

## BAB III Perkenalan LINUX System

### Philosophy of UNIX and GNU/LINUX

- Multiuser, mampu menangani banyak user secara simultan baik pada aplikasi yg sama atau pun berbeda
  - Setiap User harus memiliki account untuk memasuki system
  - Setiap User harus login untuk memasuki system
- Multitasking, mampu menangani sejumlah pekerjaan (task) secara simultan
- Kernel Linux merupakan kernel modular yg dapat dikonfigurasi dengan modul modul (berupa device driver) pluggable yg dapat dipasang atau dilepas sesuai kebutuhan. Hal ini merupakan salah satu kelebihan UNIX/Linux dibanding system lain.
- Semua device/hardware dianggap sebagai file

### Command Line Interface

- Sama seperti system operasi Windows, Linux juga memiliki GUI buat interfaces (KDE dan GNOME) tetapi pada pelatihan ini kita tidak banyak membicarakan ttg GUI tadi. Khan dah mudah seperti windowze.... :)
- Shell merupakan suatu program yang menghubungkan si pengguna/kita dengan *Kernel Linux*. Shell menggunakan *command line interfaces*. Ada banyak jenis shell tapi yg umum digunakan dan yg akan kita gunakan adalah shell *bash* ( Bourne again shell ).
- Dengan shell kita dapat memaintenance semua pekerjaan terhadap file atau direktori seperti bekerja dgn interfaces di windows/Xwindows  
Default bash prompt :

**[you@slashsmart etc]\$\_**

dimana **you** adalah nama login kamu, **slashsmart** adalah nama komputer kamu, dan **etc** merupakan direktori tempat kamu berada saat ini. Perlu diperhatikan dalam modul ini tanda **underscore** " \_ " menunjukkan kursor dimana menunggu kamu mengetikkan command.

Setelah mengetikkan command/perintah dan menekan tombol enter. Dengan segera

kita memperoleh hasil dari perintah yg kita berikan tadi.

Misalnya :

```
[you@slashsmart etc]$ date
Sat Nov 16 19:18:48 WIT 2002
[you@slashsmart etc]$_
```

### Log out atau keluar dari system

- Untuk keluar dari system ketik :

```
[you@slashsmart etc]$ exit
atau
[you@slashsmart etc]$ logout
```

- Alternative lain dengan menekan tombol Ctrl dan D. Sering dituliskan Ctrl+D atau ^d

### Command Syntax

- Banyak perintah/command yang membutuhkan beberapa argument mis, command **echo** untuk menampilkan argumen yang kita berikan :

```
[you@slashsmart etc]$ echo Selamat Berlatih
Selamat Berlatih
[you@slashsmart etc]$_
```

- Unix/Linux menggunakan case-sensitive syntax, termasuk pada command/perintah yang digunakan. Mis :

```
[you@slashsmart etc]$ ECHO Good Luck
-bash: ECHO: command not found
[you@slashsmart etc]$_
```

- Banyak command/perintah yang memiliki **options** yang biasanya diawali dengan '-' atau '--'

Misal:

```
[you@slashsmart etc]$ echo -n Selamat Berlatih
Selamat Berlatih[you@slashsmart etc]$_
```

### Files dan Direktori

- Data dapat disimpan dalam file dan beberapa direktori berisi banyak file. Nama ekstensi dari file seperti **.txt** pada file **contoh.txt** tidak merupakan keharusan pada system Linux seperti pada windowze.
- Setiap file dan direktori memiliki nama.
- Untuk menampilkan file dan direktori digunakan perintah/command **ls**. Misal di direktori home saya menampilkan :

```
[josh@slashsmart josh]$ ls
data/          josh.kwd      Linux Kernel.pdf*  profile.zip*      tmp/
Desktop/      josh.kwd~    Mail/              public_html/
[josh@slashsmart josh]$_
```

### Creating File dan Editor

- Untuk membuat suatu file kita dapat menggunakan editor. Di linux banyak editor yg bisa kita gunakan misal **pico,vi,emacs,mcedit** dan banyak lagi yang bisa dipelajari dengan mudah. Penulis lebih senang menggunakan **vi** karena lebih simple tetapi sangat powerfull :)
- Untuk membuat file berisi data sederhana misalnya **nama\_hari** , kita dapat menggunakan perintah **cat**.

```
[you@slashsmart you]$ cat > nama_hari
senin
selasa
rabu
kamis
^d
[you@slashsmart you]$_
```

- Perhatikan tanda lebih besar ( > ), ini sangat sering digunakan dalam pembuatan file
- Setelah memberikan data dalam hal ini nama nama hari , tekan Ctrl+D untuk mengakhiri
- Gunakan **ls** untuk menampilkan file tersebut.

### Menampilkan isi file

- Dalam UNIX/Linux banyak cara untuk menampilkan isi suatu file. Misalnya dengan editor
- Command **cat** sering digunakan untuk menampilkan isi file. Yakni dengan memberi argument nama file tersebut.:

```
[you@slashsmart you]$ cat nama_hari
```

```
senin
```

```
selasa
```

```
rabu
```

```
kamis
```

```
[you@slashsmart you]$_
```

- Perlu di perhatikan bahwa menampilkan isi file dengan command **cat** disarankan hanya untuk file text saja. Jika file yg ingin kamu tampilkan adalah binary maka hasilnya akan sangat amburadul.. :). Untuk menentukan jenis suatu file kita gunakan command **file**.

```
[you@slashsmart you]$ file nama_hari
```

```
nama_hari: ASCII text
```

```
[you@slashsmart you]$_
```

- Jika file text yang ingin kita lihat terlalu panjang kita dapat menggunakan command **more** atau **less** yang akan di bahas di bab selanjutnya

### Menambah isi file ( appending to a file )

- Dari shell, untuk menambah isi file menggunakan tanda lebih besar sebanyak 2 kali ( >> ), misalnya :

```
[you@slashsmart you]$ cat >> nama_hari
```

```
jumat
```

```
sabtu
```

```
minggu
```

```
[you@slashsmart you]$ cat nama_hari
```

```
senin
```

```
selasa
```

```
rabu
```

```
kamis
```

```

jumat
sabtu
minggu
[you@slashsmart you]$_

```

### Menghapus File dengan rm

- Untuk menghapus file gunakan command **rm**
- Caranya mudah, hanya dengan memberi nama file yang ingin dihapus sebagai argument  

```
[you@slashsmart you]$ rm nama_hari
```

```
[you@slashsmart you]$_
```
- Maka file nama\_hari akan dihapus. Perlu diingat bahwa di Unix/Linux tidak ada recycle bin ataupun perintah 'unrm' atau undelete :-)

### Changing Password / Mengganti Password

- Password sebaiknya :
  - kamu nggak pernah lupain
  - teman kamu nggak bisa tebak
  - Minimal terdiri dari sebuah character yang bukan huruf atau angka
  - Diubah secara periodic
- Dengan passwd command :

```
[you@slashsmart you]$ passwd
```

Changing password for user you.  
Changing password for you  
(current) UNIX password: <Masukkan password kamu yang lama>  
New UNIX password:<Masukkan password kamu yang baru>  
Retype new UNIX password:<Masukkan password kamu baru sekali lagi>  
passwd: all authentication tokens updated successfully.

```
[you@slashsmart you]$_
```

### Copying and Renaming Files/ mengcopy dan mengganti nama file

- Untuk mengcopy gunakan command **cp**  
`[you@slashsmart you]$ cp linux.pdf kernel.pdf`  
`[you@slashsmart you]$_`
- Untuk mengganti nama file gunakan command **mv**  
`[you@slashsmart you]$ mv linux.pdf secure.pdf`  
`[you@slashsmart you]$_`
- Jika nama file yg ingin dibuat sudah ada, maka file tersebut akan di *overwritten* atau di tukar

### Filename Completion

- Shell dapat membuat proses pengetikan command maupun nama file menjadi mudah
- Dengan mengetikkan beberapa char/huruf yang tidak ambigu lalu mengetikkan tombol **TAB** maka command/nama file yang ingin kita ketikkan akan muncul.
- Misalnya :  
`[you@slashsmart you]$ rm na`  
lalu menekan tombol **TAB**  
`[you@slashsmart you]$ rm nama_hari`

