

LINUX

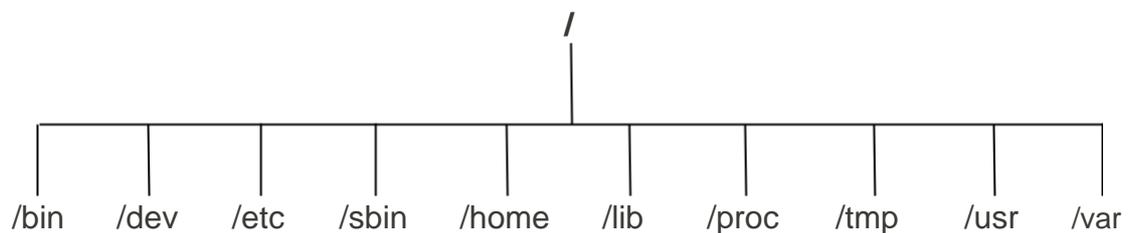
STMIK BANISALEH



HIRARKI SISTEM LINUX

Setelah kuliah ini, Mahasiswa diharapkan akan dapat :
Mengetahui Hirarki Linux
Fungsi direktori dalam hirarki linux

Topik yang akan dipelajari :
Struktur hirarki linux



Seperti halnya DOS dan Windows sistem file pada linux tersusun atas direktori-direktori lain yang tersusun secara hirarki dari dengan direktori utamanya '/' atau dikenal dengan "**root directory**", dan di bawahnya terdapat lagi direktori yang merupakan tempat penyimpanan konfigurasi-konfigurasi sistem yang ada pada linux

/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.





Tabel 4.1. Mengenal device sebagai direktori /dev di linux

Device	Kegunaannya:
/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. /dev/ttys0 mengacu ke "COM1" seperti pada MS-DOS. /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. Terkadang digunakan juga untuk mendeteksi device flashdisk
/dev/lp[...]	Mengakses port paralel. /dev/lp0 sama dengan "LPT1" pada 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 biasanya sampai Alt-F7 tergantung setting system anda). /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 konfigurasi sistem., seperti /etc/rc (script inialisasi sistem)

/sbin

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

/home

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

/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.





BAB



/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:

Tabel 4.2. Isi direktori /usr

Direktori	Kegunaannya:
/usr/X11R6	berisi X Window System, jika anda menginstallnya. X Windows merupakan sistem GUI (<i>Graphical User Interface</i>) pada sistem Linux.
/usr/bin	adalah tempat software yang nyata pada 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	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 bahasa C. 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. Petunjuk penggunaan perintah yang digunakan di linux
/usr/src	Berisi source code. 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.

Tabel 4.3. isi direktori /var

Direktori	Kegunaannya:
/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 terkoneksi ke suatu network, mail yang datang disimpan pada /var/spool/mail sampai dibacanya atau dihapusnya.

ADMINISTRASI DASAR LINUX



Modul Praktek Sistem Operasi



Setelah kuliah ini, Mahasiswa diharapkan akan dapat :

Mengenal Shell Linux

Mengenal perintah dasar di linux.

Mengenal kombinasi perintah dasar.

Topik yang akan dipelajari :

Konsole Linux.

Manajemen file dan direktori.

5.1. Konsole dan Terminal

Dalam item unix, konsole dan terminal merupakan istilah yang akan sering didengar atau digunakan. Konsole bisa dikatakan merupakan “DOS atau MS-Dos-nya” Unix atau Linux. Konsole atau “shell” merupakan antarmuka linux yang lebih tepat disebut interpreter berbasis perintah (text). Sedangkan terminal adalah perangkat keras seperti workstation atau peripheral yang terhubung dikomputer, seperti printer atau scanner, dll.

Shell adalah pengelola yang menerima input user dan memanggil perintah-perintah program. Sebuah shell akan dapat membantu kita untuk menjalankan program dengan mudah sehingga lebih mempercepat kerja, tentu saja jika kita sudah mengetahui perintah-perintah.di console atau shell tersebut, sehingga akan memudahkan kita dalam meng-konfigurasi sistem yang tidak bisa dilakukan dengan tampilan Grafis (Xwindows). Bourne Again Shell adalah interpreter perintah Unix standar original.

Untuk masuk ke console dari tampilan GUI, tekan Ctrl + Alt + F1.

5.2. Perintah Dasar Linux

Perintah-perintah dalam sistem operasi under Unix bersifat “**case sensitive**”, perintah tersebut membedakan antara penggunaan huruf besar (capital) dan huruf kecil. Artinya penulisan perintah ‘ls’ dengan ‘LS’ atau ‘Ls’ atau ‘IS’ akan diterjemahkan sebagai perintah yang berbeda.

Dalam sistem operasi under Unix banyak sekali kombinasi perintah yang bisa digabungkan dalam satu perintah artinya dalam satu perintah bisa terdiri dari beberapa perintah dasar. Apabila kita kesulitan atau bingung karena banyak kombinasi perintah, option, atau sintaks yang digunakan, kita bisa melihat howto-nya, readme, atau pada manual page-nya.

Contoh:

man ls : melihat halaman manual dari perintah ls (help)

1. **ls <option>**

fungsi : untuk melihat isi suatu direktori, contoh:

ls -l : untuk melihat file atau direktori pada direktori aktif dengan menyertakan atributnya

ls -la /etc : untuk melihat file atau direktori pada direktori /etc beserta file hiddennya





Keterangan : pada sistem operasi under Unix menggunakan '/' untuk memisahkan *parent direktory* dengan *child direktory*, sedangkan pada DOS menggunakan '\'.
Keterangan : penggunaan 'cd' tanpa nama direktori akan menuju ke home direktori, sedangkan penggunaan 'cd-' akan menuju ke direktori aktif sebelumnya.

