Pomoce do wykładu z Unixa

Polecenia help

help [-s] [wzorzec] - pomoc dla poleceń i instrukcji wbudowanych bash-a
  `$ help for` ♯ Pomoc dla instrukcji *for*
  `$ help -s lo` ♯ Lista instrukcji rozpoczynających się na *lo*
man opcje [nr_strony] nazwa - podręcznik
  `$ man man` ♯ Pomoc dla *man*
  `$ man 3 printf` ♯ Pomoc dla *printf* ze strony trzeciej man-a
  `$ man -k passwd` ♯ Wypisz wszystkie strony man-a zawierające ciąg *passwd*
  `$ man -a exit` ♯ Wszystkie strony man-a dla *exit*
info [opcje] nazwa - hipertekstowy system pomocy
  `$ info` ♯ Rozpocznij od strony głównej em info
  `$ info libc` ♯ Rozpocznij od strony biblioteki standardowej języka `C`

Pliki i katalogi

cd [opcje] [katalog] - zmień katalog
  `$ cd` ♯ Przejdź do domowego
  `$ cd -` ♯ Przejdź do poprzednio używanego
  `$ cd ..` ♯ Przejdź do nadkatalogu
chmod [opcje] tryb plik - zmień prawa dostępu
  `$ chmod ugo+x skrypt.txt` ♯ Ustaw prawa wyk. dla użytkownika, grupy i innych
  `$ chmod a-rw skrypt.txt` ♯ Odbierz wszystkim prawa czytania i zapisu
  `$ chmod 0666 skrypt.txt` ♯ Ustaw prawa czytania i zapisu
chgrp [opcje] grupa plik - zmień grupę pliku bądź katalogu
  `$ chgrp -R staff katalog` ♯ Zmień rekursywnie (-R) grupę na *staff* dla danego katalogu
cp [opcje] skąd dokąd - kopiuj pliki
  `$ cp p1 p2` ♯ Skopiuj plik *p1* na *p2* (lub do katalogu *p2*)
  `$ cp p1 p2 kat` ♯ Skopiuj pliki *p1* i *p2* do katalogu *kat*
mv [opcje] skąd dokąd - przesuń lub zmień nazwę pliku
  `$ mv p1 p2` ♯ Przesuń plik *p1* do *p2* (lub do katalogu *p2*)
  `$ mv p1 p2 kat` ♯ Przesuń pliki *p1* i *p2* do katalogu *kat*
rm [opcje]... plik... - usuń pliki
  `$ rm -i plik` ♯ Usuń plik ale po potwierdzeniu
  `$ rm -rf katalog` ♯ Usuń rekursywnie (r) katalog bez potwierdzania (f)
rmdir [opcje] katalog... - usuwa katalog(i) ale tylko puste
  `$ rmdir -p a/b/c` ♯ Usuń katalogi a/b/c, a/b, a
mkdir katalog - utwórz nowy katalog
ls [opcje]... [plik]... - wylistuj pliki i katalogi
  `$ ls -a` ♯ Wszystkie pliki (tzn. też postaci .*)
  `$ ls -l` ♯ Pliki w formacie *long*
  `$ ls -R` ♯ Pokaż też podkatalogi
pwd - podaj katalog bieżący
ln [opcje] cel [link] - utwórz link do pliku lub katalogu
  `$ ln work/prace.txt` ♯ Link twardy do *prace.txt* w bieżacym katalogu
  `$ ln -s work/p.txt q.txt` ♯ Link symb. do *work/p.txt* o nazwie *q.txt* w bieżacym katalogu
  `$ ln -s work/p.txt q.txt doc` ♯ J//w ale w katalogu *doc*

Przetwarzanie plików i katalogów, etc...