### Command History

- Disamping Filename/Command Completion pada shell terdapat juga kemudahan yakni adanya command history. Dimana perintah perintah yang sudah kita ketikkan sebelumnya dapat kita pakai kembali dengan menekan **UP** dan **DOWN**
- Kemudian untuk mengedit perintah tersebut dapat digunakan Backspace atau delete
- Untuk berpindah ke awal command digunakan **CTRL+a**

## Files dan Direktori

- Seperti di pada bab sebelumnya di katakan bahwa direktori berisikan file file dan direktori lainnya
- Kita mengenal direktori **root (/)** merupakan top level dari direktori pada sistem LINUX
- Direktori lainnya juga selalu menggunakan nama dengan tanda / di depannya

## Absolute Paths

- Sebuah Path absolute di awali dari direktori root ( / ), dan nama direktori yang akan di masuki berada di dalamnya
- Misalnya perintah ls yang sebenarnya berada pada direktori /bin/ls
- Jadi untuk menjalankan perintah ls dengan menggunakan absolute path yakni dengan cara

```
[josh@slashsmart josh]$ /bin/ls
```

- Kita juga dapat menggunakan ls untuk menampilkan file file pada sebuah direktori dengan absolute path

```
[josh@slashsmart josh]$ ls /usr/share/doc/
```

## Current Direktori

- Untuk mengetahui direktori dimana kira berada saat ini, gunakan perintah **pwd** :

```
[josh@slashsmart josh]$ pwd  
/home/josh
```

- Untuk berpindah direktori kita gunakan command **cd** :

```
[josh@slashsmart josh]$ cd /var/log/  
[josh@slashsmart log]$ pwd  
/var/log
```

- Untuk kembali ke direktori home kita, gunakan command cd tanpa diikuti option atau path.

## Membuat dan Menghapus Direktori

- Untuk membuat suatu direktori digunakan **mkdir** diikuti dgn nama direktori yang ingin dibuat :

```
[josh@slashsmart josh]$ mkdir daftar
```

- Perlu di ingat pada sistem unix/linux pada umumnya kita diperbolehkan membuat/menghapus direktori ataupun menuliskannya hanya pada home direktori kita.
- Untuk menghapus digunakan **rmdir**. Perintah ini hanya jika direktori yg akan dihapus adalah direktory kosong (tidak ada file ataupun direktori lainnya di dalamnya)

```
[josh@slashsmart josh]$ rmdir daftar
```

- Untuk menghapus direktori yang memiliki file atau direktori didalamnya gunakan **rm** diikuti dengan options **-r** :
- ```
[josh@slashsmart josh]$ rm -r daftar
```
- Hati hati menggunakan perintah ini :) karena pada system unix/linux tidak ada recycle bin ataupun perintah undelete

### Relative Paths

- Untuk memasuki suatu direktori kita tidak harus memulai dari direktori root ( / ) atau dengan katalain kita menggunakan relative path. Jadi tergantung dari direktori dimana kita berada saat itu.
- Misalnya :

```
[josh@slashsmart josh]$ cd /
```

```
[josh@slashsmart /]$ cd usr
```

```
[josh@slashsmart usr]$ cd share
```

```
[josh@slashsmart share]$
```

### Special Dot Directories

- Ada dua direktori spesial menggunakan dot pada system unix/linux

- Pertama direktori .. yang menunjukkan direktori parent atau direktori diatas kita.

```
[josh@slashsmart josh]$ cd ..
[josh@slashsmart home]$ pwd
/home
[josh@slashsmart home]$ cd ..
[josh@slashsmart /]$ pwd
/
```

- Kedua direktori . yang menunjukkan direktori tempat kita berada.

```
[josh@slashsmart josh]$ cd ./Mail
sama saja dengan
[josh@slashsmart josh]$ cd Mail
```

### Hidden Files and Directories

- Direktori spesial . dan .. merupakan hidden direktori, untuk melihatnya gunakan **ls** dengan option **-a**.

```
[josh@slashsmart josh]$ ls -a
./                .bash_logout
../               .bash_profile
.bash_history     .bashrc
public_html/
```

- Sebagai pertanda, semua file yang terhidden diawali dengan .
- Biasanya file file hidden merupakan file file konfigurasi.

### Path to Home Directories

- Symbol yang digunakan yakni ~ .Untuk memasuki direktori yang berada pada home kita dapat menggunakan :

```
[josh@slashsmart etc]$ cd /home/josh/public_html
sama saja dengan :
[josh@slashsmart etc]$ cd ~/public_html
```

- Untuk memasuki home user lain (misal pada system saya ada user lain dengan nama lognight)

kita menggunakan tanda `~` di depan nama user :

```
[josh@slashsmart josh]$ cd ~lognigh/
```

### Mencari Files dalam system

- Perintah yang digunakan untuk menampilkan file file yang memiliki nama tertentu kita gunakan perintah **locate**

```
[josh@slashsmart josh]$ locate mkdir
/usr/share/man/man1/mkdir.1.bz2
/usr/share/man/man2/mkdir.2.bz2
/bin/mkdir
...
```

### Specifying Multiple Files

- Pada shell, kita dapat bekerja dengan beberapa file sekaligus. Misalnya membuat beberapa file atau pun menghapusnya secara bersamaan.

Misal :

```
[josh@slashsmart josh]$ rm test.txt fileku index.html
[josh@slashsmart josh]$ mkdir dir1 dir2 dir3
[josh@slashsmart josh]$ cat test.txt fileku index.html
```

### Menampilkan Documentasi program/command pada system UNIX/Linux

- Pada system Unix/Linux, semua command command/program yang ada memiliki dokumentasi lengkap yang dengan mudah dapat kita tampilkan. Sehingga penggunaan setiap program dapat di ingat kembali ataupun di pelajari secara manual. Manual yang disediakan tersebut disebut dengan **manual page**
- Untuk menampilkan manual page dari suatu program kita gunakan command **man** yang diikuti dengan nama program yang ingin kita tampilkan manual pagenya.

```
[josh@slashsmart josh]$ man mkdir
```

- Untuk keluar dari manual page kita gunakan tombol **q**

### Menggunakan Wildcards dalam menampilkan file

- Untuk menampilkan file file yang sesuai dengan keinginan kita dilakukan dengan menggunakan wildcards misalnya :

```
[josh@slashsmart josh]$ ls -l *.txt
-rw-rw-r-- 1 josh  jsh      2834 Nov 27 13:08 pelatihan.txt
-rw-rw-r-- 1 josh  jsh      7415 Nov 28 07:43 tes.txt
```

- Wildcards juga dapat digunakan untuk mempermudah dalam administrasi file. Misal untuk menghapus semua file pada suatu direktori dapat digunakan :

```
[josh@slashsmart josh]$ rm *
```

### Menjalankan program secara bersamaan

- Pada system Linux kita dapat menjalankan command secara bersamaan misalnya untuk melihat/menghitung user yang sedang login kita gunakan command **who** dan **wc** :

```
[josh@slashsmart josh]$ who | wc -l
3
```

- Untuk melakukan pemanggilan lebih dari satu perintah dalam satu baris, dapat kita gunakan tanda semicolon ";" sebagai pemisah antar perintah.

```
[you@slashsmart you]$ date ; pwd ; echo Met datang di Pelatihan LINUX
Fri Dec 13 05:23:44 WIT 2002
/home/you
Met datang di Pelatihan LINUX
[you@slashsmart you]$
```

### Melihat Pemakai Aktif Saat Ini

- Perintah **who** dapat digunakan untuk menampilkan daftar pemakai yang sedang aktif dalam sistem LINUX/UNIX.

```
[you@slashsmart you]$ who
josh  vc/1      Dec 13 04:48
josh  pts/0      Dec 13 04:49
josh  pts/1      Dec 13 04:50
josh  pts/2      Dec 13 05:22
you   pts/3      Dec 13 05:28 (localhost)
[you@slashsmart you]$
```

- Atau dapat juga dengan command ' w ' saja :

```
[you@slashsmart you]$ w
5:30am up 42 min, 5 users, load average: 0.47, 0.48, 0.44
USER  TTY  FROM          LOGIN@  IDLE  JCPU  PCPU  WHAT
josh  vc/1  -             4:48am 41:22  0.31s 0.04s /bin/sh /usr/X1
josh  pts/0  -             4:49am 4037   0.00s ?     -
josh  pts/1  -             4:50am 16:46  0.40s 0.40s /bin/bash
josh  pts/2  -             5:22am 0.00s 0.53s 0.16s /bin/bash
you   pts/3  localhost    5:28am 0.00s 023s  0.05s w
[you@slashsmart you]$
```

### Mengambil karakter dan juga field dengan command cut

- Secara umum perintah **cut** akan mengambil karakter tertentu dari setiap baris data yang diberikan, sesuai dengan pilihan (option) yang diberikan. Format umumnya adalah:

#### **cut -cX file**

dimana **X** adalah jangkauan dari karakter yang akan kita ambil dari setiap baris yang diberikan.