2. **cd <option> <direktori>**

fungsi : untuk berpindah direktori (change directory), contoh:

cd /home/banisaleh : untuk berpindah ke direktori /home/banisaleh

cd .. : untuk berpindah ke parent direktori dari direktori aktif (direktori di atasnya).

Keterangan : penggunaan 'cd' tanpa nama direktori akan menuju ke home direktori, sedangkan penggunaan 'cd-' akan menuju ke direktori aktif sebelumnya.

3. **mkdir <nama_direktori>**

fungsi : untuk membuat direktori baru

mkdir banisaleh : untuk membuat direktori baru dengan nama banisaleh pada direktori aktif

Keterangan : penggunaan spasi pada nama direktori akan diasumsikan sebagai beberapa nama direktori baru.

4. **rmdir <nama_direktori>**

fungsi : untuk menghapus direktori yang sudah kosong

rmdir banisaleh : untuk menghapus direktori dengan nama banisaleh

5. **cp <target> <tujuan>**

fungsi : untuk mengkopi file atau direktori

cp so_simi /home : mengkopi file so_ti ke direktori /home

cp coba /home/kopian : apabila dalam direktori home tidak ada direktori kopian, maka file coba akan dikopi di direktori home sekaligus diganti namanya dengan nama kopian

6. **mv <target> <tujuan>**

fungsi : untuk memindahkan file atau direktori dan atau *rename*-nya

mv /home/so /etc : memindahkan file atau direktori so didirektori /home ke dalam direktori /etc dengan nama so

mv banisaleh stmik : mengubah (me-rename) file 'banisaleh' menjadi 'stmik' pada direktori aktif

7. **rm <file> atau rm -rf <nama direktori>**

fungsi : untuk menghapus file atau direktori yang berisi subdirektori & file

rm dataku : menghapus file dataku pada direktori aktif

rm -rf data : menghapus direktori data

rm -rf * : menghapus semua folder/direktori pada direktori aktif

8. **pwd**

fungsi : untuk mengetahui path direktori yang sedang aktif (print working directory)

9. **cat <nama_file>**

fungsi : untuk melihat isi file atau membuat suatu file

cat banisaleh : melihat isi file banisaleh





cat > data : membuat file baru dengan nama data (untuk mengakhirinya menggunakan ^D (Ctrl+D)) *lihat kombinasi perintah

10. find <option>

fungsi : untuk mencari file/direktori

find / -name data.txt -print : mencari file data.txt keseluruhan direktori dan menampilkannya pada layar

Keterangan : untuk mencari file atau direktori pada direktori tertentu misal /home setelah 'find' diikuti dengan nama direktorinya misal '**find /home -name...**'

11. pico <nama_file> atau vi <nama_file>

fungsi : editor file / text editor.

pico banisaleh.txt : mengedit file banisaleh.txt

12. who atau w

fungsi : untuk melihat user yang sedang aktif

13. ps <option> * melihat manajemen proses

fungsi untuk melihat proses yang sedang aktif / sedang dijalankan oleh system, contoh: ps -ax

14. kill <signal> <PID>

fungsi untuk menghentikan atau mematikan proses yang ada **kill**

-9 253 : untuk menghentikan proses dengan nomor PID 253

Keterangan : untuk <signal> bisa menggunakan nomor signalnya bisa juga menggunakan nama signalnya.

15. more <nama_file>

fungsi : untuk melihat isi file dan melihat output di layar, bila dalam satu layar tidak cukup maka akan ditampilkan secara perlayar.

Keterangan : apabila **more** dikombinasikan dengan perintah lain menggunakan pipeline maka output perintah pertama akan ditampilkan perlayar apabila outputnya tidak memenuhi layar.

16. clear

fungsi : untuk membersihkan layar dari tampilan text (Ctrl + I)

17. reboot atau CTRL+ALT+DEL

fungsi : untuk booting ulang komputer (restart)

18. halt, poweroff atau shutdown <option> atau init 0

fungsi : untuk mematikan atau men-shutdown computer

5.3. Kombinasi Perintah (Pengenalan redirection dan pipeline)

Redirection dan pipe merupakan standar dari fasilitas shell di unix.

5.3.1. Redirection





Fasilitas redirection memungkinkan kita untuk dapat menyimpan output dari sebuah proses untuk disimpan ke file lain (Output Redirection) atau sebaliknya menggunakan isi dari file sebagai input dalam suatu proses (Input redirection). Komponen-komponen dari redirection adalah `<`, `>`, `<<`, `>>`. Untuk lebih jelas nya akan di berikan contoh untuk masing-masing redirection :

Output Redirection

Kita akan menggunakan perintah `cat` yang telah tersedia secara default di shell linux.

Syntax-nya sbb :

```
$ cat > <namafile>
```

Contoh :

Membuat File dengan nama test

```
banisaleh@NOS-Lab:~$ cat > test
```

```
ini hanya sebuah test dari output redirection
hehehehehe.....
```

```
^D
```

Melihat isi file test

```
banisaleh@NOS-Lab:~$ cat test
```

```
ini hanya sebuah test dari output redirection
hehehehehe.....
```

Penambahan untuk file test

```
banisaleh@NOS-Lab:~$ cat >> test
```

```
baris ini tambahan dari yang lama...
```

```
^D
```

Melihat kembali isi file test setelah ditambahkan

```
beberapa kata banisaleh@NOS-Lab:~$ cat test
```

```
ini hanya sebuah test dari output redirection
hehehehehe.....
```

```
baris ini tambahan dari yang lama...
```

banisaleh@NOS-Lab:~\$ Disini kita melihat bahwa output dari perintah `cat` akan di belokkan (redirect) ke sebuah file dengan nama test.

