

| | |
|--|-----------|
| I. SYSTEM PLIKÓW..... | 2 |
| MORE | 2 |
| PG | 2 |
| HEAD / TAIL..... | 2 |
| CAT | 2 |
| PRZEKIEROWANIE STANDARDOWEGO WEJŚCIA / WYJŚCIA..... | 2 |
| SORT | 3 |
| OPERACJE NA KATALOGACH: MKDIR ; RMDIR ; CD | 3 |
| LS | 3 |
| PRAWA DOSTĘPU | 3 |
| OPERACJE NA PLIKACH..... | 3 |
| TWORZENIE DOWIĄZAŃ..... | 3 |
| POTOK..... | 4 |
| II. NARZĘDZIA PRZESZUKUJĄCE SYSTEM PLIKÓW | 5 |
| GREP..... | 5 |
| FIND..... | 5 |
| COMM | 6 |
| TR..... | 6 |
| WC | 6 |
| DATE | 6 |
| TEE..... | 6 |
| UNIQ..... | 6 |
| AWK | 6 |
| CUT | 6 |
| ECHO | 7 |
| III. SKRYPTY..... | 8 |
| DEFINIOWANIE ZMIENNYCH | 8 |
| STRUKTURY STERUJĄCE | 8 |
| FOR..... | 8 |
| IF..... | 9 |
| CASE..... | 9 |
| WHILE I UNTIL..... | 10 |
| BREAK..... | 10 |
| CONTINUE..... | 10 |
| EXPR..... | 10 |
| READ | 10 |
| IV. PROCESY..... | 11 |
| PS | 11 |
| KILL..... | 11 |
| SLEEP | 11 |
| AT..... | 11 |
| NICE..... | 11 |
| PROCESY DRUGOPLANOWE..... | 11 |
| GRUPOWANIE PROCESÓW | 11 |
| V. KOMUNIKACJA MIĘDZY UŻYTKOWNIKAMI..... | 13 |
| WRITE | 13 |
| WALL | 13 |
| TALK | 13 |
| BŁOKOWANIE TERMINALA | 13 |
| MAIL..... | 13 |

I. SYSTEM PLIKÓW

MORE - służy do wyprowadzania zawartości pliku na ekran. Program wyświetla zawartość pliku aż do zapelnienia ekranu, następnie czeka na naciśnięcie klawisza

- [Enter]- wyświetla kolejną linię tekstu
- [spacja]-wyświetla kolejny ekran
- [b]-wyświetla poprzedni ekran
- [=]-podaje numer linii znajdującej się u szczytu ekranu
- [/]-szukanie miejsca wystąpienia zadanego tekstu
- [v]- uruchomienie edytora vi i automatyczne wczytanie tekstu, po zakończeniu pracy następuje powrót do programu more.
- [h]-help
- [q]-quit
- [-f]-długie wiersz wyświetlanego tekstu nie są przełamywane, na ekranie pojawia się tylko ich fragment
- [- liczba]-wyświetlanie po n linii na ekranie
- [+liczba]-wyświetlanie od n-tej linii

PG - prosty edytor do przeglądania plików tekstowych. Pozwala wyświetlać tekst strona po stronie

- [Enter]- wyświetla kolejną stronę
- n[Enter]-n-ta strona
- +n[Enter]-przeskok o n stron do przodu
- -n[Enter]-przeskok o n stron do tyłu

HEAD / TAIL - pozwalają wyświetlać początek/koniec pliku. Domyślnie wyświetlają 10 pierwszych/ostatnich linii. Można zmienić wartość domyślną na wartość n (opcja -n)

CAT - służy do konkatenacji plików, może być wykorzystane do wyświetlenia plików na ekranie terminala)

- -n przy wyświetlaniu numeruje wiersze
- -b puste wiersze wyłączone z numeracji
- ctrl s wstrzymywanie wyświetlania
- ctrl q wznowienie wyświetlania