file [opcje] plik - określ zawartość pliku
cat [opcje] [plik]... - wyświetl zawartość pliku lub plików
od [opcje] [plik] - wyświetl zawartość pliku w różnych formatach
  `$ od plik` ♯ W słowach ósemkowo
  `$ od -b plik` ♯ W bajtach ósemkowo
  `$ od -c plik` ♯ W bajtach i w kodach ASCII
  `$ od -x plik` ♯ W słowach szesnastkowo
find [scieżka] [wyrażenie] - znajdź pliki w strukturze katalogów

$ find . -name oops -print ♯ Wylistuj pliki *oops*
$ find work -name 'core*' -exec rm ♯ Usuń pliki *core** z (pod)kat. *work*
$ find / -ctime +10 -print ♯ Wylistuj pliki niemodyfikowane przez ostatnie 10 dni
$ find . -perm 666 -print ♯ Wylistuj pliki z prawami dostępu 666
$ find . ! -name '*.txt' -print ♯ Wylistuj pliki których rozszerzenie nie jest *.txt*

grep [opcje] wzorzec [plik...] - szukaj wzorca w plikach
$ grep void *.c ♯ Znajdź wzorzec *void* w plikach *.c*
$ grep '^void main' *.c ♯ Wzorzec na początku wiersza
$ grep 'main()$' *.c ♯ Wzorzec na końcu wiersza
$ grep [Ss]ortuj *.txt ♯ Mała lub duża litera
$ grep '[A-Z]....' *.txt ♯ 5-znakowe frazy rozpoczynające się kapitalikiem
$ grep -l WZORZEC PLIK(I) ♯ Wypisz tylko nazwy plików z wystąpieniem wzorca
$ grep -n WZORZEC PLIK(I) ♯ Wypisz numery linii wystąpień wzorca

awk - język do przetwarzania plików tekstowych (man awk)

cmp [-ls] plik1 plik2 - porównaj dwa pliki i pokaż różnice
$ cmp p1 p2 ♯ Pokaż pierwszą różnicę
$ cmp -l p1 p2 ♯ Pokaż wszystkie różnice

cut [opcje]... [plik]... - wytnij kolumny lub znaki z plików
$ cut -c1-3,7 plik ♯ Pokaż znaki 1,2,3 i 7 z każdej linii
$ cut -f3,6- plik ♯ Pokaż pola (separator TAB) 3,6,7,8... z każdej linii
$ cut -f1-3 -d' ' plik ♯ Pokaż pola 1,2,3 (separator spacja) z każdej linii

diff plik1 plik2 - znajdź różnice między plikami

sort [opcje]... [plik]... - sortuj linie pliku
$ sort -b plik ♯ Sortuj z pominięciem białych znaków
$ sort -u plik ♯ Sortuj z pominięciem duplikatów
$ sort -f plik ♯ Nierozróżniaj dużych i małych liter
$ sort -t: -n +2 plik ♯ Sortuj numerycznie, pomiń dwa pierwsze pola, separ. to :
$ sort -n +2 -4 plik ♯ Sortuj num., kluczem jest 3 i 4 pole.
$ sort +3.2nr plik ♯ Omiń trzy pola i 2 znaki, sort. num. i odwrotnie

paste [opcje] [plik]... - połącz linie plików w tabelę
$ paste p1 p2 ♯ j/w
$ paste -d:x p1 p2 ♯ Kolejne separatory kolumn to :x:x...
$ paste -s plik ♯ Połącz linie szeregowo
$ paste -s -d'\n' plik ♯ Linie łączone po dwie

tail [+/-[liczba[lbc][f]]] [plik] - wypisz ostatnie linie pliku
$ tail -20 plik ♯ Ostatnie 20 linii
$ tail +20 plik ♯ Pomiń pierwsze 20 linii
$ tail +50c plik ♯ Pomiń pierwsze 50 znaków

head [-[liczba[c]] [plik] - wypisz pierwsze linie pliku
$ head -10 plik ♯ Pierwsze 10 linii
$ head -2kc plik ♯ Pierwsze 2k (czyli 2048) znaków