Input Redirection

Isi dari suatu file akan kita gunakan sebagai input dari suatu proses yang akan kita jalankan. Sebagai contoh kita akan mencoba mengirimkan mail ke diri sendiri dengan isi mail berasal dari file test tadi

```
banisaleh@NOS-Lab:~$ mail pipiet < test
```

Sekarang kita lihat mail yang telah dikirim tadi dengan perintah sebagai berikut :

```
banisaleh@NOS-Lab:~$ mail
```

```
Mail version 8.1 6/6/93. Type ? for help.
```

```
"/var/spool/mail/pipiet": 1 message 1 new
```

```
N 1 jarkom@neilea.yahoo Thu Nov 9 14:51 14/525
```

```
& 1
```

```
Message 1:
```





```
From jarkom@neilea.yahoo.com Thu Nov 9 14:51:12 2000
Delivered-To: jarkom@neilea.yahoo.com
To: jarkom@neilea.adhyaksa.net
Date: Thu, 9 Nov 2000 14:51:12 +0700 (JAVT)
From: jarkom@neilea.yahoo.com (/me)
ini hanya sebuah test dari output
redirection hehehehehe.....
baris ini tambahan dari yang lama...
& q
banisaleh@NOS-Lab:~$
```

Isi dari file test di jadikan sebagai input dari proses yang kita lakukan yaitu proses pengiriman mail.

5.3.2. Pipeline

Pipeline (|) adalah fasilitas di shell UNIX yang berfungsi untuk memberikan input dari suatu proses dari output proses yang lain. Misalkan sebagai contoh :

Sebelum kita gunakan pipeline

```
banisaleh@NOS-Lab:~$ find *
dead.letter
mbox
test
```

Setelah kita gunakan pipeline

```
banisaleh@NOS-Lab:~$ find * | grep
test
test
banisaleh@NOS-Lab:~$
```

Pada contoh pipeline di atas, mempunyai arti bahwa output dari perintah find menjadi input dari perintah grep yang kemudian hanya mengambil kata "test" dari output find.

5.4. Command line completion (melengkapi perintah)

Misalnya anda ingin mengetikkan perintah startx, cukup mengetikkan star<tab> maka di lajur perintah akan muncul startx, anda hanya tinggal menekan<enter> untuk menjalankan perintah tersebut. Hal diatas juga berlaku pada file dan direktori. di home direktori ada file .bash_history (titik depan=tersembunyi sama dengan hidden pada dos).

Bila anda mengetikkan cat.bas<tab>(perintah cat berfungsi melihat isi file tersebut, maka bash akan mengeluarkan bunyi beep, artinya ia tidak memiliki cukup informasi untuk mencari file yang diinginkan. Bila anda menekan tab beberpa kali maka akan muncul nama file yang depannya juga bash, misalnya bash_profile, bash_logout dan lain-lain.

5.5. Command History (pengingatan perintah)

Fasilitas ini memungkinkan anda tidak perlu mengetik perintah yang sudah anda ketikkan. Bagi yang sudah mengenal DOSkey, tidak akan asing dengan perintah ini. Tekan panah atas untuk naik dalam daftar perintah yang telah anda ketik, panah bawah untuk turun.





5.7. Wildcards

Wildcards dalam bash serupa yang ada dalam DOS.

- * menyamai semua karakter berapapun jumlahnya
- ? menyamai semua karakter, satu saja
- [...] menyamai semua karakter yang terkandung dalam kurung

5.8. Alias

Alias dalam bash berguna memberikan nama lain ke satu perintah atau lebih.

Alias biasanya adalah singkatan dari perintah-perintah lain.

Misal:

```
Alias dir='ls'
```

```
Alias ren='mv'
```

```
Alias copy='cp'
```

```
Alias l='ls -l'
```

```
Alias 2='ls -AF'
```

```
Alias 3='ps ax'
```

```
Alias 4='top'
```

```
Alias vi='vim'
```

```
Alias compile.kernel='make dep; make clean; make boot; make modules; make modules_install'
```

Tanda ; bisa digunakan untuk menyatukan beberapa perintah dalam satu lajur. Supaya alias terdefinisi setiap kali anda memakai bash, anda harus menaruh dalam file inialisasi bash. Yaitu /etc/bashrc dan /etc/profile yang mempengaruhi semua sistem dan ~/.bashrc serta ~/.bash_profile yang mempengaruhi account anda saja. ~ (tilde) adalah karakter yang berarti home directory anda.

ADMINISTRASI SYSTEM LINUX





Setelah kuliah ini, Mahasiswa diharapkan akan dapat :

Mengenal manajemen user dan grup di

Linux Mengenal perijinan file dan direktori.

Mengenal manajemen proses dan disk.

Topik yang akan dipelajari :

Membuat user dan group.

Manajemen perijinan / hak akses file dan direktori. Mengamati proses sistem dan menghentikan proses. Mounting device.

6.1 Penanganan Pengguna (User dan Group)

Administrasi sistem pada LINUX hanya dapat dijalankan oleh super user atau administrator sistem tersebut.

1. User

User hanya dapat dibuat dan dihapus oleh administrator atau super user saja. Untuk membuat atau menambah user ada dua cara, yaitu dengan *shell script* (mode text) atau melalui GUI (X-Window). Untuk *shell script* dengan perintah :

useradd so : digunakan untuk membuat user **so**, atau

adduser so : untuk menambah user. bila perintah **useradd** tidak ditemukan

rmuser -r so : untuk menghapus user **so** beserta home direktorinya

Untuk GUI tergantung dari distro Linux dan desktop managernya. Biasanya ada pada menu **configuration**

2. Group

Sama halnya dengan pembuatan user, pembuatan group dapat dilakukan dengan dua cara, yaitu dengan *shell script* dan GUI. Untuk *shell script* dengan perintah :

groupadd stmik : digunakan untuk membuat group stmik, atau

addgroup stmik: untuk menambah group. bila perintah **groupadd** tidak ditemukan

groupdel/rmgroup : digunakan untuk menghapus group yang sudah ada

6.2 Batasan File (File Restriction)

Di dalam linux setiap file mempunyai batasan file. Batasan file (hak akses) meliputi hak perijinan akses (permission), kepemilikan (owner), serta group pemilik file.