- Jangkauan memiliki ekspresi tersendiri seperti pada contoh berikut:

```
[you@slashsmart you]$ cat coba
```

Selamat datang di pelatihan LINUX

abcdefghijklmnopqrstuvwxy

1234567890987654321

**[you@slashsmart you]\$** cut -c5 coba

m

e

5

**[you@slashsmart you]\$**

- Untuk mengambil karakter ke-5 dan seterusnya gunakan :

**[you@slashsmart you]\$** cut -c5- coba

mat datang di pelatihan LINUX

efghijklmnopqrstuvwxy

567890987654321

**[you@slashsmart you]\$**

- Untuk mengambil karakter ke-2 dan ke-7 gunakan :

**[you@slashsmart you]\$** cat coba | cut -c2,7

et

bg

27

**[you@slashsmart you]\$**

- Untuk mengambil karakter ke-2 sampai ke-7 gunakan :

**[you@slashsmart you]\$** cat coba | cut -c2-7

alamat

bcdefg

234567

**[you@slashsmart you]\$**

- Disamping bekerja berdasarkan karakter, perintah cut dapat bekerja berdasarkan field, yaitu menggunakan pilihan -f.
- Pemisah field standar adalah blank. Untuk mengubah pemisah field standar dapat

kita gunakan pilihan -d.

```
[you@slashsmart you]$ cat /etc/passwd  
root:x:0:0:root:/root:/bin/bash  
bin:x:1:1:bin:/bin:/bin/sh  
daemon:x:2:2:daemon:/sbin:/bin/sh  
adm:x:3:4:adm:/var/adm:/bin/sh  
lp:x:4:7:lp:/var/spool/lpd:/bin/sh  
sync:x:5:0:sync:/sbin:/bin/sync  
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
```

- Untuk menampilkan field pertama yang dipisahkan oleh tanda ':' gunakan :

```
[you@slashsmart you]$ cut -d: -f1 /etc/passwd  
root  
bin  
daemon  
adm  
lp  
sync  
shutdown
```

- Untuk menampilkan field pertama dan keenam :

```
[you@slashsmart you]$ cut -d: -f1,6 /etc/passwd  
root:/root  
bin:/bin  
daemon:/sbin  
adm:/var/adm  
lp:/var/spool/lpd  
sync:/sbin  
shutdown:/sbin
```

**Mengurutkan data pada file dengan perintah sort**

- Untuk mengurutkan data pada file kita gunakan perintah `sort`, misalnya:

```
[you@slashsmart you]$ cat mahasiswa
```

```
rudi
```

```
andi
```

```
siska
```

```
josh
```

```
feby
```

```
roy
```

```
[you@slashsmart you]$ sort mahasiswa
```

```
andi
```

```
feby
```

```
josh
```

```
roy
```

```
rudi
```

```
siska
```

```
[you@slashsmart you]$
```

- Secara default, perintah `sort` akan mengurutkan data dari kecil ke besar. Untuk melakukan hal yang sebaliknya, yaitu urut dari besar ke kecil digunakan pilihan `-r`.

```
[you@slashsmart you]$ sort -r mahasiswa
```

```
siska
```

```
rudi
```

```
roy
```

```
josh
```

```
feby
```

```
andi
```

## Membuat Program Shell Sederhana

- Pemrograman shell dapat dilakukan dengan menggunakan dua cara, yaitu mengetikkan program langsung ke terminal dan pengetikan program ke dalam file. File yang berisi program shell disebut script.
- Pada bab-bab sebelumnya, sesungguhnya kita telah melakukan pemrograman secara pengetikan program langsung ke terminal.
- Perhatikan program/command yang pernah kita lakukan pada bab terdahulu. (program sangat sederhana)

```
[you@slashsmart you]$ ls |wc -l
```

```
6
```

```
[you@slashsmart you]$
```

- Namun cara pemrograman seperti di atas sangat tidak praktis karena setiap kali kita akan menggunakannya, kita harus mengetik ulang perintah-perintah tersebut.
- Hal ini akan terasa lebih tidak efisien lagi jika perintah yang digunakan dalam program tersebut adalah lebih dari satu perintah. Untuk itu kita cenderung menggunakan cara yang kedua yaitu program diletakkan dalam file.
- Sebagai contoh, misalnya kita ingin membuat sebuah program shell yang bernama **useraktif** untuk mengetahui ada berapa pemakai aktif dalam sistem saat ini. Maka langkah pertama kita adalah menuliskan rangkaian perintah tersebut ke dalam file dengan menggunakan editor teks yang biasa anda gunakan. Perhatikan isi file yang berisi perintah untuk mengetahui jumlah pemakai aktif dalam sistem.

```
[you@slashsmart you]$ cat useraktif
```

```
who|wc -l
```

```
[you@slashsmart you]$
```

- Untuk menjalankan program yang baru kita buat adalah dengan memanggil nama programnya.

```
[you@slashsmart you]$ bash useraktif
```

```
5
```

```
[you@slashsmart you]$_
```

atau dengan :

```
[you@slashsmart you]$ sh useraktif
```

```
5
```

```
[you@slashsmart you]$_
```

- Dengan memanggil shell terlebih dahulu, program di atas dapat dijalankan dengan baik. Namun sesungguhnya ada cara lain yang lebih praktis sehingga kita tidak perlu selalu memanggil shell terlebih dahulu sebelum menjalankan program kita, yaitu dengan mengubah mode perijinan file program tersebut sehingga diijinkan untuk dieksekusi. Untuk itu digunakan perintah chmod :

```
[you@slashsmart you]$ ls -l useraktif
```

```
-rw-rw-r-- 1 you you 10 Dec 13 06:32 useraktif
```

```
[you@slashsmart you]$ chmod +x useraktif
```

```
[you@slashsmart you]$ ls -l useraktif
```

```
-rwxrwxr-x 1 you you 10 Dec 13 06:32 useraktif*
```

- Selanjutnya kita dapat menjalankan program secara langsung tanpa harus memanggil shell (sh atau bash) lagi. Yakni dengan cara:

```
[you@slashsmart you]$ ./useraktif
```

```
5
```

```
[you@slashsmart you]$
```

- Tanda ./ disertakan karena pada umumnya path untuk setiap user dalam system linux tidak mengikut sertakan path current direktori. Untuk mengubahnya dapat kita ketikkan :

```
[you@slashsmart you]$ PATH=$PATH:.
```

```
[you@slashsmart you]$ useraktif
```

```
5
```

```
[you@slashsmart you]$
```

- Sekarang coba kita membuat script shell yang menampilkan informasi berikut:
  - Waktu system
  - Info tentang anda
  - jumlah pemakai yang sedang login di system
 contoh scriptnya:

```
[you@slashsmart you]$ cat myinfo
```

```
#!/bin/bash
```

```
#filename myinfo
```

```
#membersihkan tampilan layar
```

```
clear
```

```
#menampilkan informasi
```

```
echo -n "Waktu system saat ini      :" ; date
```

```
echo -n "Username anda              :" ; whoami
```

```
echo -n "Jumlah pemakai saat ini    :" ; who | wc -l
```

```
[you@slashsmart you]$
```

- Sebelum dijalankan jangan lupa untuk merubah permission file myinfo sehingga dapat dieksekusi oleh anda

```
[you@slashsmart you]$ ./myinfo
```

```
Waktu system saat ini :Fri Dec 13 06:56:36 WIT 2002
```

```
Username anda      you
```

```
Jumlah pemakai saat ini : 5
```

```
[you@slashsmart you]$
```

## Environment Variable

- Merupakan variabel lingkungan yang digunakan khusus oleh shell atau system linux

kita untuk proses kerja system seperti variabel PS1, PS2, HOME, PATH, USER, SHELL, dsb...

