

# MODUL PRAKTIKUM SISTEM OPERASI

## PRAKTIKUM II

### Organisasi File Pada Sistem Operasi Linux

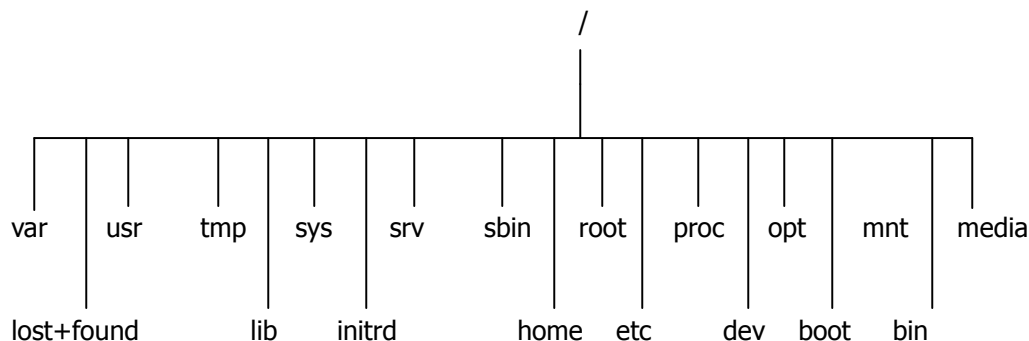
#### A) TUJUAN

1. Mengetahui organisasi File di Linux
2. Menciptakan dan manipulasi direktori
3. Mempelajari izin akses (permission) dari file dan direktori
4. Mengetahui konsep Owner dan Group
5. Mengetahui konsep Link dan symbolic link

#### B) DASAR TEORI

##### ORGANISASI FILE

Sistem file pada Linux menyerupai pepohonan (tree), yaitu dimulai dari root, kemudian direktori dan sub direktori. Sistem file pada Linux diatur secara hirarkhikal, yaitu dimulai dari root dengan symbol “/”.



Kita dapat menciptakan File dan Direktori mulai dari root ke bawah. Direktori adalah file khusus, yang berisi nama file dan INODE (pointer yang menunjuk ke data / isi file tersebut). Secara logika, Direktori dapat berisi File dan Direktori lagi (disebut juga Subdirektori).

# MODUL PRAKTIKUM SISTEM OPERASI

## PRAKTIKUM II

### TIPE FILE

Pada Linux terdapat 6 tipe file yaitu:

- Ordinary file
- Direktori
- Block Device (Peralatan I/O)

Merupakan representasi dari peralatan hardware yang menggunakan transmisi data per block (misalnya 1 KB block), seperti disk, floppy, tape.

- Character Device (Peralatan I/O)

Merupakan representasi dari peralatan hardware yang menggunakan transmisi data karakter per karakter, seperti terminal, modem, plotter dll

- Named Pipe (FIFO)

File yang digunakan secara intern oleh system operasi untuk komunikasi antar proses

- Link File

### PROPERTI FILE

File mempunyai beberapa atribut, antara lain :

Tipe file : menentukan tipe dari file, yaitu :

Karakter	Arti
-	File biasa
d	Direktori
l	Symbolic link
b	Block special file
c	Character special file
s	Socket link
p	FIFO

ijin akses : menentukan hak user terhadap file ini.

Jumlah link : jumlah link untuk file ini.

Pemilik (*Owner*) : menentukan siapa pemilik file ini

Group : menentukan group yang memiliki file ini

Jumlah karakter : menentukan ukuran file dalam byte

Waktu pembuatan : menentukan kapan file terakhir dimodifikasi

Nama file : menentukan nama file yang dimaksud

# MODUL PRAKTIKUM SISTEM OPERASI

## PRAKTIKUM II

**Contoh :**

-rw-r--r--	1	root	root	5	2011-05-26 03:36	f2	
↓	↓	↓	↓	↓	↓	↓	
tipe file	izin akses	jum link	pemilik	group	jumlah karakter	waktu	nama file

### NAMA FILE

Nama file maksimal terdiri dari 255 karakter berupa alfanumerik dan beberapa karakter spesial yaitu garis bawah, titik, koma, dan lainnya kecuali spasi dan karakter "&", ";", "|", "?", "`", "'", "\"", "[", "]", "(", ")", "\$", "<", ">", "{", "}", "^", "#", "\", "/".