Fungsinya adalah untuk mengatur serta membatasi akses-akses terhadap file yang bersangkutan. Untuk melihat hak akses file menggunakan perintah `ls -l`, sedangkan perintah untuk menentukan hak akses adalah `chmod`. Dalam batasan file ada beberapa ketentuan pengaturan untuk perijinan dan kepemilikan suatu file, yaitu :

- Setiap file selalu dimiliki oleh pemilik yang berhak mengatur hak akses terhadap file tersebut, hal ini berlaku juga untuk **root / administrator** atau **super user / substitute user**
- Hak akses file dibagi menjadi tiga :
 - read (baca) hak akses ini disimbolkan dengan huruf `'r'`,
 - write (tuliskan) hak akses ini disimbolkan dengan huruf `'w'`,
 - execute (eksekusi) hak akses ini disimbolkan dengan huruf `'x'`
- Pemilik file dibagi menjadi
 - tiga : user (pemilik file)
 - group (group dari memiliki file), group di sini merupakan group dari pemilik file.
 - other (selain user dan group), adalah **user** yang tidak berada dalam satu group dengan **user pemilik file**

contoh :

```
# ls -l /home/data
-rwxr-xr-x 1 so stmik 5 Mar 25 2004 /home/data
```

Pada kolom pertama terdapat hak perijinan akses dari file `/home/data`. File `'data'` dimiliki oleh `'so'` dan dimiliki oleh group `'stmik'`, dengan ukuran file 5 byte, serta tanggal pembuatan/editing terakhir. Perhatikan untuk kolom pertamanya `'-rwx-r-xr-x'` (dibaca dari kiri ke kanan),

```
  - R w x r - x r - x
  | | | | | | | | |
  1 2 3 4 5 6 7 8 9 10
```

Terlihat ada 10 karakter untuk hak akses. Maksud dari tampilan tersebut adalah

- Nomor 1 menunjukkan bahwa `data` merupakan file (`-` → file biasa, `d` → direktori, `l` → link)
- `rwx` (nomor 2,3,4) adalah hak akses untuk pemilik file / user (read/baca, write/tulis, execute/eksekusi)
- `r-x` (nomor 5,6,7) adalah hak akses untuk group dari pemilik file (read/baca, execute/eksekusi) **bukan** group yang memiliki file (pada contoh adalah group `'stmik'`)





- $r-x$ (nomor 8,9,10) adalah hak akses untuk other (user-user selain pemilik file dan groupnya)(read/baca, execute/eksekusi)

Pemilik file atau super user / administrator dapat mengubah hak akses terhadap file-filenya. Pengaturan hak akses terhadap suatu file dapat dilakukan dengan dua cara / metode :

6.2.1 Metode Huruf

Pada metode huruf ini sintaknya diwakili oleh huruf-huruf yang menerangkan tentang hak akses dan pemilik file tersebut. Huruf-huruf yang dipakai adalah :

r = menyimbolkan read / baca

w = menyimbolkan write / tulis

x = menyimbolkan execute / eksekusi

u = menyimbolkan user / pemilik file

g = menyimbolkan group

o = menyimbolkan other / user lain selain pemilik file dan group

Sintak :

`chmod [ugo]+/-[rwx] nama_file`

Dengan tanda '+' digunakan untuk memberi hak akses. tanda '-' digunakan untuk menghilangkan hak akses, contoh :

`chmod u+rw aku` : memberi hak akses kepada user / pemilik file dengan hak akses read / baca dan write / tulis pada file stmik.

`chmod go-w aku` : menghilangkan hak akses write / tulis untuk group dan other pada file oman

`chmod ugo+w aku` : menambahkan hak akses write / tulis untuk user, group dan other pada file oman,

6.2.2 Metode Angka

Selain menggunakan metode huruf, untuk mengatur hak akses dapat menggunakan metode angka. Angka-angka yang digunakan merupakan hasil dari konversi dari bilangan biner ke desimal, yaitu :

4 mewakili hak akses read / baca ($r--$) --> 100 (biner)
= 4

2 mewakili hak akses write / tulis ($-w-$) --> 010 = 2

1 mewakili hak akses execute/ eksekusi($--x$) --> 001
= 1





Omenghilangkan semua hak akses (---) --> 000 = 0

Untuk kombinasi dari hak akses yang ada, yaitu kombinasi antara hak akses r, w dan atau x adalah dengan cara menambahkan bilangan-bilangan yang mewakilinya, atau mengkonversinya dalam bentuk biner selanjutnya di desimalkan. Contoh kombinasi antara hak akses:

r-x → sama dengan 5 berasal dari 4(read) + 1(execute) atau dari konversi biner 101

Sintak perintah untuk dengan menggunakan metode angka adalah :

chmod A1A2A3 nama_file

dengan

A1 = angka yang mewakili hak akses user si pemilik file

A2 = angka yang mewakili hak akses group pemilik file

A3 = angka yang mewakili hak akses other

contoh :

chmod 651 saya

Hak akses untuk user adalah read dan execute (6) hak akses untuk group adalah write dan execute (5) hak akses untuk other adalah execute (1)

chmod 740 kamu

hak akses untuk user adalah read, write dan execute (7), (*full akses*).hak akses untuk group adalah write. hak akses untuk other tidak ada (0)