- Jika digunakan akan berdampak pada system, misalkan variabel PS1 yang digunakan untuk mengeset prompt shell pertama yaitu prompt tempat anda menetikkan perintah - perintah shell (defaultnya "\s-\v\\$"), PS2 untuk prompt pelengkap perintah.
- Prompt ini akan ditampilkan jika perintah yang dimasukkan dianggap belum lengkap oleh shell (defaultnya ">"). anda dapat mengeset PS1 dan PS2 seperti berikut.
- Simpan dahulu isi PS1 asli system anda, sehingga nanti dapat dengan mudah dikembalikan

```
[you@slashsmart you]$PS1OLD=$PS1
```

sekarang masukkan string yang diinginkan pada variabel PS1

```
[you@slashsmart you]$PS1="LINUX==OK>>"
```

```
LINUX==OK>>ls
```

```
coba mahasiswa myinfo~ secure.pdf* useraktif*
linux.pdf* myinfo nama_hai tmp/
```

```
LINUX==OK>>_
```

maka prompt pertama akan berubah, untuk mengembalikan PS1 anda ke prompt semula ketikkan perintah

```
LINUX==OK>>PS1=$PS1OLD
```

```
[you@slashsmart you]$ls
```

```
coba mahasiswa myinfo~ secure.pdf* useraktif*
linux.pdf* myinfo nama_hai tmp/
```

```
[you@slashsmart you]$_
```

- Jika anda ingin mengkonfigurasi prompt shell, bash telah menyediakan beberapa backslash karakter diantaranya adalah:

|     |                                                                 |
|-----|-----------------------------------------------------------------|
| \a  | ASCII bell character (07)                                       |
| \d  | date dengan format "Weekday Month Date" (misalnya "Tue May 26") |
| \e  | ASCII escape character (033)                                    |
| \H  | hostname (namahost)                                             |
| \n  | newline (karakter baru)                                         |
| \w  | Direktory aktif                                                 |
| \t  | time dalam 24 jam dengan format HH:MM:SS                        |
| dll | man bash : -)                                                   |

contoh pemakaiannya:

```
[you@slashsmart you]$PS1="\t[\u@\h:\w]\$"
[07:14:04][you@slashsmart:~]$_
```

- Agar prompt shell hasil konfigurasi anda dapat tetap berlaku setiap kamu login (permanen) sisipkan pada file `.bashrc` atau `.bash_profile`

## Positional Parameter

- Parameter posisi yaitu variabel yang digunakan shell untuk menampung argumen yang diberikan terhadap shell baik berupa argumen waktu sebuah file dijalankan atau argumen yang dikirim ke subrutin.
- Variabel yang dimaksud adalah 1,2,3,dst..lebih jelasnya lihat contoh script berikut :

```
[you@slashsmart you]$ cat argumensatu
#!/bin/bash
#filename:argumensatu
```

```
echo $1 suka sekali belajar $2 khususnya $3
```

Jika di eksekusi dengan memberikan argumen seperti berikut :

```
[you@slashsmart you]$ ./argumensatu Josh Linux Mandrake
```

Josh suka sekali belajar Linux khususnya Mandrake

```
[you@slashsmart you]$
```

- Pada proses eksekusi script diatas ada 3 argumen yang disertakan pada script argumen1 yaitu Josh, Linux, Mandrake, masing2 argumen akan disimpan pada variabel 1,2,3 sesuai posisinya.
- Variabel spesial lain yang dapat digunakan diperlihatkan pada script berikut:

```
[you@slashsmart you]$ cat argumendua
```

```
#!/bin/bash
```

```
#filename:argumendua
```

```
clear
```

```
echo "Filename script anda....: $0";
```

```
echo "Banyak argumen.....: $#";
```

```
echo "Argumen yg anda masukkan: $*";
```

```
[you@slashsmart you]$
```

- Jika di eksekusi dengan memberikan beberapa argumen seperti :

```
[you@slashsmart you]$ ./argumendua satu dua 3 4
```

```
Filename script anda....: ./argumendua
```

```
Banyak argumen.....: 4
```

```
Argumen yg anda masukkan: satu dua 3 4
```

```
[you@slashsmart you]$
```

### Input dengan read

- Menggunakan statement read yang cukup ampuh untuk membaca atau menerima masukan dari input standar

syntax :

```
read -opsi [nama_variabel...]
```

- Berikut contoh scriptnya:

```
[you@slashsmart you]$ cat readsatu
#!/bin/bash
#filename:readsatu

echo -n "Masukkan nama anda : "
read name;

echo "Selamat siang $name, apa khabar?";
echo -n "Keadaan gue : ";
read
echo "Nama kamu $name, keadaan kamu $REPLY";

[you@slashsmart you]$
```

- Setelah dieksekusi hasilnya:

```
[you@slashsmart you]$ ./readsatu
Masukkan nama anda :josh
Selamat siang josh, apa khabar?
Keadaan gue :sedang sedih
Nama kamu josh, keadaan kamu sedang sedih
[you@slashsmart you]$
```

- Jika nama\_variabel tidak disertakan, maka data yang diinput akan disimpan di variabel REPLY.

## Administrasi System, User dan Group

### Menganalisis system Linux

- Menganalisis sistem biasanya dilakukan oleh seorang administrator sistem untuk mendeteksi adanya gejala-gejala yang dapat menyebabkan kerusakan atau masalah pada sistem kelak.
- Linux Mandrake sudah menyediakan tool-tool untuk melakukan hal itu, diantaranya adalah dua buah program tool yang sangat berguna yaitu vmstat dan top.
- Dengan menggunakan kedua program tersebut administrator memperoleh informasi mengenai sistem seperti penggunaan CPU, memori atau proses-proses yang sedang berjalan dan dengan bantuan informasi ini administrator sistem bisa mendeteksi secara cepat apakah sistemnya mengalami gangguan atau tidak

### Menggunakan vmstat

- Program atau perintah ini akan melakukan test dan menampilkan informasi penggunaan CPU, memori, proses-proses yang sedang berjalan serta operasi I/O. Sintaks penulisan perintahnya adalah sebagai berikut:

```
$ vmstat [interval] [count]
```

- Interval adalah waktu jeda test dalam detik dan count adalah jumlah test yang kita kehendaki. Misalnya akan dilakukan test sebanyak 3 kali dengan waktu jeda setiap 4 detik maka perintah dan hasilnya adalah seperti dibawah ini :

```
[you@slashsmart you]$ vmstat 4 3
procs          memry  swap      io  system      cpu
r b w  swpd  free  buff  cache  si so  bi  b  in  cs us sy id
1 0 0 44400 2328 1548 45960 7 14 107 21 3300 916 22 8 71
2 0 0 44400 2184 1548 46088 0 0 32 0 3370 971 12 6 82
4 0 0 44400 2180 1548 46088 0 0 0 0 3342 917 11 5 84
[you@slashsmart you]$
```

- Tiga kolom pertama menunjukkan proses-proses yang sedang dikerjakan, yaitu waiting for runtime(r), uninterrupted sleep(b) dan swapped out(w).

- Banyaknya proses yang mengalami waiting for runtime(r) dapat merupakan indikasi terjadinya suatu masalah misalnya telah terjadi bottleneck yaitu penumpukan proses-proses disuatu tempat.
- Kolom memory dan swap digunakan untuk mendeteksi adanya kesalahan yang ditimbulkan oleh manajemen memori. Kolom-kolom swpd, free, buff dan cache berturut-turut menjelaskan jumlah memori virtual yang digunakan, jumlah memori idle, jumlah memori yang dipakai sebagai buffer dan jumlah memori yang tersisa dalam cache, semuanya dalam KB. Perhatikan kolom swap in(si), merupakan jumlah memori yang di-paging dari disk dalam satuan KB/detik sedangkan swap out(so) adalah kebalikannya.
- Tiga kolom terakhir adalah persentasi penggunaan CPU yaitu persentasi penggunaan CPU untuk tugas-tugas user(us), persentasi penggunaan CPU untuk tugas-tugas sistem termasuk waktu tunggu I/O, pelaksanaan fungsi-fungsi sistem operasi secara umum(sy) dan persentasi CPU idle atau saat tidak digunakan(id).
- Untuk mendeteksi suatu kesalahan yang diakibatkan oleh masalah CPU sebaiknya dengan melihat persentasinya secara signifikan dalam satu periode waktu. Misalnya pada saat komputer idle dalam satu periode waktu, perhatikan kolom us dan sy seharusnya menunjukkan angka yang rendah sedangkan kolom id menunjukkan angka yang tinggi. Bila tidak seperti itu maka dipastikan ada masalah yang berkaitan dengan CPU

### **Menggunakan top**

- Menampilkan informasi proses yang sedang dilakukan CPU secara real time.
- Kelebihan dari top ini adalah kita dapat memanipulasi proses-proses tersebut secara interaktif dengan menekan tombol-tombol menu.
- Misalnya untuk meng-kill sebuah proses, kita cukup menekan tombol K dan mengisi nomor proses(PID) yang ingin di-kill.