Przykł. cat plik1 ; cat plik1 plik2 ; cat plik[12]

Przekierowanie standardowego wejścia / wyjścia

Każdy proces domyślnie korzysta ze standardowego wejścia/wyjścia tj. klawiatury terminala i ekranu terminala. Można przedadresować standardowe wejście i wyjście kierując wynik wykonanego polecenia do podanego pliku

Wykorzystuje się symbole:

< **plik** - przedadresowanie standardowego wejścia, tj. pobranie danych wejściowych z pliku

> **plik** - przedadresowanie standardowego wyjścia, tj. utworzenie pliku i zapisanie w nim tego, co proces wypisałby na standardowym wyjściu

>> **plik** - przedadresowanie standardowego wyjścia z dopisaniem informacji do pliku

Przykł. cat plik1 plik2 >plik3

SORT - program sortujący tekst, umożliwia porządkowanie linii wg dowolnej kolumny (kolejna kolumna = kolejne wyrazy) w kolejności alfabetycznej lub liczbowej, rosnąco lub malejąco

- +2 pominięcie podczas sortowania 2 pierwszych kolumn w każdej linii
- -3 ograniczenie sortowania do 3 kolumny (np. sort +2 -4 plik – posortuje wg 3 i 4)
- +3.1 pominięcie 3 kolumn i 1 znaku kolumny następniej
- -3.3 pominięcie wszystkiego co się znajduje za 3 kolumnami i 3 znakami
- -n sortowanie numeryczne
- -d sortowanie słownikowe
- -f duże i małe litery traktowane są identycznie
- -r odwrócenie porządku sortowania – sortowanie w kolejności malejącej
- -b pominięcie pustych znaków
- -u usunięcie duplikatów klucza sortowania
- -o zapisanie wyniku sortowania w pliku o podanej nazwie.

Operacje na katalogach: mkdir ; rmdir ; cd

LS - wyświetlanie zawartości katalogu, podkatalogu

Wzorce uogólniające:

W konkretnych sytuacjach możemy mieć do czynienia z większą liczbą plików o wspólnych cechach. Stąd shell umożliwia mechanizm generacji listy nazw plików, które odpowiadają podanemu wzorcowi.

- ? odpowiada dowolnemu znakowi
- * odpowiada dowolnemu ciągowi znaków
- [...] odpowiada jednemu ze znaków wymienionych w nawiasach
- [...-- ...] odpowiada dowolnemu ze znaków z podanego zakresu
- [!...] odpowiada każdemu znakowi za wyjątkiem znaków wymienionych w nawiasach
- [...]* odpowiada jednemu ze znaków znajdujących się w nawiasach i napisom rozpoczynającym się od tych znaków
- [!...]* odpowiada dowolnemu znakowi za wyjątkiem znajdujących się w nawiasach i wszystkim napisom nie rozpoczynającym się od tych znaków

Pełna informacja o plikach w katalogu bieżącym:

-l -a -t (sort wg czasu ostatniej modyfikacji)

-u (sort wg czasu ostatniego dostępu)

Prawa dostępu

| | PLIK | KATALOG |
|-------|---|--------------------------------|
| r (4) | do odczytu, umożliwia kopiowanie | do przeglądania |
| w (2) | do zapisu, zmiana zawartości lub usunięcie zawartości | do tworzenia i usuwania plików |
| x (1) | do uruchomienia | prawo dostępu do plików |

Przykł. `chmod (u,g,o,a) +- = (x w r) plik`
`chmod 777plik`

Operacje na plikach

- **-i cp pl_wejściowy pl_wyjściowy** - pliki_ww katalog (musi istniec)
- **mv pl_wejściowy pl_wyjściowy** – zmiana nazwy w obrębie katalogu, przeniesienie do innego i zniszczenie pierwotnego
- **rm pl_wejściowy**
- **cmp file**