6.3. Kepemilikan File (File Ownership)

Hak kepemilikan dari sebuah file atau direktori dapat diubah-ubah. Baik mengubah kepemilikan dari seorang user ke user lain atau dari suatu group ke group lain. Ada yang perlu diperhatikan dalam merubah kepemilikan file:

- Hanya super user / administrator yang dapat merubah hak kepemilikan sebuah file
- Dalam hal pemindahan kepemilikan dari suatu group ke group lain yang perlu diperhatikan adalah group yang memiliki file, **bukan group dari user si pemilik file** . pada kasus dibawah ini adalah user pemilik adalah 'so' dan group pemilik adalah 'stmik'

```
-rwxr-xr-x 1 so stmik 5 Mar 25 2004 /home/data
```

Jadi pada kasus di atas bisa saja 'so' bukan merupakan anggota dari group 'stmik'. Harap anda perhatikan dan bedakan pada materi pengaturan hak akses sebelumnya.

Untuk mengubah kepemilikan file menggunakan perintah **chown** (untuk merubah kepemilikan file dari seorang user ke user lain) dan **chgrp**





(untuk merubah kepemilikan file dari satu group ke group lain), sintaknya adalah :

chown owner nama_file

Dengan owner adalah nama user lain yang akan diberi file

contoh :

chown mahasiswa data --> akan menjadikan file data menjadi milik user mahasiswa

chgrp group nama_file

Dengan group adalah nama group lain yang akan diberi file

contoh :

chgrp banisaleh data --> akan menjadikan file data menjadi milik dari group banisaleh

chown owner:group nama_file

contoh :

chown saya:banisaleh data → akan menyebabkan file data menjadi milik user saya dan juga menjadi milik dari group **banisaleh**

chown :banisaleh data → sama hasilnya dengan perintah **chgrp**

6.4. Manajemen Proses

Proses adalah sesuatu yang mendasar pada suatu sistem operasi. Sebuah proses dapat memboot komputer dan dapat pula mematakannya. Secara konteks sebuah proses dapat diumpamakan sebuah program yang sedang dijalankan. Proses-proses dapat dimulai/dijalankan, diakhiri/stop, dimatikan, ditentukan prioritasnya, dan dijadwalkan. Untuk memahaminya, sebagai contoh anda membuka program berarti menjalankan proses, anda menutup program berarti mematikan proses.

Dalam Linux/UNIX Untuk melihat proses dan penanganan proses-proses yaitu program konsol 'top' dan 'ps' dan untuk mematakannya atau memanipulasinya menggunakan perintah seperti 'kill' dan 'nice'. Semua informasi mengenai proses yang sedang berlangsung / berjalan disimpan pada direktori '/proc' yang berubah secara real time. Dengan membaca file-file yang ada di direktori '/proc' akan sangat merepotkan, ada dua perintah yang dapat melihat proses yang sedang berlangsung .

ps

Perintah ps digunakan untuk melihat proses yang sedang berlangsung, dengan menggunakan **ps** dapat dilihat informasi proses yang sedang berlangsung. contoh tampilan ps

PID	TTY	TIME	CMD
235	tty1	00:00:00	bash
780	tty1	00:00:00	ps





Tampilan di atas adalah untuk menampilkan informasi proses itu sendiri. karena menjalankan **ps** di baris perintah yang menginduk pada shell command line (program yang menerjemahkan perintah, biasanya **bash**). **ps** sendiri mempunyai banyak option contoh :

ps aux

Keterangan tentang field/kolom untuk perintah di atas

USER: adalah nama user yang menjalankan proses yang bersangkutan atau atas ijin siapa proses tersebut berjalan. sebagian proses yang dijalankan oleh root merupakan proses boot yang dijalankan oleh init (central dari semua proses, dapat dilihat dengan perintah 'pstree', init mengendalikan proses mana yang akan dijalankan dan dihentikan)

PID : adalah id dari proses

TTY: adalah terminal/konsol (teletype) dari mana program tersebut dijalankan, semua proses yang dijalankan oleh init tidak mempunyai aturan terminal ini.

STAT: status proses sat ini. keterangannya adalah

S (sleeping) → proses dalam keadaan sleep //kurang lebih

20 menit R (running) → proses sedang berjalan

D (defunct) → proses tidak dipakai, merupakan proses yang berjalan sembarangan, kill secepat mungkin

I (idle) → proses sedang tidak dijalankan/tidak dipakai

Z (zombie) → proses yang sudah mati, akan hilang saat di shutdown berikutnya, tidak berpengaruh pada sistem

T (terminate)→ proses dihentikan

TIME menunjukkan cpu-time yang dihabiskan (akumulasi waktu yang digunakan proses), bukan waktu proses dijalankan

%CPU persentasi dari waktu CPU yang sudah digunakan, untuk prosesor modern proses akan berisi nol, kecuali untuk keperluan X-Window

%MEM persentasi memori sistem yang digunakan oleh proses, nilai berhubungan dengan RSS, bukan VSZ

VSZ ukuran memori virtual proses. besar memori proses termasuk shared libraries

RSS (resident set size) ukuran sebenarnya. biasanya nilainya lebih kecil dari nilai yang diberikan 'top'.karena berbeda dalam hitungannya

CMD/COMMAND adalah nama program / perintah yang digunakan

top

Perintah **top** akan memberikan tampilan secara dinamis mengenai status proses yang sedang berlangsung / berjalan. tampilan tabel proses yang ada





akan di *update* pada interval waktu tertentu (default 5 detik). *option* penting pada '**top**' adalah **-d [detik]** untuk menentukan interval update dalam detik.