**[you@slashsmart you]\$ top**

11:11pm up 59 min, 4 users, load average: 0.80, 0.76, 0.61

82 processes: 79 sleeping, 3 running, 0 zombie, 0 stopped

CPU states: 17.9% user, 10.8% system, 0.0% nice, 71.1% idle

Mem: 93920K av, 89488K used, 4432K free, 0K shrd, 1560K buff

Swap: 208804K av, 44820K used, 163984K free 43372K cached

| PID  | USER | PRI | NI | SIZE  | RSS  | SHARE | STAT | %CPU | %MEM | TIME | COMMAND     |
|------|------|-----|----|-------|------|-------|------|------|------|------|-------------|
| 2038 | josh | 14  | 0  | 2804  | 2252 | 1648  | R    | 8.6  | 2.3  | 4:50 | artsd       |
| 2068 | josh | 11  | 0  | 25712 | 22M  | 19116 | S    | 7.3  | 24.0 | 3:29 | soffice.bin |
| 2576 | josh | 10  | 0  | 6652  | 5128 | 2104  | S    | 3.2  | 5.4  | 0:06 | xmms        |
| 2583 | you  | 14  | 0  | 1048  | 1048 | 816   | R    | 2.3  | 1.1  | 0:00 | top         |
| 1892 | root | 12  | 0  | 50448 | 8320 | 1884  | R    | 1.9  | 8.8  | 2:26 | X           |
| 2051 | josh | 12  | 0  | 5396  | 4104 | 2984  | S    | 1.1  | 4.3  | 0:11 | kdeinit     |
| 2098 | josh | 9   | 0  | 5828  | 4772 | 3776  | S    | 0.9  | 5.0  | 0:11 | kdeinit     |
| 2046 | josh | 10  | 0  | 3948  | 2616 | 2084  | S    | 0.3  | 2.7  | 0:05 | kdeinit     |
| 2064 | josh | 9   | 0  | 2160  | 968  | 756   | S    | 0.1  | 1.0  | 0:00 | korgac      |

## Administrasi User Dan Group

### Sekuritas dan hak akses file/direktori :

- Karena Linux bersifat multiuser, maka harus ada mekanisme untuk mencegah akses-akses ilegal terhadap file.
- Prinsip-prinsip pengaturan hak akses atas file :
- Setiap file selalu dimiliki oleh seseorang yang berhak mengatur hak akses terhadap file tersebut.
- Hak akses terbagi menjadi tiga bagian: baca, tulis, dan eksekusi. Hak-hak ini dapat diberikan ke 3 kelompok pemakai: pemilik file, group dimana si pemilik bergabung, dan semua pemakai lainnya.
- Hak akses ditunjukkan dengan susunan kombinasi 3 bit dalam 3 bilangan oktet.

|         |           |
|---------|-----------|
| Bit ke- | Hak Akses |
| 1       | Baca      |
| 2       | Tulis     |
| 3       | Eksekusi  |

|                |                  |
|----------------|------------------|
| Bil. Oktet ke- | Kelompok Pemakai |
| 1              | Pemilik file     |
| 2              | Group            |
| 3              | Pemakai lain     |

|  |         |   |   |         |   |   |         |   |   |
|--|---------|---|---|---------|---|---|---------|---|---|
|  | Oktet 1 |   |   | Oktet 2 |   |   | Oktet 3 |   |   |
|  | r       | w | x | r       | w | x | r       | w | x |

- Hak akses dapat berubah dengan perintah 'chmod'. Hanya si pemilik file atau root yang dapat melakukannya.

### Apa yang dapat dilakukan sebagai USER ?

- Membuat direktori ( mkdir )

Perintahnya adalah

```
mkdir nama_directory
```

- Merubah hak akses pada file/direktori anda ( chmod )

Untuk permission, ada tiga macam : ijin membaca (r), ijin menulis (w) dan ijin menjalankan (x)

```
-rw-r--r-- 1 you you 163 Jan 12 12:00 myfile
```

'-' pertama menunjukkan bahwa file ini adalah file biasa (bukan direktori).

---

***Bila myfile adalah suatu direktori, maka '-' pertama akan digantikan oleh d, dan bila myfile suatu link maka akan digantikan oleh l.***

---

- Misalnya kesembilan karakter lainnya dipecah menjadi tiga kelompok dengan ijin rwx (baca, tulis, dan menjalankan) untuk pemilik, grup dan others ( pemakai selain user dan grup yang bersangkutan. ) Dalam kasus myfile, pemilik boleh membaca dan menulis (rw), grup bisa membaca (r) dan semua boleh membaca (r).
- Jadi ketiga kelompok mendapatkan semua ijin akan menjadi  
-rwxrwxrwx 1 you you 163 Jan 12 12:00 myfile

- myfile akan berubah menjadi *executable*. Tentu saja bila myfile tidak berupa file binary (program terkompilasi) atau program interpreted (basic, perl, dll) atau shellscript, dia tetap tidak akan bisa berjalan. Sebetulnya grup rwx adalah interpretasi dari tiga bilangan biner (mis --w = 001), sebagai setelah dikonversi menjadi desimal menjadi:

---

|            |   |                                  |
|------------|---|----------------------------------|
| 0 atau --- | : | tidak ada permission sama sekali |
| 4 atau r-- | : | baca saja                        |
| 2 atau -w- | : | tulis saja (jarang)              |
| 1 atau --x | : | eksekusi                         |
| 6 atau rw- | : | baca tulis                       |
| 5 atau r-x | : | baca dan eksekusi                |
| 3 atau -wx | : | tulis dan eksekusi (jarang)      |
| 7 atau rwx | : | baca tulis dan eksekusi          |

---

- Untuk mengubah permission gunakan perintah **chmod**. Bisa dengan kode numerik (**chmod 775 myfile** = -rwxrwx-r.) Atau menggunakan kode huruf (u --> user, g --> group, a --> all) lalu diikuti tanda + untuk menambahkan permission atau - untuk mengurangi permissions. Misal **chmod go + r myfile** menambahkan permission r pada untuk group dan others pada myfile.)

- Merubah hak kepemilikan pada file/direktori anda ( chown )

Gunakan perintah chown untuk mengganti ownership file.

Contoh :

```
chown josh.josh nama_file
```

( pada Mandrake secara default tiap user mempunyai grup sendiri).

Perintah **chown josh.josh myfile** mengubah file myfile menjadi milik josh dan grup josh. Selanjutnya silakan baca **man chown**.

#### **Apa yang dapat dilakukan sebagai SUPERUSER ?**

- Menambah user/group ( adduser / groupadd )

- Perintah untuk menambah user adalah menggunakan perintah **adduser user**. Perintah ini sebenarnya sebuah script yang memakai program dasar yang terdapat pada UNIX yaitu perintah **useradd**, kemudian dimodifikasi untuk otomatis mengeset environment user baru seperti home dir, shell , group dll. Jadi bila kita memakai perintah useradd harus mengeset itu semua sendiri dalam satu jalur perintah.

- Menghapus user/group ( deluser )

Perintah untuk menghapus user adalah dengan menggunakan perintah :

**userdel user**

- Untuk memodifikasi environment yg telah ada maka gunakan perintah *usermod* user. Parameter-parameter yang mengikuti dapat dibaca pada man userdel dan man usermod.
- Merubah hak akses dan kepemilikan file dan direktori  
Sama dengan perintah untuk mengubah hak akses dan kepemilikan file pada direktori anda. Tapi untuk mengubah file, atau direktori diluar direktori home , anda memerlukan priveledge root.