uniq [-cdu] [[plik] plikwy] - usuń sąsiadujące powtórzenia linii
$ uniq -c plik ♯ Wszystkie linie z liczbą powtórzeń
$ uniq -d plik ♯ Tylko powtórzone linie
$ uniq -u plik ♯ Tylko jednoznaczne linie

wc [-lwc] [plik] - policz znaki (c), słowa (w) i linie (l) w pliku
$ wc -l plik ♯ Tylko linie

sed -n [polecenie ed] [plik] - wykonaj operacje na pliku
$ sed '2d' plik ♯ Usuń drugą linie
$ sed '3,7 s/tree/graph/' plik ♯ Zamień 1-sze wystąpienie tree na graph w liniach 3..7
$ sed 's/[Yy]aer/Year/g' plik ♯ Zamięn wszystkie wystąpienia Yaer lub yaer na Year
$ sed -n '$p' plik ♯ Wypisz ostatnią linię
$ sed -n '/can/p' plik ♯ Wypisz wszystkie linie z can
$ sed -n '/can/!p' plik ♯ Wypisz wszystkie linie bez can
$ sed '/zvx/ s/ab/AB/g' plik ♯ Zmień ab na AB ale tylko w liniach z zvx
$ sed 's/\(.*\).mp3/\1/g' plik ♯ Usuń .mp3. /n == wyr. reg. z n-tego \(*regexpr*\)
$ sed -f skrypt plik ♯ Wykonaj skrypt sed-owy (zbiór poleceń sed-a z pliku txt) na pliku

dd [opcja=val]... - kopiuj i konwertuj pliki
$ dd if=plik1 of=plik2 conv=lcase ♯ Zamień duże litery na małe, wynik w *plik2*

join [opcje] plik1 plik2 - połącz pliki wg wspólnego pola

split [opcje] [input [prefix]] - Podziel plik na części

$ split -b 1k plik ♯ Podziel plik na 1kB kawałki o nazwach xaa, xab,...
$ split -b 2m plik dd ♯ Podziel plik na 2MB kawałki o nazwach ddxaa, ddxab,...
tr [opcje]... zestaw1 [zestaw2] - zamień lub usuń znaki
$ tr ab XY ♯ Zamień a na b, X na Y
$ tr "[A-Z]" "[a-z]" ♯ Zamień duże litery na małe
$ tr -d " \n" ♯ Kasuj (-d) spacje i newline
nl [opcje]... [plik]... - ponumeruj linie w pliku
spell [opcje] [plik] - sprawdź słowa z pliku (słownik angielski)
gzip plik - skompresuj plik
$ gzip zad.ps ♯ Skompresuj plik, zad.ps zostanie zastąpiony przez zad.ps.gz
gunzip plik.gz - odkompresuj plik.gz
$ gunzip zad.ps.gz ♯ Odkompresuj plik, zad.ps.gz zostanie zastąpiony przez zad.ps
tar opcje... - program do archiwizowania plików
$ tar cvf moje.tar * ♯ Stwórz (c) archiwum (f) moje.tar ze wszystkich plików z *
$ tar czvf moje.tar.gz * ♯ J/w ale skompresowane (z) gzip-em archiwum
$ tar xvf moje.tar * ♯ (x) Odtwórz pliki z archiwum moje.tar
$ tar xzvf moje.tar.gz * ♯ j/w ale ze skompresowanego (z) archiwum moje.tar.gz
du [opcje]... [plik]... - pokaż ile miejsca zajmują pliki
$ du -k ♯ dla bieżącego katalogu - rozmiary w kB
df [opcje]... [plik]... - pokaż wykorzystanie systemów plików
$ df -m ♯ rozmiary w MB