Linux membedakan huruf kecil dengan huruf besar (case sensitive). Contoh nama file yang benar :

```
Abcde5434
3
prog.txt
PROG.txt
Prog.txt,old
report_101,v2.0.1
5-01.web.html
```

### IJIN AKSES

Setiap obyek pada Linux harus mempunyai pemilik, yaitu nama pemakai Linux (*account*) yang terdaftar pada `/etc/passwd`.

Ijin akses dibagi menjadi 3 peran yaitu :

- Pemilik (**Owner**)
- Kelompok (**Group**)
- Lainnya (**Others**)

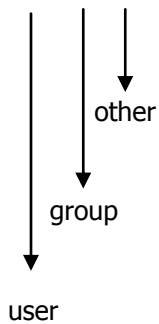
# MODUL PRAKTIKUM SISTEM OPERASI

## PRAKTIKUM II

Setiap peran dapat melakukan 3 bentuk operasi yaitu :

- Pada File :
  - R (Read) : Ijin untuk membaca
  - W (Write) : Ijin untuk mengubah / membuat
  - X (Execute) : Ijin untuk menjalankan program
- Pada Direktori :
  - R (Read) : Ijin untuk membaca daftar file dalam direktori
  - W (Write) : Ijin untuk mengubah/membuat file di direktori
  - X (Execute) : Ijin untuk masuk ke direktori (cd)

**-rw-r--r-- 1 root root 5 2008-05-26 03:36 f2**



# MODUL PRAKTIKUM SISTEM OPERASI

## PRAKTIKUM II

Format perintah untuk mengubah ijin akses:

```
chmod [ugoa] [= + -] [rwx] File(s)
```

```
chmod [ugoa] [= + -] [rwx] Dir(s)
```

keterangan:

u = user (pemilik)

g = group (kelompok)

o = others (lainnya)

a =all

Format lain kode ijin akses (chmod) adalah menggunakan bilangan **octal** (3 bit biner) sebagai berikut:

r      w      x  
**4      2      1      =      7**

Chmod			
Permission	Owner	Group	Other
Read	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Write	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Execute	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

### UMASK (User file-creation mode mask)

Untuk mengatur ijin akses ketika suatu file atau direktori akan dibuat dapat menggunakan perintah **umask**. Angka-angka di umask ini memiliki fungsi terbalik dari angka-angka chmod. Untuk menghitungnya dapat dilakukan kalkulasi sebagai berikut :

<b>Kreasi file (biasa)</b>	<b>: 666</b>
Nilai umask	: 022
	----- -
Nilai ijin akses	644

<b>Kreasi direktori</b>	<b>: 777</b>
Nilai umask	: 022
	----- -
Nilai ijin akses	755

# MODUL PRAKTIKUM SISTEM OPERASI

## PRAKTIKUM II

### LINK/ SHORTCUT

Link adalah sebuah teknik untuk memberikan lebih dari satu nama file dengan data yang sama (semacam *shortcut* pada sistem operasi windows). Terdapat dua tipe link pada linux, yaitu **hard-link** dan **soft-link** (symbolic link).

Perlu diketahui bahwa setiap file di Linux memiliki apa yang disebut **inode**. Inode ini adalah suatu struktur yang menyimpan informasi-informasi dasar mengenai isi dari suatu file, antara lain ukuran file, waktu modifikasi terakhir, mode akses, serta lokasi file tersebut di dalam harddisk. Inode inilah yang dituju oleh nama file yang ada di direktori-direktori Linux. Setiap inode juga memiliki nomor yang unik di dalam sistem. Gunakan parameter `-li` untuk melihat nomor inode tersebut, berikut contoh perintah nya : `$ ls -li`

```
[root /]# ls -li /etc/home
total 2420
17612053 drwx----- 2 root    root      4096 Mar 31 2011 MySQL-install
17612054 drwx--x--x 12 aalfcom  aalfcom  4096 Dec 31 19:09 aalfcom
17630722 drwx--x--x 17 accuread  accuread  4096 Dec 23 05:36 accuread
18767150 drwx--x--x 18 aloiinf   aloiinf   4096 Dec 31 18:31 aloiinf
16262832 drwx--x--x 11 auslock2  auslock2  4096 Dec 23 05:36 auslock2
```