### **Bekerja dengan Group**

- Dalam Linux, file-file di-*share* (digunakan secara bersama-sama) oleh para pengguna dan sekelompok group dari pengguna. Jika seorang user menginginkan file yang menjadi haknya tidak dapat diakses oleh orang lain maka hal tersebut dapat dilakukannya. Pada kebanyakan sistem, *default*-nya adalah membiarkan pengguna yang lain untuk membaca file-file anda tetapi tidak dapat memodifikasinya atau menghapusnya dengan cara apapun juga.
- Setiap file dimiliki oleh pengguna tertentu. Tetapi file-file juga dimiliki oleh **group** tertentu, yaitu **group** yang didefinisikan untuk sekelompok pengguna. Setiap pengguna ditempatkan pada paling tidak satu group ketika account user tadi dibuat. Tetapi *seorang sistem administrator dapat memberikan seorang user untuk*

*mengakses lebih dari satu group.*

- Group biasanya didefinisikan oleh tipe dari pengguna yang mengakses ke mesin. Misalnya pada sebuah sistem Linux di Universitas, para pengguna kemungkinan ditempatkan pada group **wheel**, **student**, **staff**, **faculty** atau **guest**. Juga terdapat beberapa group yang didefinisikan oleh sistem (seperti **bin** dan **admin**) yang digunakan oleh sistem itu sendiri untuk melakukan akses kontrol ke resources, sangat jarang pengguna aktual masuk dalam group ini.
- File `/etc/group` berisi informasi mengenai group-group yang ada dalam sistem. Format dari tiap barisnya adalah sebagai berikut.

```
group name:password:GID:other members
```

Beberapa contoh group-group:

```
root*:0:
```

```
users*:500:josh,bgnight
```

```
guest*:510:
```

```
other*:520:you
```

- Group pertama, `root`, adalah group sistem spesial yang disediakan untuk `root`. Group `users`, adalah untuk pengguna reguler. Memiliki GID 500. Pengguna `josh` dan `lognight` memiliki akses ke group tersebut. Ingat bahwa di `/etc/passwd` tiap pengguna diberikan sebuah GID default. Perintah `groups` memperlihatkan daftar group apa saja anda diberikan akses.
- Group ketiga, `guest`, adalah untuk pengguna `guest`, dan `other` untuk pengguna "other". User `you` juga diberikan akses ke group ini.
- Field "password" dari `/etc/group` terkadang digunakan untuk mengeset password pada akses group. Untuk memproteksi pengguna mengubah group ke group yang lain (dengan perintah `newgroup`), set field password ke "\*".
- Perintah `addgroup` atau `groupadd` dapat digunakan untuk menambahkan group-group ke sistem anda.
- Biasanya, lebih mudah untuk langsung menambahkan entri ke `/etc/group`. Dan untuk menghapusnya tinggal menghapus pada file `/etc/group`.

## Membuat backup data

- Sebelum kita berbicara mengenai backups, kita perlu untuk memperkenalkan tool-tool untuk melakukan archiving file-file pada sistem UNIX.

### Menggunakan tar.

- Perintah tar paling sering digunakan untuk meng-archive file-file. Sintaks perintahnya adalah :

```
tar options files
```

- dimana options adalah daftar dari perintah-perintah dan opsi-opsi bagi tar, dan files adalah daftar file-file yang ditambahkan atau di-ekstrak dari archive.

Sebagai contoh :

```
# tar cvf backup.tar /etc
```

- Mengepak semua file-file dari /etc ke tar archive backup.tar.

c -> menyuruh tar membuat file archive baru.

v -> agar tar menggunakan mode verbose, mencetak tiap nama file sesuai yang diarchive.

f -> memberitahu tar bahwa argumen berikutnya, backup.tar, adalah nama dari archive yang akan dibuat.

- Argumen sisanya adalah nama-nama file dan direktori dari akan ditambahkan ke archive.

```
# tar xvf backup.tar
```

akan meng-ekstrak file tar backup.tar pada direktori dimana kita berada sekarang.

- Dan file yang diekstrak oleh sistem akan ditempatkan pada kedudukan aslinya. Jadi bila file atau direktori pada tempat asalnya terhapus, dan kita mempunyai cadangan (back up) dalam bentuk file tar, maka kita dapat langsung mengekstraknya.

- Sebelum meng-ekstrak file tar, penting adanya untuk mengetahui dimana file-file harus di-unpak. Misal anda men-archive file-file berikut:

```
/etc/hosts, /etc/group, /etc/passwd.
```

Jika anda menggunakan perintah:

```
# tar cvf backup.tar /etc/hosts /etc/group /etc/passwd
```

Nama direktori /etc/ ditambahkan ke ke permulaan dari tiap nama file.

Untuk mengekstrak ke lokasi yang benar gunakan

```
# cd /  
# tar xvf backup.tar
```

karena file-file diekstrak dengan nama path disimpan pada file archive.

Sedangkan perintah :

```
# tar tvf backup.tar
```

dapat digunakan untuk menampilkan daftar dari file-file archive tanpa mengekstraknya.

- Ada sedikit catatan, bila kita memindahkan file tar yang kita buat ke direktori lain (bukan direktori tempat kita mengarchive file atau direktori tadi), kita akan mendapatkan hasil yang berbeda bila kita mengekstraknya. Misalnya, kita mengarchive direktori /home/yogi di direktori / . Setelah itu kita membuat salinannya di direktori /var, maka bila kita mengekstraknya maka kita akan mendapatkan hasil yang berbeda. Perbedaannya cuma letak hasil file yang diekstrak.

### **Menggunakan gzip.**

- Tidak seperti program archiving bagi MS-DOS, tar tidak secara otomatis melakukan kompresi file-file ketika melakukan archiving.
- Jika anda melakukan dua archiving, file-file 1-megabyte, menghasilkan file tar dalam dua megabyte ukurannya. Perintah gzip melakukan kompresi suatu file (tidak harus file tar). Perintahnya :

```
# gzip -q backup.tar
```

mengkompresi backup.tar dan menjadikannya backup.tar.gz.

- Perintah gunzip atau gzip -d dapat digunakan untuk meng-unkompresi suatu file yang ter-gzip.
- Untuk melakukan archive dan kompresi sekaligus, gunakan perintah:

```
# tar cvf backup.tar /etc
```

```
# gzip -q backup.tar
```

yang ekuivalen dengan perintah:

```
# tar cvfz backup.tar.gz /etc
```

Hasilnya backup.tar.gz.

- Untuk kebalikannya (mengestraknya) gunakan perintah:

```
# gunzip backup.tar.gz
```

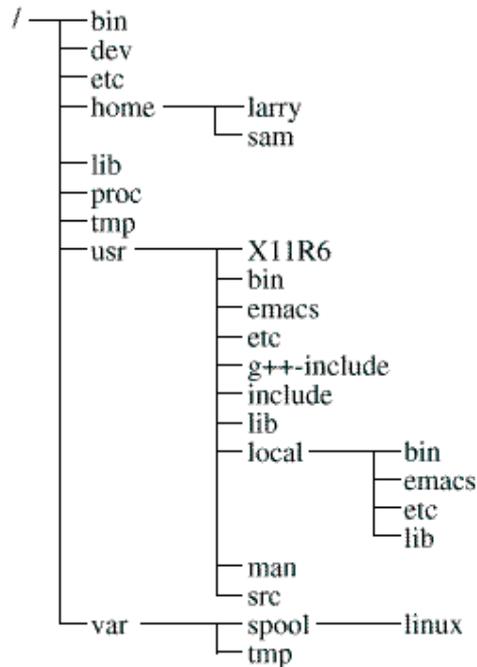
```
# tar xvf backup.tar
```

yang ekuivalen dengan perintah:

```
# tar xvfz backup.tar.Z
```

## File System Linux

- Hierarki dari file sistem di Linux dapat digambarkan sebagai berikut :



### Hirarki Direktori dari Suatu Sistem Linux

- Suatu **file system** adalah kumpulan file dan hirarki direktori-direktori pada suatu sistem. Anda dapat melihat peta dari file system Linux pada bagan di atas. Pertama kali, pindah ke direktori root (`cd /`), dan kemudian ketikkan `ls -F` untuk menampilkan daftar dari isinya. Maka anda akan melihat direktori-direktori sebagai berikut: `bin`, `dev`, `etc`, `home`, `install`, `lib`, `mnt`, `proc`, `root`, `tmp`, `user`, `usr`, dan `var`.