Drukowanie i wyświetlanie plików

less [plik] - przeglądaj plik
Return ♯ Pokaż następny wiersz
Spacja lub f ♯ Pokaż następną stronę
b ♯ Pokaż porzednią stronę
\wzorzec ♯ Szukaj wzorca w tekście
q ♯ Zakończ
tee [opcje] plik... - czytaj ze std. wejścia, zapisz do plików i na std. wyjście
$ grep abc * | tee result.txt ♯ Wyświetl wynik z *grep* i zapisz w *result.txt*
lpr [opcje] [plik] - wyślij plik do druku, ew. formatuj
$ lpr -Plj -# 2 praca.ps ♯ Drukuj plik w dwóch kopiach na drukarce lj
$ lpr -p tekst.txt ♯ Sformatuj plik (nagłówek, data, etc) i wydrukuj
lpq [opcje] - pokaż kolejkę zadań drukarki
$ lpq ♯ Stan domyślnej drukarki
$ lpq -Plj ♯ Stan drukarki lj
lpstat [opcje] - pokaż stan drukarek
lprm [-Pdrukarka] [-] [jobId] - usuń zadanie (job) drukarki
$ lprm 1244 ♯ Usuń zadanie 1244
$ lprm - ♯ Usuń wszystkie zadania
nroff [opcje] [plik] - formatuj plik (tekstowy)
$ nroff zadanie.txt | less ♯ Formatuj do oglądania na ekranie
groff [opcje] [plik] - formatuj plik (tekstowy)
$ groff zadanie.txt > zadanie.ps ♯ Formatuj do postsript
troff [opcje] [plik] - system do formatowania dokumentów

Procesy, system i użytkownicy

at [opcje] czas - uruchom polecenie o określonym czasie
$ at 3.15AM ♯ Uruchom polecenie o 3.15. Dalej wpisujemy polecenia. Koniec - Ctrl-D.
kill [opcje] pid - wyślij sygnał do procesu
$ kill 1441 ♯ Wyślij sygnał TERM (terminacji) do procesu
$ kill -9 1441 ♯ Wyślij sygnał KILL (zabicia) do procesu
nohup [opcje] polecenie - wykonaj polecenie, nie kończ przy wylogowaniu.
$ nohup find -name praca.ps ♯ j/w
ps [opcje] - wypisz informacje o procesach
$ ps -e ♯ Pokaż wszystkie procesy
$ ps -f ♯ Pokaż procesy - pełny wydruk
$ ps -a ♯ Pokaż procesy związane z terminalem

$ ps -t pts/3,pts/4,pts/5 ♯ Pokaż procesy związane z podanymi terminalami
$ ps -u pawel,root ♯ Pokaż procesy podanych użytkowników
$ ps -help ♯ Krótkie opisy opcji
top - pokaż procesy zużywające najwięcej czasu CPU
time [opcje] polecenie [argumenty...] - Pokaż czas działania procesu i zużycie zasobów CPU
$ time tar czvf moje.tgz * ♯ J/w dla polecenia archiwizacji
date [opcje] [MMDDhhmm[[CC]YY][.ss]] - pokaż lub ustaw (tylko root) datę
$ date +%d.%m.%y ♯ Pokaż datę w postaci dzień.miesiąc.rok
$ date 11041539200314 ♯ Ustaw datę 4.11.2003 godz. 15:39.14
$ date 11041539 ♯ Ustaw datę 4.11.rok-bieżący godz. 15:39.00
tty [opcje]... - pokaż nazwę pliku terminala podłączonego do std. wejścia
stty [opcje]... - pokaż lub zmień ustawienia terminala
$ stty -echo ♯ Wyłącz echo
passwd [opcje] [user] - zmień hasło
who - kto jest zalogowany w systemie
who am i - kto jest zalogowany na bieżącym terminalu
whoami - kto jest zalogowany - tylko *username*
finger [opcje] [user...] [user@host...] - znajdź informacje o użytkownikach

