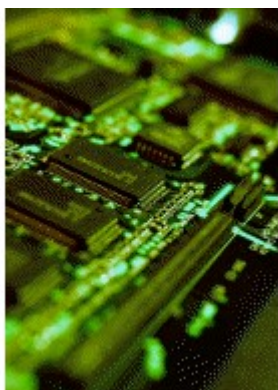




Systemy Operacyjne i Sieci Komputerowe



Sprzęt
komputerowy



System Operacyjny
+
Programy



Łatwe
użytkowanie

Prowadzący: Robert Szmurło
szmurlor@iem.pw.edu.pl
GE 229



Pomoc w Unixie

- Pomoc:
 - Dla posiadaczy internetu: www.google.com
 - Podręcznik złożony z rozdziałów: `man [słowo]` (np.: `man man`)
 - Wyszukiwanie haseł: `apropos [hasło]`
 - System `info`
 - Dokumentacje aplikacji: `/usr/share/doc` lub `/usr/local/share/doc`
- Wyszukiwanie plików i w plikach:
 - `find .` lub `find . -name "nazwa"`
 - `find . | grep "nazwa"`
 - `locate` (oparte na systemowej bazie danych `updatedb`, która jest uruchamiana zazwyczaj raz dziennie)
 - `grep [wyrażenie] [gdzie]`
 - `whereis [komenda]`



Monitorowanie stanu systemu

- Obciążenie procesów: `top`
- Liczba zajętego obszaru na dyskach: `df`
- Czas od uruchomienia systemu: `uptime`
- Wyświetlenie wersji systemu: `uname -a`
- Sprawdzenie ile mamy dozwolonego miejsca na dyskach: `quota -v`



Konfiguracja środowiska pracy użytkownika

- Powłoka: **sh, zsh, bash**
- Zmienne środowiskowe:
 - **env**
 - **export EDITOR=pico**
 - **echo \$zmienna**
- Pliki konfiguracyjne:
 - **/etc/bashrc**
 - **~/.bash_profile**
 - **~/.bashrc**
- Aliasy:
 - **alias ll='ls -la'**
- Wyświetlanie plików:
 - **cat** (cat /etc/fstab)
 - **more** (ls -l | more)
 - **less** (less /etc/dhcpd.conf)
- Zmiana znaku zachęty:
 - **export prompt="%n%\: %m% #"**
 - Przykładowe znaki:
 - **%n** nazwa użytkownika,
 - **%m** krótka nazwa hosta ,
 - **%M** pełna nazwa hosta (np.: apple.cs.byu.edu)
 - **%T** aktualny czas w formacie 24h
- Katalog domowy:
 - **/home/ziutek**
 - Jak szybko przejść do katalogu domowego nie pamiętając dokładnej lokalizacji?
 - **cd \$HOME**
 - **cd ~**
 - **cd**

Czyli na co ma wpływ zwykły użytkownik.



Przetwarzanie Potokowe

- **STDIN** - Standardowe wejście,
STDOUT - standardowe wyjście,
STDERR - standardowe wyjście błędów.

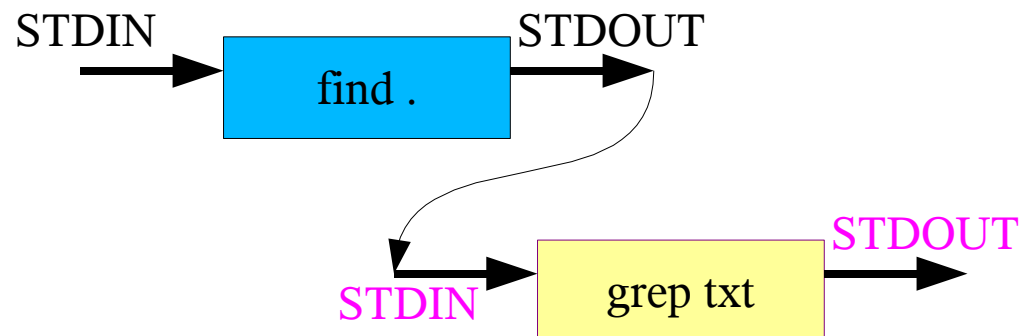
- Programowanie strumieni:

- przekierowanie do i z plików:

- `ls -la > do_pliku.txt`
- `ls -la >> do_pliku_dopisujac.txt`
- `grep ala < z_pliku.txt`
- **CTRL+C, CTRL+D**

- przkierowanie jednego wyjścia na wejście drugiego:

```
find . | grep txt
```



Przetwarzanie potokowe pomaga analizować większe zbiory danych oraz automatyzować niektóre zadania. Wynik jednego programu jest przekazywany jako dane wejściowe do następnego, itd...