contoh :

```
top -d 2
```

ada beberapa field/kolom baru diantaranya

PRI Prioritas dari proses. waktu penghitungan maksimum dalam milidetik untuk proses ini

NI Nilai 'nice' nilai prioritas yang diberikan secara manual.

nice

Semua proses memiliki hak yang sama dalam pembagian sumber daya. namun dapat diubah dengan perintah **nice** untuk memulai proses dengan prioritas yang diberikan. sintak :

```
nice -n [value] [process]
```

keterangan :

Nilai negatif menambah prioritas hanya bisa dilakukan oleh 'root' namun nilai positif dapat dilakukan oleh siapa saja. Untuk merubah prioritasnya lagi dengan perintah **renice**.

sintak :

```
renice [prioritas] [pid]
```

kill

Kadang ada proses yang tidak diperlukan kehadirannya dalam sistem atau terlalu banyak memakan resource komputer, sehingga akan menurunkan kinerjanya. cara untuk menghentikan proses tersebut adalah dengan menggunakan perintah kill.

sintak :

```
kill -[signal] PID
```

Menghentikan proses berdasarkan nomor ID proses, signal adalah nomor signal yang dapat digunakan, selain nomor signal dapat juga dengan menggunakan nama signalnya, misal untuk nomor signal 9 dapat diganti dengan **KILL**

```
killall -[signal] nama_proses
```

Menghentikan proses berdasarkan nama prosesnya. contoh:

```
kill -9 253 → menghentikan proses dengan PID 253
```

```
kill -KILL 253 → mematikan proses dengan PID 253
```

```
killall httpd → menghentikan proses dengan nama httpd
```

Keterangan





Tanpa menyebutkan signal secara default akan diberikan signal 15 (**sigterm**) yang akan menutup program “menunjukkan jalan keluar bagi program”. 9 (**sigkill**) akan mematikan program / mengeluarkan program dari sistem secara paksa (membasmi program) beberapa signal yang digunakan adalah:

- | | | | |
|----|------|-------------|-----------------------------------|
| 1 | HUP | Hangup | : menggantungkan proses |
| 2 | INT | Interrupt | : mereboot program / proses |
| 3 | QUIT | Quit | : menutup program |
| 6 | ABRT | abort | : membatalkan proses |
| 9 | KILL | Kill | : mematikan proses (dengan paksa) |
| 14 | ALRM | Alarm clock | |
| 15 | TERM | terminate | : mengakhiri program |

6.5. Manajemen Disk

Pada sistem operasi *under* UNIX seperti LINUX tidak mengenal adanya drive **c:** atau drive **A:**, karena pada LINUX menggunakan sistem hirarki dan penyatuan (direktori di dalam direktori) dan memperlakukan file, direktori dan device driver (termasuk floppy dan cdrom) sebagai file.

Sedangkan device driver untuk disk drive ditempatkan pada direktori **/dev** (direktori tempat semua device). Dan disk drive yang belum dikenal tersebut harus dikenalkan / dimuatkan / dipasangkan pada suatu direktori agar bisa dibaca / digunakan.

Ada beberapa tipe file sistem yang sering digunakan dan perlu anda perhatikan untuk mengenalkan device yang akan digunakan :

msdos → untuk mengenalkan file-file dari DOS dan Windows

vfat → juga untuk mengenalkan file-file dari DOS dan Windows(FAT 16 dan FAT 32)

ext2 → untuk mengenalkan file-file dari

Linux **ntfs** → untuk file sistem dari NT

ufs → untuk mengenalkan file-file dari UNIX

untuk mengenalkannya menggunakan perintah **mount**.

sintak standarnya adalah :

mount -t [fstype] [device]

[dir] keterangan :

mount -t → adalah perintah untuk mengenalkan sebuah file sistem ke dalam Linux

fstype → adalah tipe dari file sistem pada device yang akan dipakai





- device** → adalah direktori tempat device driver (driver dari alat yang digunakan) berada, terletak pada direktori **/dev**
- dir** → adalah direktori yang digunakan untuk menampung file-file pada device yang akan digunakan

Setelah melakukan **mount** (apabila sudah selesai dipakai) sebaiknya device yang telah di **mount** harus dinonaktifkan (di **umount**).

sintaknya :

```
umount [device]
```

Floppy

Pada sistem operasi Linux / under UNIX media floppy disk dikenal dengan **fd0** untuk drive A (istilah pada Windows) dan **fd1** untuk driver B. Untuk mengenalkan media floppy disk pada sistem operasi Linux menggunakan perintah **mount**. contoh :

```
mount -t msdos /dev/fd0 /mnt/floppy
```

maksud dari perintah di atas adalah :

File sistem yang akan dikenalkan berasal dari DOS atau Windows, dan device/alat yang digunakan adalah floppy disk (drive A). Dan hasil dari pengenalan file sistem tersebut akan ditempatkan pada direktori **/mnt/floppy**.