Inne polecenia

echo [opcje] ... - wypisz tekst
$ echo "Hello world!" ♯ Hello world!
$ echo -n Ala ma kota ♯ Wypisz bez przejścia do nowej linii
mesg [y|n] - zezwolenie lub zabronienie odbioru komunikatów (por. write)
write user [terminal] - wysłanie komunikatu
mail - wyślij lub odbierz pocztę
pine - popularny program pocztowy
cal [opcje] [miesiąc] [rok] - kalendarz
$ cal ♯ Pokaż bieżący miesiąc
$ cal 2003 ♯ Pokaż kalendarz na rok 2003
bc - kalkulator
latex plik[.tex] - przekształca plik .tex w .dvi
$ latex zadanie.tex ♯ Zrób zadanie.dvi
dvips [opcje] plik[.dvi] - przekształć plik .dvi w .ps
$ dvips -o zadanie.ps zadanie.dvi ♯ Zrób zadanie.ps
gv plik[.ps] - program (w X) do oglądania plików postscript (.ps) i pdf
$ gv zadanie.ps ♯ Oglądaj zadanie.ps
mc [opcje] - Midnight Commander - podobny do Norton Commander-a
$ mc -a ♯ Gdy problemy z wyświetlaniem semigrafiki
ed - edytor liniowy
vi - edytor tekstowy
emacs - mój ulubiony edytor
expr [wyrażenie | opcje] - oblicz lub porównaj wyrażenia
$ expr 3 * 2 ♯ Mnożenie
$ expr 3 \>= 2 ♯ Sprawdź czy większe lub równe
$ expr 3 = 2 ♯ Sprawdź czy równe
$ expr ziutek : 'ziut.*' ♯ *arg1 : regexp* ile znaków pasujących w *arg1* z *regexp*
basename name [suffix] - wypisz bazową nazwę pliku ze ścieżki
$ basename /home/pawel/tekst.txt .txt ♯ Wynik to *tekst*
dirname name - wypisz katalog ze ścieżki
$ dirname /home/pawel/tekst.txt ♯ Wynik to */home/pawel*
$ dirname tekst.txt ♯ Wynik to .
test wyrażenie - oblicz wyrażenie, forma alternatywna to [wyrażenie]
$ test -r plik ♯ Prawda, gdy plik istnieje i daje się czytać
$ [$VAL -gt 0] ♯ Prawda, gdy $VAL >= 0
$ [-z $STR] ♯ Prawda, gdy $STR na długość 0
$ [$STR = $NAP] ♯ Prawda, gdy $STR = $NAP
$! -d pl ♯ Prawda, gdy pl nie jest (negacja to !) katalogiem (-d)
$ wyr1 -a wyr2 ♯ Koniunkcja wyrażeń
$ wyr1 -o wyr2 ♯ Alternatywa wyrażeń

$\LaTeX$, DVI, postscript i inne

latex plik[.tex] - przetwórz dokument, tworzy em plik.dvi
dvips -o plik.ps plik[.dvi] - przetwórz dvi do ps (postscript)
gv plik[.ps] - obejrzyj plik postscriptowy
xdvi plik[.dvi] - obejrzyj plik dvi
lpr plik.ps - drukuj plik postriptowy
psnup - wiele stron na jednej
  `$ psnup -n 2 plik[.ps] > plik2.ps` ♯ dwie strony na jednej
psbook - zmień kolejność stron (książkowa)
  `$ psbook plik1.ps plik2.ps`
pstoimg - konwertuj postscript do bitmap (png)
  `$ pstoimg -antialias -aaliastext -transparent -multipage plik.ps`