Sekarang, marilah kita lihat dari tiap-tiap direktori ini.

#### **/bin**

Merupakan kependekan dari “*binaries*”, atau *executables*, dimana banyak terdapat program-program sistem yang esensial. Gunakan `ls -F /bin` untuk melihat file-filenya. Jika anda melihat daftar file-nya, anda akan melihat beberapa commands yang telah anda kenal, seperti `cp`, `ls`, dan `mv`. Ketika anda menggunakan perintah `cp`, maka anda berarti menjalankan program `/bin/cp`. Dengan menggunakan `ls -F`, anda akan melihat bahwa sebagian besar (tidak semuanya) dari file-file di `/bin` memiliki tanda (“\*”). Ini menandakan file tersebut dapat dieksekusi.

#### **/dev**

File-file di `/dev` merupakan file-file device—file-file tersebut mengakses *system devices* dan *resources* (sumber daya) seperti *disk drive*, *modems*, dan *memory*. Misalnya, untuk membaca input dari mouse maka dilakukan dengan mengakses

/dev/mouse. Nama file yang diawali dengan fd adalah floppy disk devices. fd0 adalah floppy disk drive pertama, dan fd1 merupakan yang kedua.

Beberapa file device yang umum:

| <b>Device</b>          | <b>Kegunaannya:</b>                                                                                                                                                                                                           |
|------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| /dev/console           | Mengacu pada system's console—yaitu monitor yang terkoneksi secara langsung ke sistem anda.                                                                                                                                   |
| /dev/ttys dan /dev/cua | untuk mengakses port serial.<br>/dev/ttys0 mengacu ke "COM1" seperti pada MS-DOS.<br>/dev/cua merupakan "callout" devices" dan digunakan dengan modem.                                                                        |
| /dev/hd[...]           | Akses ke harddisk. /dev/hda mengacu kepada seluruh harddisk pertama, sedangkan /dev/hda1 mengacu pada partisi pertama dari /dev/hda.                                                                                          |
| /dev/sd[...]           | merupakan drive SCSI. Tape SCSI diakses lewat device st, dan CD-ROM SCSI melalui device sr.                                                                                                                                   |
| /dev/lp[...]           | Mengakses port paralel. /dev/lp0 sama dengan "LPT1" pada dunia MS-DOS.                                                                                                                                                        |
| /dev/null              | Sebagai "black hole"—data dikirim pada device ini akan hilang selamanya. Mengapa ini berguna? Jika anda ingin menyembunyikan output dari perintah yang terlihat pada layar anda, maka anda dapat mengirimkannya ke /dev/null. |
| /dev/tty[...]          | Mengacu pada virtual console pada sistem anda. (Diakses dengan menekan Alt-F1, Alt-F2, dan seterusnya). /dev/tty1 mengacu pada VC pertama anda.                                                                               |
| /dev/pty[...]          | Pseudo-terminal, digunakan untuk menyediakan suatu terminal ketika sesi login jarak jauh. Misal, ketika terhubung ke sistem lain dengan menggunakan telnet.                                                                   |

### **/etc**

Berisi sejumlah file-file konfigurasi sistem. Termasuk /etc/passwd (basis data pengguna), /etc/rc (script inialisasi sistem), dan seterusnya.

### **/sbin**

Berisi binary-binary sistem yang esensial dimana digunakan untuk administrasi sistem.

### **/home**

Berisi direktori rumah dari para pengguna sistem. Sebagai contoh /home/josh merupakan home directory dari user "josh".

**/lib**

Berisi shared library images, merupakan file-file yang berisi kode dimana banyak program-program menggunakannya secara umum.

**/proc**

/proc mendukung “*virtual file system*”, dimana file-file disimpan pada memori, tidak pada disk. File-file ini mengacu pada berbagai proses yang jalan pada sistem, sehingga dapat memberikan anda informasi tentang program-program dan proses-proses yang berjalan pada saat tertentu.

**/tmp**

Banyak program menyimpan informasi sementara di sini.

**/usr**

/usr merupakan direktori yang sangat penting dimana berisi sub-direktori yang berisi beberapa dari program-program dan konfigurasi file-file yang digunakan pada sistem yang paling penting. Isinya antara lain:

| Direktori        | Kegunaannya:                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| /usr/X11R6       | /usr/X11R6 berisi X Window System, jika anda menginstallnya. X Windows merupakan sistem GUI ( <i>Graphical User Interface</i> ) pada sistem Linux.                                                                                                                                                                                                                                                                                                                                                      |
| /usr/bin         | /usr/bin adalah gudang software yang nyata pada sembarang sistem Linux, berisi sebagian besar dari executable file bagi program yang tidak ditemukan pada tempat lain seperti /bin.                                                                                                                                                                                                                                                                                                                     |
| /usr/etc         | berisi berbagai file-file dan utilitas yang pada umumnya tidak esensial terhadap sistem.                                                                                                                                                                                                                                                                                                                                                                                                                |
| /usr/include     | /usr/include berisi include files bagi kompilator C. File-file ini (sebagian besar diakhiri dengan .h, untuk header) mendeklarasikan nama-nama struktur data, subroutine, dan konstan yang digunakan ketika menulis program pada C.<br>File-file pada /usr/include/sys pada umumnya digunakan ketika memprogram pada level sistem UNIX. Jika anda familier dengan bahasa pemrograman C, di sini anda akan menemukan file header seperti stdio.h, dimana mendeklarasikan fungsi-fungsi seperti printf(). |
| /usr/g++-include | Berisi file-file bagi kompilator C++                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| /usr/lib         | Berisi library “stub” dan “static” ekuivalen dengan file-                                                                                                                                                                                                                                                                                                                                                                                                                                               |

|            |                                                                                                                                                                                                 |
|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|            | file pada /lib.                                                                                                                                                                                 |
| /usr/local | Berisi berbagai program dan file yang tidak esensial bagi sistem (hampir seperti /usr). Hanya saja program tersebut lebih ke sistem.                                                            |
| /usr/man   | Direktori ini berisi halaman manual. Terdiri atas dua subdirektori.                                                                                                                             |
| /usr/src   | Berisi source code (instruksi yang belum terkompilasi bagi berbagai program pada sistem anda). Direktori paling penting disini adalah /usr/src/linux yang berisi source code bagi kernel linux. |

### **/var**

Berisi direktori yang sering berubah ukurannya. Misal basis data.

| <b>Direktori</b> | <b>Kegunaannya:</b>                                                                                                                                                                                                                                                        |
|------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| /var/log         | berisi berbagai file untuk administrasi sistem, terutama system logs, dimana mencatat error atau masalah pada sistem.                                                                                                                                                      |
| /var/spool       | Berisi file-file yang “spooled” ke program yang lain. Contoh, jika mesin anda terkoneksi ke suatu network, mail yang datang disimpan pada /var/spool/mail sampai anda membacanya atau menghapusnya. Artikel yang datang dan keluar ada di /var/spool/news, dan sebagainya. |

### **Konsep File Device dan Konfigurasi File**

- Dalam Linux, piranti-piranti komputer seperti harddisk, printer, dan mouse diakses dengan cara memetakannya ke dalam suatu file yang dinamakan device file. Letak dari file-file device tersebut berada di direktori /dev. Mereka diklasifikasikan menurut tipe, antarmuka, dan urutan penginstall-annya.
- Skema penamaannya dapat ditabelkan sebagai berikut.