Tworzenie dowiązań

- „twarde” **ln nazwa_pliku nazwa_dolaczenia**

W wyniku dowiązania tworzony zostaje nowy plik, który ma własną nazwę, ale jego atrybuty i położenie na dysku jest identyczne z oryginałem. Nie dotyczy katalogów i i plików z innego systemu plików

- „miękkie” **ln -s**

Zamiast tworzyć nowy plik system zaznacza, że wszystkie odwołania do tak utworzone dowiązania mają być faktycznie skierowane do oryginalnego pliku. Symbolicznie można dowiązać katalog

Potok

Przeadresowanie wejścia / wyjścia można użyć do tworzenia potoków, wtedy każdy proces czyta dane z wejścia, które zostało przeadresowane na wyjście procesu poprzedniego.

Przykł.ls -la | more

II. NARZĘDZIA PRZESZUKUJĄCE SYSTEM PLIKÓW

GREP wzorzec_tekstu nazwy_plików

przełata pojedynczy katalog w poszukiwaniu plików zawierających podany ciąg znaków, np.

*Grep „to jest” pl** - przeszukuje wszystkie pliki w bieżącym katalogu których nazwy pasują do wzorca *pl** i z ich zawartości wypisze na ekran te linie, które zawierają tekst „to jest”.

Wzorzec podawany na podstawie wyrażenia regularnego – może być ujęte w „” lub ‘’ – w większości przypadków nie ma różnicy. W wyrażeniach regularnych nast. Znaki mają znaczenie specjalne:

. = dowolny 1 znak
^ = początek linii tekstu
\$ = koniec linii tekstu
[abc]
[a-z]
[^aA]
[^a-z]

Jeśli szukamy znaku specjalnego trzeba zarządzić, by ten znak był potraktowany dosłownie - \ np. \\$
(*?{ }_!;&|^<>\$@#(){}=)

- -i ignorowanie pisowni małych i wielkich liter
- -n poprzedzenie każdego wyprowadzonego wiersza jego numerem w pliku.
- -l wyprowadzenie tylko nazw plików, zawierających wiersze z napisem określonym jako wzorzec
- -v wyprowadzenie tylko tych wierszy pliku, które nie zawierają szukanego tekstu

FIND kat_początkowy spec_plików [akcja]

Przeszukuje drzewo podkatalogów w poszukiwaniu plików spełniających zadane kryteria.

Specyfikacja pliku może obejmować opcje:

- -name „wzorzec” nazwa pliku musi odpowiadać wzorcowi
- -user ident. szukane są tylko pliki, których właścicielem jest użytkownik o danym identyfikatorze
- -grup grupa tylko pliki odpowiadające danej grupie właściciela
- -atime czas tylko pliki których ostatni dostęp nastąpił w ciągu podanego czasu (w dniach)
- -mtime czas tylko pliki modyfikowane w ciągu danego czasu (w dniach)
- -type c = {f,d,l} zwraca wartość true, gdy aktualnie dostępny plik jest plikiem typu c (f- zwykły plik; d – katalog; l – dowiązanie symboliczne)
- -links n zwraca wartość true jeśli aktualnie dostępny plik ma liczbę dowiązanych nazw określoną przez n.
- - (wyrażenie) zwraca wartość true, jeśli wartością wyrażenia jest również true. Nawiasy powinny być cytowane, bo mają specjalne znaczenie. Wyrażenia mogą być łączone wyrażeniami logicznymi:
 - wyrażenie spacja wyrażenie – koniunkcja logiczna
 - wyrażenie –o wyrażenie – alternatywa logiczna
 - !wyrażenie – negacja logiczna

Jeżeli w wyrażeniu występuje parametr n, określający liczbę całkowitą,

n – wartość = n (np. n dni temu, 0 –dzisiaj)

+n – wartość > n (ponad n dni temu)

-n – wartość <n (mniej niż n dni temu)

W poleceniach operuje się symbolem {}, w miejsce którego wstawiana jest pełna nazwa znalezionej pliku.