Przykłady Przetwarzania Potokowego

- `ps -aux | grep ziutek`
- `ls -la | more`
- `find . | grep txt`
- `make 2>&1 | tee PLIK_LOG`
- Z ilu różnych hostów zostały zarejestrowane odpytania naszego serwera WWW:
`tail -10000 /var/log/www | cut -d ' ' -f 1 | sort | uniq | wc`
- Wyświetlanie fragmentów plików:
 - od początku: `head -20 /var/log/messages`
 - od końca: `tail -20 /var/log/messages`
- Sortowanie wyników zwracanych przez program:
 - `sort`



Dalsze Podstawy Linii Komend

- Pliki ukryte rozpoczynają się od kropki:
 - `.nazwa`
- Zmienna systemowa PATH definiuje liste katalogów, które są przeszukiwane w celu znalezienia pliku wykonywalnego.
 - `export PATH=$PATH:`
 - `export PATH=/opt/fx/bin:$PATH`
- Znaki specjalne:
 - `.` - bieżący katalog,
 - `|` - strumień
 - `$` - zmienna
 - `'`, `"` - zmienne tekstowe
 - ``` - podstawienie komendy
- Kontrola zadań:
 - zawieszenie zadania: **CTRL+Z**
 - przerwanie zadania: **CTRL+C**
 - przywrócenie na wierzch: **fg**
 - przywrócenie w tło: **bg**
- Uruchamianie procesów w tle:
 - Na końcu komendy dodać znak **&**.



Archiwizacja Danych

- Kompresja pojedynczego pliku:
 - `gzip [nazwa.txt]` (w wyniku powstanie nazwa.txt.gz)
 - `bzip2 [nazwa.txt]` (w wyniku powstanie nazwa.txt.bz2)
- Dekompresja plików:
 - `gzip -d *` lub `gzip -d [nazwa.txt.gz]`
 - `bzip2 -d nazwa.txt.bz2`
- Scalanie drzew folderów i plików w jeden plik:
 - `tar -cvf [n_archiwum.tar] [katalog]`
- Przywracanie drzewa z pliku:
 - `tar -xvf [n_archiwum.tar] [katalog docelowy]`
- Scalanie i przywracanie z kompresją:
 - `tar -cvzf [n_archiwum.tar] [katalog docelowy]` lub `tar -cvjf [n_archiwum.tar] [katalog docelowy]`
 - `tar -xvzf [n_archiwum.tar]`



Edytory Unix

- Wszędzie jest edytor **vi**.
- Nie wszędzie są: **pico**, **nano**, **mcedit**, **joe**.
- Podstawowe informacje o **vi**:

- **vi** umożliwia pracę w dwóch trybach: edycji i komend. Standardowo po uruchomieniu edytora pracujemy w trybie komend.
- **a** - przejście do trybu edycji (rozpoczęcie dodawania nowego tekstu za znakiem aktualnie zasłoniętym przez kursor).
- **i** - przejście do trybu edycji (rozpoczęcie dodawania nowego tekstu przed znakiem aktualnie zasłoniętym przez kursor).
- **r** - przejście do trybu edycji tylko na jeden znak (Zastąpienie znaku aktualnie zasłoniętego przez kursor).
- **R** - przejście do trybu edycji (rozpoczęcie zastępowania wszystkich znaków).
- **ESC** - przejście do trybu komend.
- **:** - przejście do trybu komend wpisywanych na ekranie.

i – rozpocznijw stawianie
przed aktualną pozycją

a – rozpocznijw stawianie
za aktualną pozycją

Pisanie w vi jest proste.

Pozycja
kursora



Edytory Unix: vi

- Przykład
 - Utworzenie nowego dokumentu wpisanie jednego zdania, zapisanie zmian i wyjście:
 - `max# vi nowy.txt`
 - Rozpoczęcie dodawania tekstu:
 - `[Klawisz i]Pisanie w vi jest proste.[ESC][:][w][q]`
 - Objasnienie:
 - `[Klawisz i]` - rozpoczęcie wstawiania
 - `'Pisanie w vi jest proste.'` - wpisane zdanie
 - `[ESC]` - powrót do trybu komend
 - `[:]` - przejście do trybu komend wpisywanych na ekranie
 - `[w]` - zapisanie zmian
 - `[q]` - wyjście z vi
 -



Edytory Unix: vi

- Jak wyjść z vi?
 - **[ESC]:q!** - wyjście z vi bez zapisywania zmian
 - **[ESC]:wq** - wyjście z vi z zapisaniem zmian (dokument musi mieć przyporządkowaną nazwę)
- Jak zapisać dokument pod inną nazwą?
 - **[ESC]:w nowa_nazwa.txt** - zapisanie dokumentu pod nową nazwą
- Jak skopiować i wkleić fragment dokumentu?
 - **[ESC]yy** - skopiowanie linii w której aktualnie znajduje się kursor
 - **[ESC]dd** - wycięcie linii w której aktualnie znajduje się kursor
 - **[ESC]p** - wklejenie skopiowanego tekstu za kursorem
 - **[ESC]P** - wklejenie skopiowanego tekstu przed kursorem

Kopiowanie i wklejanie całych linii tekstu

yy – skopiowanie linii w której aktualnie znajduje się kursor

p – w trybie całych linii wklejanie powoduje wstawienie skopiowanej linii za aktualną

Ala ma szarego kota.
Franek ma ładnego psa.

Ala ma szarego kota.
Ala ma szarego kota.
Franek ma ładnego psa.



Edytory Unix: vi

- Więcej informacji: <http://www.iem.pw.edu.pl/wikidyd/LOP/VI>



Automatyzacja Zadań

- Pętle w powłocie:
 - `for i in [zbior]; do [komenda]; done`
- Warunki:
 - `if [$a -eq "iles"]; then [komenda]; fi`
- Przykład:
 - Chcemy pobrać z konkretnego adresu internetowego 100 plików o nazwach różniących się liczbą pod konkretnym numerem:
 - `for i in `seq 1 100`; do wget http://www.adres/com/plik$i.jpg; done`



Automatyzacja Zadań: Skrypty

- Skrypt w unix:
 - musi mieć uprawnienia do wykonywania (chmod u+x skrypt.sh)
 - pierwsza linijka skryptu musi być postaci:
`#!/ściezka_do_powloki`
(np: `#!/bin/bash`)
 - każda linijka skryptu to po prostu komenda
 -