Bash

polecenie 2>plik - przekierowanie stderr
$HOME/.bashrc - skrypt uruchamiany przy uruchomieniu bash-a
$HOME/.bash_profile - skrypt uruchamiany po zalogowaniu
$HOME/.bash_logout - skrypt uruchamiany przy wyłogowaniu
set - operacje na zmiennych
  `$ set` ♯ wypisz środowisko
  `$ set par1 par2` ♯ $1 będzie *par1*, $2 będzie *par2*
[export] VAR=val - przypisz wartość zmiennej
  `$ PATH=$PATH;.` ♯ dodaj . do $PATH
  `$ export HOME` ♯ wyeksportuj zmienną HOME
  `$ export NAME=`**"pawel"** ♯ przypisz i wyeksportuj
test ... - testowanie (patrz pomoc do Unixa i Linuxa)
nawiasy: { POLECENIA } - nie ma znaczenia
nawiasy: (POLECENIA) - polecenia będą wykonane w nowym interpreterze
  `$ (export HOME=/tmp; ls $HOME); ls $HOME` ♯ wynik: sprawdzić

Skrypty

./skrypt [argumenty] - uruchom skrypt (musi mieć prawo x); tworzony jest nowy bash
bash skrypt - uruchom nowy bash, który wykona skrypt
source skrypt - wykonaj skrypt w bieżącym bashu; skrót: . *skrypt*
$0 - nazwa skryptu
$1,$2... - kolejne parametry skryptu
$# - liczba parametrów
$* - wszystkie parametry, tzn. $# napisów
$@ - wszystkie parametry, jeden napis
$$ - identyfikator bieżącego procesu
shift [n] - przesuń parametery o n pozycji (domyślnie 1)
  `$ shift` ♯ $2 staje się $1, $3 -> $2, etc...
  `$ shift 4` ♯ $5 staje się $1, $6 -> $2, etc...

Bash - instrukcje

- średnik w instrukcjach można zastąpić nową linią
- COM - polecenia
- warunek COM jest prawdziwy gdy kod zakończenia COM jest 0

if COM; then COM1; [else COM2;] fi - instrukcja warunkowa
while COM; do COM1; done - pętla dopóki
for NAME [in WORDS...] ; do COM; done - pętla for
  `$ for i in *; do echo $i; done` ♯ wylistuj pliku z bieżącego katalogu
  `$ for i ; do echo $i; done` ♯ wylistuj argumenty (iteracje dla $1,$2,...)
  `$ for i ; in ‘seq 5‘ do echo $i; done` ♯ wypisz liczby od 1 do 5
case ... - instrukcja case => *help case*
COM | ...| COM - potok poleceń z łączonymi strumieniami stdout => stdin
  `$ grep Ziuta *.txt | less` ♯ sprawdzić

COM1 || COM2 - wykonanie warunkowe, wykonaj COM2 gdy COM1 zakończyło się porażką
COM1 && COM2 - wykonanie warunkowe, wykonaj COM2 gdy COM1 zakończyło się sukcesem
`$ latex mgr && dvips -o mgr.ps mgr && gv mgr` ♯ sprawdzić
COM1 ; COM2 - wykonanie bezwarunkowe, wykonaj zawsze COM2 i COM1
COM & - wykonanie asynchroniczne (w tle)
`$ xemacs &`
function NAME { COM } - definiuj funkcje, argumenty to 1,2,...
`$ function MYPRINT { echo $*; }` ♯ prosta funkcja - drukuj argumenty
`$ MYPRINT arg1 arg2` ♯ wołaj funkcje MYPRINT

Bash - cudzysłowy, etc.

cudzysłów "..." - zmienne rozwijane, znaki *,? nie
`$ echo "$HOME *.txt"` ♯ wynik: */home/gorecki *.txt*
apostrof '...' - kasuje znaczenie specjalne znaków
`$ echo '$HOME *.txt'` ♯ wynik: *$HOME *.txt*
odwrotny apostrof \`...\` - wykonuje polecenie i wstawia w jego miejsce dane ze stdout
``$ grep "`date`" *`` ♯ wzorcem dla em grep będzie wynik wykonania *date*
`$ grep "$(date)" *` ♯ j/w, $(...) forma preferowana dla \`...\`

P.Górecki, gorecki@mimuw.edu.pl, 18.12.2003