Nawiasy - odstępy

Akcja:

-print wyprowadza pełne nazwy znalezionych plików

-exec polec \ np. exec rm () \;

-ok. polec podobny efekt jak exec, pyta o potwierdzenie

COMM [-123] plik1 plik2

Program porównuje dwa pliki, które powinny być uprzednio posortowane za pomocą sort i szuka różnic między nimi. Wiersze są wypisywane w 3 kolumnach

Te strony, które znajdują się tylko w pierwszym pliku

Te, które znajdują się w drugim pliku

Znajdujące się w obu plikach

Można zrezygnować z wypisania jednej lub dwóch kolumn przez wprowadzenie odpowiedniej opcji -123

TR [opcje] [napis1 [napis2]]

Polecenie wykonuje zamianę lub kasowanie wskazanych znaków. Znaki wymienione w argumencie napis1 są zamieniane na znaki wskazane na tych samych pozycjach w argumencie napis2. Można tym poleceniem zamieniać wielkie litery na małe, konwertować polskie litery z jednego standardu na inny.

-d usunięcie ze strumienia wejściowego wszystkich znaków podanych w pierwszym napisie

Plik wejściowy podajemy poprzedzając go znakiem <, inaczej program czeka na podanie danych wejściowych z klawiatury. Nazwa pliku wyjściowego musi być różna od nazwy pliku wej., inaczej plik wynikowy jest zerowany.

WC [-lwc] plik

Zliczanie wierszy, słów, znaków, gdy pominięto opcje zliczane są wszystkie wymienione elementy

DATE

wyprowadza bieżącą datę i czas systemowy

DATE [-u] + format – wyświetlanie daty w formacie standardowym lub zgodnie z podanym formatem.

Domyślnie jest to format mmddhhmmyy(miesiąc|dzień|godzina|minuty|2 ost. Cyfry roku) – by ustawić datę wymagane są prawa root-a.

TEE [-a] plik

Rozwidlenie standardowego wyjścia – przesyła zawartość strumienia wejściowego do strumienia wyjściowego i tworzy jego kopię w pliku o podanej nazwie. Polecenie przydatne gdy użytkownik chce np. jednocześnie oglądać wyniki poleceń na ekranie i zachować je w pliku

- -a – dopisanie na końcu podanego pliku bez niszczenia jego zawartości

UNIQ – przetwarza pojedynczy plik, porównując pary kolejnych wierszy, znajdując duplikaty i usuwając je

AWK [‘prosty program awk’] [-f plik programu awk][plik danych]

Polecenie awk uruchamia interpreter języka awk, przeznaczony do przeszukiwania i manipulowania danymi tekstowymi. Umożliwia wyszukiwania w wierszu wierszy spełniających jakiś warunek, dodawanie liczb z kilku kolumn, itp.

CUT - c lista [plik1 plik2 ...]

CUT -f lista [-d znak] [-s] [plik1plik2 ...]

Wycinanie fragmentów wierszy pliku. Zwykle jest to wycinanie odpowiednich kolumn z informacji wyprowadzonych na standardowe wejście.

Lista – określa przedział wartości całkowitych np. 1,2,3; 1-3,

- -clista – określenie pozycji znakowych wycinanych fragmentów wierszy pliku np. c1-72 – pierwsze 72 znaki każdego pliku
- -flista – określenie listy numerów wycinanych pól np. -f1,7 – kopiowanie 1 i 7 pola każdego wiersza.

Domyślny separator – znak tabulacji. Wiersze nie zawierające separatora są kopiowane w całości

- -dznak – określenie znaku będącego separatorem pól
- -s –pomijanie wszystkich wierszy nie zawierających separatora przy wystąpieniu opcji -f

Przykł.