Sedangkan cara memformat disket dengan kapasitas 1,44 MB menggunakan perintah:

```
fdformat /dev/fd0h1440
```

Setelah selesai proses pemformatan dilanjutkan dengan menentukan tipe file sistem yang akan digunakan, untuk tipe file sistem ext2 / linux dengan perintah

```
mkfs /dev/fd0            atau
```

```
mke2fs /dev/fd0        atau
```

```
mkfs.ext2 /dev/fd0
```

Untuk membuat tipe file sistem DOS atau Windows dengan menggunakan perintah :

```
mkfs -t msdos /dev/fd0            atau
```

```
mkdosfs /dev/fd0                    atau
```

```
mkfs.msos /dev/fd0
```





Bagaimana cara mengcopy dari disket ke disket ? Ada dua langkah untuk mengcopy disket ke disket, yaitu :

- Masukkan disket yang akan dicopy ke floppy drive, ketikkan perintah `dd if=/dev/fd0 of=/tmp/kopian`
- Ganti disketnya dengan disket kedua, kemudian ketikkan perintah `dd if=/tmp/kopian if=/dev/fd0`
atau bisa juga dengan menggunakan fasilitas cat untuk pengalihan file
- Masukkan disket pertama, ketikkan `cat /dev/fd0 > /tmp/kopian`
- Ganti disket, masukkan disket kedua, ketikkan `cat tmp/kopian > /dev/fd0`

Direktori /tmp/kopian bisa anda ganti menggunakan direktori lain asalkan output file (`of=...`) pada perintah pertama untuk cara 1 harus sama dengan input file (`if=...`) pada perintah keduanya. Begitu juga yang menggunakan perintah cat.

Harddisk

Untuk mengenalkan media harddisk prinsipnya sama dengan floppy. Untuk mengetahui partisi-partisi yang ada dalam harddisk dapat menggunakan perintah `df`. Untuk mengenalkan media harddisk

contoh :

```
mount -t vfat /dev/hda1
```

`/mnt/windows` keterangan :

File sistem yang akan dikenalkan berasal dari DOS atau Windows, dan device/alat yang digunakan adalah harddisk primary . Dan hasil dari pengenalan file sistem tersebut akan ditempatkan pada direktori `/mnt/windows`. Selain menggunakan tipe file sistem `msdos` bisa juga menggunakan tipe `vfat` untuk tipe file sistem dari Windows.

Beberapa tipe file system seperti NTFS, untuk beberapa distro terkadang tidak dapat dikenali (seperti distro Redhat dan Fedora). Agar tetap dapat dikenali, user harus menambahkan modul untuk mendeteksi file system NTFS pada distro tersebut.

Beberapa distro lain yang dapat mengenali file system NTFS terkadang hanya dapat membaca file-file dan direktori pada partisi hardisk dengan format file system NTFS, tetapi tidak dapat diedit, copy atau modifikasi lainnya.

CDROM

Untuk mengenalkan CDROM prinsip sama saja dengan dua media di atas, hanya saja untuk tipe file sistemnya tidak perlu dituliskan





contoh :

```
mount /dev/cdrom /mnt/cdrom
```

keterangan :

/cdrom adalah device untuk cdrom yang akan dikenalkan, dan
/mnt/cdrom adalah direktori tempat menampung file dari cdrom

Flash Disk

Untuk mengenalkan flashdisk prinsip sama saja dengan media CDROM di atas, untuk tipe file sistemnya tidak perlu dituliskan

contoh :

```
mount /dev/sda1
```

/mnt/flash keterangan :

/sda1 adalah device untuk flashdisk dari **usb1** yang akan dikenali oleh linux Slackware, distro lain mungkin saja berbeda, seperti **/dev/sd0** dan **/mnt/flash** adalah direktori tempat menampung file dari cdrom. Ingat direktori flash sebelumnya harus dibuat terlebih dahulu, baru proses mounting device dilakukan





INSTALASI PAKET APLIKASI di LINUX

Setelah kuliah ini, Mahasiswa diharapkan akan dapat :
Mengetahui paket instalasi program di Linux
Mengetahui perintah dan cara-cara install program.
Mengetahui konfigurasi paket program yang sederhana.

Topik yang akan dipelajari :
Instalasi paket .rpm dan tar.gz.

7.1 Pembuatan Paket Aplikasi Web Server

Server web berfungsi banyak, salah satu fungsinya adalah memberikan informasi kepada orang yang mengakses kedalamnya. Server web selain berfungsi untuk internet, juga bisa berfungsi sebagai intranet, yang akan menjembatani komunikasi antar anggota dalam jaringan tersebut. Dalam hal ini server yang di gunakan adalah server Apache. Disini akan dijelaskan bagaimana cara instalasi dan setting server web apache, yaitu dengan cara instalasi dengan menggunakan rpm dan Tar.gz.

7.1.1 Instalasi paket .rpm

Salah satu utilitas yang paling powerful dan inovatif dari RedHat Linux adalah RPM, *RedHat Package Manager*. Teknologi RPM adalah sebuah cara yang paling mudah untuk melakukan instalasi, uninstalasi, *upgrade*, *query*, *verify* dan *mem-build* paket *software*.

Paket *software* RPM adalah sebuah arsip dari file-file dengan beberapa informasi yang berhubungan dengan file-file tersebut, seperti nama, versi dan deskripsinya.

Berbeda dengan cara tradisional menggunakan file tar.gz, metode RPM membuat user tidak direpotkan masalah manajemen file seperti letak file, direktori dsb.

Untuk menjalankan RPM sangat mudah, berikut ini adalah perintah umum untuk:

Install

```
rpm -i [option] [nama_paket_rpm]
```

misalnya:

```
rpm -ivh vim-4.5-2.i386.rpm
```

v dan h adalah option untuk menampilkan jalannya instalasi dan menampilkan grafik berbentuk hash (#). Option lainnya dapat di lihat di **man rpm**.





Bila paket tersebut telah di instalasi atau ditemukan *dependency*, RPM akan memberitahukan dan instalasi di batalkan. Gunakan - -replacepks atau - -replacefiles bila ingin menimpa paket lama.

Untuk *dependency error*, paket atau file yang di dibutuhkan harus di instalasi dulu. Sebaiknya **tidak** menggunakan option - -nodeps untuk mengatasi *error dependency*.

