruchomienie 2 procesów. Zostaną wypisane wyniki tego i tego

cc - kompilowanie

ps || ls - wykona się ps, chyba, że jest fałsz, to wtedy wykona się ls

ps | ls - potok. Przypadek, gdzie jeden przekazuje dane bezpośrednio na wejściu drugiego

cc pr.c && ./a.out

sleep 100 &

ps -e

ps -u

ps -u root

ps -f

ps -l

ps -o pid, ppid, uid

kill 25859

kill - zabicie procesu. Totalna eksterminacja

kill -l

-KILL

nice - uruchamia proces z odpowiednio obniżonym priorytetem

renice - pozwala operować priorytetem (obniżać lub podnosić priorytet już istniejących procesów)

nice -10

renice 10