*Device driver*

*Drive*

|           |                                      |
|-----------|--------------------------------------|
| /dev/hda  | Master IDE drive, primary IDE bus.   |
| /dev/hdb  | Slave IDE drive, primary IDE bus.    |
| /dev/hdc  | Master IDE drive, secondary IDE bus. |
| /dev/hdd  | Slave IDE drive, secondary IDE bus.  |
| /dev/sda  | First SCSI hard drive.               |
| /dev/sdb  | Second SCSI hard drive.              |
| /dev/st0  | First SCSI tape drive.               |
| /dev/st1  | Second SCSI tape drive.              |
| /dev/scd0 | First SCSI CD-ROM drive.             |
| /dev/scd1 | Second SCSI CD-ROM drive.            |

## KONFIGURASI JARINGAN

### Jaringan Komputer

- Jaringan komputer adalah beberapa (dua atau lebih) komputer yang saling terhubung satu dengan yang lainnya dengan menggunakan interface tertentu, interface yang digunakan adalah network card (biasanya disebut LAN card).
- Dengan terhubung satu dengan yang lainnya, komputer-komputer itu bisa saling berbagi resource (sumber daya) yang dimiliki, misalnya space pada hardisk.
- Pada linux, dikenal juga adanya jaringan komputer. Bahkan linux akan terasa lebih bermanfaat apabila terhubung ke jaringan komputer, sebab linux memang didesain untuk beroperasi secara maksimal pada jaringan komputer.
- Agar dapat terhubung ke jaringan, kita harus mengatur beberapa parameter. Parameter-parameter tersebut adalah alamat IP, dan subnet mask. Tapi, langkah awal yang terpenting agar satu komputer dapat terhubung ke jaringan adalah bagaimana caranya network card dapat dikenali oleh sistem kita.

### Konfigurasi LanCard

- Pada linux sendiri, ada beberapa cara untuk mengkonfigurasi network card. Hal lain yang perlu dijadikan perhatian adalah pemilihan network card, yang biasanya kita gunakan adalah LAN card PCI.

- Sedangkan LAN card dengan tipe ISA sudah jarang lagi digunakan, karena sudah tidak lagi diproduksi oleh industri. Keuntungan LAN card PCI adalah konfigurasinya yang tidak begitu rumit, dan biasanya langsung terdeteksi oleh linux-linux distro yang baru.
- Sedangkan untuk LAN card tipe ISA, beberapa distro linux yang baru kadang-kadang tidak menyiapkan modul-modul driver yang mendukung hardware tersebut.
- Di Linux, kartu jaringan kita dikenal dengan nama eth. Misalnya kita mempunyai dua kartu jaringan dalam satu mesin, satu kartu jaringan akan bernama eth0 dan yang lain akan bernama eth1.
- Langkah pertama dalam menset LAN card adalah mengecek apakah LAN card itu telah terkonfigurasi dengan baik. Caranya dengan mengetikkan perintah :

```
# ifconfig eth0
```

- Bila telah terkonfigurasi dengan baik, maka kita tidak perlu mendeteksi lagi apakah LAN card kita telah ada. Kita tinggal menyesuaikan alamat IP kita dengan jaringan tempat kita bergabung.
- Bila ternyata belum terdeteksi, maka kita harus membangkitkan modul yang diperlukan agar LAN card kita dapat bekerja. Caranya dengan mengetikkan perintah :

```
# modprobe ne2k-pci
```

- Parameter ne2k-pci diatas merupakan jenis modul yang diperlukan oleh LAN card NE2000 yang tipenya PCI. Apabila LAN card yang digunakan ternyata bukan tipe NE2000 maka driver yang digunakan harus sesuai dengan jenisnya.
- Setelah ada pesan bahwa sistem mendeteksi adanya LAN card dengan tipe yang dimaksud, maka kita sekarang tinggal mengisi parameter yang diperlukan, yaitu nomor IP dan netmask. Caranya ketikkan perintah dibawah ini:

```
# ifconfig eth0 192.168.20.xxx netmask 255.255.255.0
```

- Parameter xxx merupakan nomor yang diisi secara unik untuk masing-masing komputer. Sedangkan 192.168.20 adalah nama jaringan yang ditempati oleh komputer yang bersangkutan.
- Selain menggunakan perintah **# ifconfig**, kita dapat juga

mengkonfigurasi LAN card menggunakan tools yang disediakan. Misalnya **# linuxconf** atau **# kudzu**. Penggunaan **# kudzu** sangatlah mudah, kita tinggal mengetikkan perintah **# kudzu** dan ikuti petunjuk dan pilihan yang ada. Serta kita akan memasukkan nomor IP komputer kita dan netmask-nya. Sedangkan untuk **# linuxconf**, caranya adalah:

- ketikkan **# linuxconf**
- lalu pilih baris **networking**
- kemudian pilih barisan **Hostname and IP network devices**
- setelah masuk, lalu pilih bagian **adapter 1** untuk mengisi hal-hal yang diperlukan seperti:
  - a. **hostname + domain** : nama komputer (penamaannya bebas, sesuai nama komputer masing-masing).
  - b. Pilih option **enable** dan **manual** (kedua option diberi tanda pilih)
  - c. **Primary name + domain** : **indra** (sesuai nama hostname)
  - d. **IP address** : **172.16.40.xxx** (nomor IP komputer kita)
  - e. **Net device** : **eth0**
  - f. **Kernel Module** : **ne2k-pci** (sesuai network adapter masing-masing yang digunakan).
- Bila memiliki lebih dari satu network adapter, maka langkah diatas diulangi untuk bagian adapter 2.
- Cara laen untuk mendeteksi dan mengkonfigurasi jaringan dengan Linux Mandrake yakni dengan menggunakan tool yakni **drakconnect** yang bisa di jalankan user **root**.
- Cara laennya melakukan setting dua kartu jaringan dalam satu mesin. Diasumsikan ke-dua kartu jaringan tersebut adalah **PCI** dan **NE2000 compatible**. Langkah-langkahnya adalah sebagai berikut :
  1. Edit file **/etc/conf.modules** dengan editor Anda, misalnya dengan **mcedit** sebagai berikut:
 

```
[root@server /root] # mcedit /etc/conf.modules
```

 Edit atau tambahkan baris-baris berikut :
 

```
alias      eth0  ne2k-pci
alias      eth1  ne2k-pci
```
  2. Edit konfigurasi untuk **eth0** dan **eth1**, dengan perintah sebagai berikut :
 

```
[root@server /root] # cd /etc/sysconfig/network-scripts
[root@server /root] # mcedit ifcfg-eth0
```

 Edit file **ifcfg-eth0**, sesuaikan dengan jaringan Anda, misalnya sebagai berikut :
 

```
DEVICE = eth0
IPADDRESS = 192.168.20.xx
NETMASK = 255.255.255.0
NETWORK = 192.168.20.0
BROADCAST = 192.168.0.255
ONBOOT = YES
```

 Lakukan hal yang sama untuk **eth1**
  3. Edit file **/etc/sysconfig/network** , sesuaikan dengan jaringan Anda, misalnya sebagai berikut:

```
NETWORKING = yes
FORWARD_IPV4 = true
HOSTNAME = server.pelatihan.net
DOMAINNAME = pelatihan.net
GATEWAY = 192.168.20.x
GATEWAYDEV = eth0
```

- Setelah selesai, kita ketikkan perintah ini :

```
# /etc/rc.d/init.d/network restart
```

- Perintah ini berguna untuk mengaktifkan konfigurasi network yang telah kita buat. Setelah itu, kita coba apakah network card kita telah terkonfigurasi dengan baik. Kita ketikkan perintah # ping 192.168.20..xxx (sesuai IP address komputer kita). Bila ada tanggapan (reply) dari alamat yang kita maksud, maka network card kita berhasil di konfigurasi.
- Sekarang kita akan mencoba apakah jaringan kita telah terkoneksi dengan baik. Caranya dengan mengetikkan perintah berikut ini # ping 192.168.20.xxx (diisi dengan IP address dari komputer lain dalam satu jaringan). Bila ada balasan (reply) dari komputer yang dituju, berarti komputer tersebut telah terkonfigurasi dengan baik pada jaringan kita.

## Remote Login

- Kita akan mencoba mensimulasikan layanan server dalam jaringan kita pada saat ini, yaitu ssh (secure shell).
- Sebenarnya fungsi program ini adalah untuk melakukan login dari terminal yang berbeda. Untuk dapat melakukannya, kita harus mempunyai login di

komputer yang menjadi ssh server. Berarti salah satu komputer disini dijadikan server ssh. Caranya mudah saja, pertama-tama kita coba lihat apakah paket ssh dan sshd sudah terinstal di komputer yang akan dijadikan ssh server.

- Bila sudah kita akan mengaktifkan servis ssh dengan cara, ketikkan

```
# /etc/rc.d/init.d/sshd start.
```

Setelah itu, silakan login ke komputer yang dijadikan ssh server.

- Dengan yang cara yang sama kita dapat mengaktifkan services telnet, yang berbeda adalah nama daemon yang di hidupkan :

```
# /etc/rc.d/init.d/telnetd start
```