Upgrade

```
rpm -U [option] [nama_paket_rpm]
```

misalnya :

```
rpm -Uvh emacs-19.34-4.i386.rpm
```

Upgrade ini adalah kombinasi dari uninstalasi dan instalasi, artinya RPM akan menguninstalasi dulu paket versi lama bila ditemukan dan menggantinya dengan versi baru. Bila tidak ditemukan versi lamanya maka otomatis RPM melakukan instalasi.

Uninstall

```
rpm -e [option] [nama_paket]
```

misalnya:

```
rpm -e emacs
```

Untuk uninstalasi tidak dipakai nama paket RPM tapi nama paket itu sendiri. *Dependency error* juga terjadi bila paket yang ingin di uninstalasi dibutuhkan oleh program lain. Gunakan - -nodeps bila ingin tetap menghapusnya. Untuk uninstalasi ini sebaiknya dilakukan dengan meng-uninstalasi paket-paket sesuai urutan *dependency*-nya.

Query

```
rpm -q [option] [nama_paket]
```

misal:

```
rpm -ql kernel
```

Option l menampilkan semua file yang dimiliki oleh paket kernel.

Verify

```
rpm -V [nama_paket]
```

misal:

```
rpm -V bash
```

Verify digunakan untuk mendeteksi bila terdapat masalah saat instalasi. RPM akan membandingkan paket yang telah di instalasi dengan informasi yang terdapat pada file paket RPM aslinya. Bila tidak ada kesalahan, RPM tidak menampilkan sesuatu.





Implementasi pada instalasi Apache Web Server :

```
[root@NOS-lab/tmp#rpm -ivh apache_1.3.19-i386.rpm
```

7.1.2 Instalasi paket .tar.gz

Jika mendownload versi source-codenya, pertama kali copy-kan, uncompress dan un-tar file tersebut pada suatu direktori sementara, misalnya /tmp, dengan perintah (dengan catatan anda harus login sebagai root):

1. Mengcopy file Apache :

```
root@NOS-lab#cd /tmp
```

2. Uncompress file source code : root@NOS-

```
lab/tmp]#tar xfvz apache_1.3.19.tar.gz
```

Disini anda akan didapatkan distribusi file- file source code Apache pada direktori /tmp/apache_1.3.19. Pindahlah ke direktori tersebut, dan masuk ke direktori /src yang ada didalamnya, dengan perintah :

```
root@NOS-lab #cd /tmp/apache_1.3.19
```

```
root@NOS-lab/tmp/apache_1.3.19]#cd src
```

Sangat dianjurkan untuk membaca file README dan INSTALL sebelum melakukan proses kompilasi. Pada file itu terdapat informasi umum mengenai program yang akan dikompilasi dan langkah-langkah instalasi yang diperlukan.

Konfigurasi module yang akan di install

Pilihlah module-module yang hendak disertakan pada program Apache nantinya, dengan melakukan pengeditan terhadap file configuration.

Module adalah program tambahan yang melakukan fungsi tertentu yang berguna untuk menambah fitur Apache web server. Semakin banyak module yang disertakan, maka semakin besar ukuran total space harddisk yang dibutuhkan Apache.

Jika ingin melakukan perubahan apapun terhadap module standar (default), tinggalkan saja langkah ini.

Konfigurasi Sistem Operasi

Selanjutnya harus menentukan konfigurasi sistem operasi yang digunakan:

Ketikan saja perintah: `root@NOS-`

```
lab/tmp/apache_1.3.19]#./configure
```

Langkah ini akan menghasilkan file-file makefile yang diperlukan untuk langkah selanjutnya.

Kompilasi

Langkah terakhir adalah melakukan kompilasi. Ketikkan perintah :





```
root@NOS-lab/tmp/apache_1.3.19]#make
```

Tampilan selanjutnya adalah tampilan proses kompilasi yang tidak perlu diperhatikan, selama tampilan itu tidak melaporkan adanya error. Jika ada error, maka proses akan berhenti dan memberitahukan apa penyebab kesalahan dan pada baris ke berapa. Perbaiki kesalahan ini, dan ulangi proses kompilasi sampai tidak ada kesalahan apapun.

Instalasi

Setelah proses kompilasi selesai, maka akan terdapat file httpd yang bersifat executable pada direktori src ini. File inilah yang sebenarnya disebut file program Apache web server.

Sekarang saatnya untuk melakukan proses instalasi Apache, yaitu pemasangan file-file yang diperlukan, selain file httpd, ke direktori-direktori yang benar. Ketikkan perintah berikut:

```
root@NOS-lab/tmp/apache_1.3.19]#make install
```

Perintah ini secara otomatis akan meletakkan file-file program Apache pada direktori /usr/local/apache.

File-file yang di-copy-kan ke direktori ini adalah file httpd, file-file konfigurasi yang diletakkan pada direktori conf, file-file log yang diletakkan pada direktori logs, dan file-file gambar standar yang diletakkan pada direktori icons.

Secara default, program httpd akan mencari file konfigurasi /usr/local/apache/conf/httpd.conf. jika sengaja meletakkan file-file apacheselain pada direktori /usr/local/apache, maka perlu menjalankan httpd dengan option -f, misalnya: **root@NOS-lab# ./httpd/data/httpd/conf/httpd.conf**

REFERENSI

Team SO Amikom, Modul Praktikum SO, 2004, 2005

<http://pdpt.ui.ac.id/ppkk/DOS.html>

M Taufik AR, **Windows Xp Professional**, Petunjuk Penggunaan User, 2005

http://cmis.yarsi.ac.id/manual_cmis/manual_xp.html

Tutorial Microsoft Windows, <http://www.ilmukomputer.com>, 2002

Sistem Operasi, Masyarakat Digital Gotong Royong (MDGR), 2003-2005 <http://www.ilmukomputer.com/umum/indy-linuxdesktop.php>