Wyprowadzić nazwy użytkowników uprawnionych do korzystania z systemu (cut -f1 -d: /etc/passwd)

ECHO

Powłoka może zrozumieć znaki uogólniające w sposób nieprzewidziany, stąd polecenie echo pozwala sprawdzić efekt danego wiersza z poleceniem. Wypisuje na terminalu arg. podane w po nazwie polecenia, w rozwiniętej postaci, np. echo pl* to plik1,plik2,...


```
for i in `ls`
do
chmod a+x $1
done
```

IF

```
if lista_polecen_1 lub przy wielokrotnym zagnieżdżeniu:
then lista_polecen_2
else lista_polecen_3
fi
```

```
if ...
then ...
elif ...
then ...
elif ...
else
fi
```

Konstrukcja *if* jest przydatna w połączeniu z poleceniem *test* sprawdzającym czy zaistniał dany warunek. Jeśli warunek jest spełniony, polecenie zwraca wartość true, w przeciwnym przypadku false. Test – jego argumenty i zwracane wartości:

```
test -f plik      0, jeżeli plik istnieje
test -r plik      0, jeżeli plik można czytać
test -w plik      0, jeśli w pliku można pisać
test -d plik      0, jeżeli plik jest katalogiem
test s1 = s2      0, jeżeli napisy s1 i s2 są zgodne.
```

Do porównania liczb można używać operatorów oznaczonych skrótami

```
Gt >   Ge >=  Eq =   Ne !=   Lt <   Le <=
```

W przypadku wyrażeń logicznych można używać spójników logicznych: **!**, **-a**, **-o**

Np. polecenie *test -f plik -a -r plik* – sprawdza, czy plik jest plikiem zwykłym i czy użytkownik posiada prawo do czytania go.

Przykł.

```
if test -w $1 -a -f $1
then cp $1 $1.bak
fi
```

Jeżeli warunek w poleceniu test będzie w [], to słowo test można pominąć i instrukcje test zapisać:

```
if [ $# = 0 ] then ...
```

Przykł.

```
if [ $1 = $2 ] then rm $2
```

Polecenie *if polecenie1 then polecenie2 fi* można zapisać skrótowo:

polecenie1&& polecenie2 – jeśli polecenie1 jest prawdziwe to należy wykonać polecenie2

polecenie1 || polecenie2 - jeśli polecenie1 nie zostało wykonane pomyślnie, to należy wykonać polecenie2

CASE

Case słowo in

Szablon [[szablon] ...) lista_poleceń;; / |oznacza alternatywę

...

esac

Nie jest wykonywany test tego, czy tylko jeden wzorzec odpowiada sprawdzanemu słowu. Pierwsza zgodność określa listę wykonywanych poleceń:

Case \$# in / tu wykonywane będą tylko polecenia występujące po pierwszym a

a) ...

a) .

esac.

Przykł.

Case \$# in

1) **cat >> \$;;**

2) **cat >> \$2 <\$1 ;;**

*) **echo „Błąd”**

esac

WHILE I UNTIL

While lista_poleceń_1

Do

Lista_poleceń_2

Done

While test \$1 /wykonuj cyklicznie jeśli wartość parametru \$1 nie jest pustym znakiem

Do / shift – polecenie wewnętrzne shella, przemianowuje parametry pozycyjne w drodze \

Shift / przesuwania ich w lewo – pierwszy parametr jest usuwany, a \$2, \$3 stają się \$1, \$2

Done

Until lista_poleceń_1 / pętla jest wykonywana tak długo, jak długo jak kod polecenia ostatniego polecenia z

Do /listy_poleceń1 jest nieprawdziwy

Lista_poleceń_2

done

BREAK – Użyte wewnątrz pętli przerywa jej wykonanie i przekazuje sterowanie do pierwszego polecenia występującego po pętli

CONTINUE – wymusza przejście do kolejnej iteracji pętli

EXPR arg operator arg

Obliczanie wartości wyrażeń

operator: +, -, *, /, =, <, itd. &, |

Przykł. max = 0 , max = expr \$max + \$liczba

READ [nazwa(Y)]