```
#!/bin/sh
if [ "$LANG" = "pl_PL" ]
then
    echo Czesc
else
    echo Hello
fi
```

```
#!/bin/bash
for i in $( ls ); do
    echo item: $i
done
```

```
#!/bin/bash
COUNTER=0
while [ $COUNTER -lt 10 ]; do
    echo The counter is $COUNTER
    let COUNTER=COUNTER+1
done
```

```
#!/bin/bash
echo Please, enter your name
read NAME
echo "Hi $NAME!"
```



Automatyzacja Zadań: cron

- Zaplanowane zadania:
 - **cron**
 - **/etc/crontab** : ustawienia systemowe
 - **crontab -e** : zaplanowane zadania jednego użytkownika.
- minuta, godzina, dzień, miesiąc, dzień tygodnia, komenda
- Liczba oznacza wartość o której ma być uruchamiana komenda.
- Konstrukcja: ***/5** oznacza: uruchamiaj co 5...

```
# Rotate log files every hour, if necessary.
0 * * * * root newsyslog
#
# Perform daily/weekly/monthly maintenance.
1 5 * * * root periodic daily
16 5 * * 6 root periodic weekly
31 5 1 * * root periodic monthly
#
# Adjust the time zone if the CMOS clock keeps local time, as opposed to
# UTC time. See adjkerntz(8) for details.
1,31 0-5 * * * root adjkerntz -a
# AT
*/5 * * * * root LC_ALL= /usr/local/bin/mrtg /usr/local/etc/mrtg.cfg
```



Automatyzacja zadań: Perl

- Pierwsza linijka: `#!/usr/bin/perl`
- Zmienne: `$` - skalarne, `@` - tablice, `%` - mapy (hashe, słowniki)

```
#!/usr/bin/perl
$login = $ARGV[0];
open PASSWD, '/etc/passwd';
while(<PASSWD>){
    @t = split ':';
    $IN{$t[0]} = $t[4];
}
print "$login to ".$IN{$login}."\n";
```

```
#!/usr/bin/perl
while(<STDIN>){
    @t = split ':';
    $IN{$t[0]} = $t[4];
}
@logins = keys %IN;
foreach $login (@logins) {
    print "$login to $IN{$login}\n"
}
```

```
#!/usr/bin/perl
$imie = $ARGV[0];
$i = 0;
while(<STDIN>){
    if ( $_ =~ /$imie/ ) {
        $i = $i + 1;
    }
}
print "W standardowym wejściu znalazłem $i linii, w których wystąpiło imię $imie.\n";
```




Interakcja

- Jeżeli coś cię zainteresowało i chciałbyś aby na następnym wykładzie zostało rozszerzone, powtórzone, omówione dokładniej, to nie krępuj się i napisz maila:

szmurlor@iem.pw.edu.pl

- Jeżeli coś było nie jasne, napisz maila:

szmurlor@iem.pw.edu.pl

- Jeżeli coś cię znudziło, napisz maila:

szmurlor@iem.pw.edu.pl