Ze standardowego wejścia czytany jest jeden wiersz a następnie zmiennym nazwa przypisuje kolejno wartości odczytanych słów, tworząc ten wiersz. Jeżeli wiersz zawiera więcej słów niż liczba wyspecyfikowanych zmiennych to ostatnia z nich będzie zawierać resztę wiersza

IV. PROCESY

PS

Lista procesów aktualnej powłoki

-f pełna informacja o procesach aktualnej powłoki

-e pełna informacja o wszystkich procesach

KILL PID

Usunięcie procesu o numerze identyfikacyjnym przez właściciela

kill -9 PID - bezwarunkowe przerwanie procesu o identyf. PID

SLEEP (sek)

Zawieszenie wykonywania procesu na liczbę sek sekund

AT czas [data] [przesunięcie]

Wykonanie poleceń w późniejszym czasie

czas – am; pm; noon; midnight; now

data – today; tomorrow, ...

przesunięcie – przesunięcie czasu wykonywania polecenia +n jednostek

jednostka : minutes, hours, days, weeks, months, years

Przykł: at 08.15 am Jan 24

at now + 1 day

at 5 pm Friday next week

NICE

Wykonanie polecenia z innym priorytetem – zwykle obniżonym

Domyślna wartość priorytetu dla procesu = 20.

Im większa liczba tym mniejszy priorytet, możliwe wartości 1 -40

Wykonanie polecenia bez parametru = nice -30

Przykł: nice - 10 cc main.c

Procesy drugoplanowe

Procesy uruchamiane z klawiatury terminala są nazywane pierwszoplanowymi. Powłoka czeka na ich zakończenie i dopiero wtedy przechodzi do przyjmowania kolejnych poleceń

Uruchomienie procesu w tle (znak & po ostatnim parametrze)– powłoka tworzy nowy proces potomny będący powłoką i jej nakazuje wykonanie polecenia, a sama powraca do stanu gotowości.

CTRL +Z = zatrzymanie aktualnie uruchomionego procesu

Grupowanie procesów

(...) – samodzielny proces, duplikacja powłoki

{ ...; } – wykonanie procesu przez aktualną powłokę

Przykł

a)

pwd : /usr/student/inf222

(cd /bin ; ls -la mkdir)

pwd: /usr/student/inf222

b)

pwd : /usr/student/inf222

{ cd /bin ; ls -la mkdir ; }

pwd: /bin

V. KOMUNIKACJA MIĘDZY UŻYTKOWNIKAMI

Jednostronne wysyłanie komunikatów:

WRITE nazwa

- wysłanie komunikatu do innego użytkownika pracującego w systemie
- **ctrl +D** – informacja o zakończeniu pisania treści, **dopiero wtedy** komunikat zostaje **przesłany**

WALL

- podobnie jak write, ale bez podania adresata
- komunikat wysyłany **do wszystkich** pracujących w systemie, którzy **nie mają zablokowanych** terminali.
- komunikat trafia też do nadawcy

Komunikacja dwustronna

TALK nazwa

Przesyłanie informacji w obu kierunkach między użytkami pracującymi w systemie

- rozmowa, gdy drugi użytkownik odpowie (talk nazwa_nadawcy)
- przerwanie rozmowy – ctrl+C

Blokowanie terminala

blokowanie = **mesg n**,

odblokowanie = **mesg y** (* przy numerze terminala dla polecenia finger)

MAIL

- dostarczenie wiadomości niezależnie czy adresat jest zalogowany w systemie
- posiadanie nowych listów w skrzynce – informacja przy logowaniu
- treść listu – wpisana z klawiatury lub z pliku, np.
mail -s Temat darek@cs.put.poznan.pl < plik
- mail – bez parametrów = wyświetlenie zawartości skrzynki
 - s subject
 - h all headers
 - f forward
 - v edit with vi