### Hard LINK

Ketika suatu file dibuat, nama file yang diberikan akan secara otomatis dihubungkan secara hardlink ke inode baru, yang kemudian menunjuk ke isi file yang terletak di harddisk. Jadi setiap file yang bisa diakses di linux pasti memiliki minimal satu buah hardlink, yaitu hard link default ketika file tersebut dibuat. Namun kita bisa juga menambahkan hardlink ke suatu inode. Berikut adalah format untuk membuat suatu hard link:

```
ln namafile_asli namafile_hardlink
```

Pada hard link, bila file asli dihapus maka file hardlink (duplikat) tidak hilang. Namun bila semua file yang merujuk ke satu inode dihapus, maka serta merta inode tersebut juga akan dilupakan oleh linux karena dianggap sudah tidak digunakan lagi.

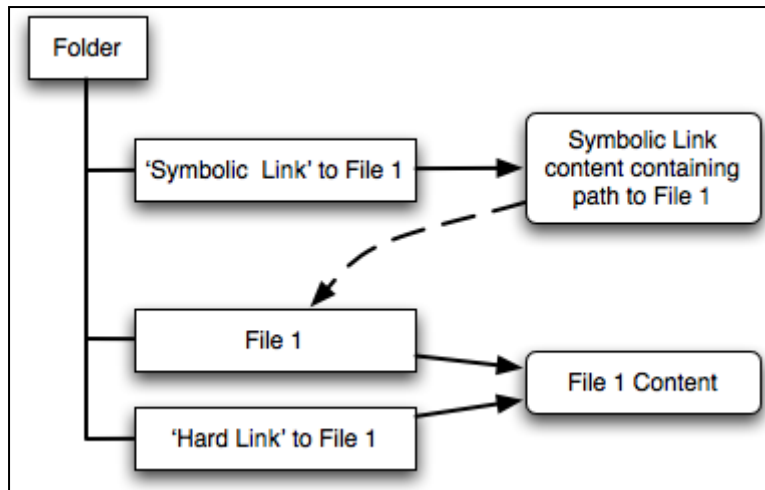
# MODUL PRAKTIKUM SISTEM OPERASI

## PRAKTIKUM II

### Soft Link

Bagi yang sudah terbiasa dengan windows, softlink sangat serupa dengan prinsip suatu shortcut. Perbedaan antara hardlink dengan softlink adalah bahwa softlink merujuk pada nama file asli ( bukan inode, dan juga bukan isi file di harddisk). Dengan memerintahkan membuat suatu softlink berarti meminta linux untuk membuatkan suatu inode baru yang merujuk ke isi file baru (yang menunjuk ke file asli). Sehingga bila file asli dihapus, maka data yang baru juga terhapus. Format membuat soft Link adalah sebagai berikut:

```
ln -s namafile_Asli namafile_softlink
```



# MODUL PRAKTIKUM SISTEM OPERASI

## PRAKTIKUM II

### C) LANGKAH – LANGKAH PRAKTIKUM

#### Percobaan 1 : Ijin Akses

1. Melihat identitas diri melalui `etc/passwd` atau `etc/group`

```
$id
$grep [nama_user] /etc/passwd
$grep [nomor_group_id] /etc/group
```

2. Memeriksa direktori home

```
$ls -ld /home/nama_user
```

3. Mengubah Ijin akses (`chmod`)

```
$touch f1 f2 f3
$ls -l
(catat hasil yang ditampilkan)

$chmod u+x f1
$chmod g=w f1
$chmod o-r f1
$ls -l
(catat ijin akses file f1)

$chmod a=x f2
$chmod u+x,g-r,o=w f3
$ls -l
(catat ijin akses file f2 dan f3)

$chmod 751 f1
$chmod 624 f2
$chmod 430 f3
$ls -l
(catat ijin akses f1, f2, f3)
```

4. Menguji ijin akses (`chmod`)

```
$echo "belajar chmod" >> f1
$echo "belajar chmod" >> f2
$echo "belajar chmod" >> f3
1# Tulis kesimpulan anda pada form praktikum.
```

5. Untuk mengganti kepemilikan digunakan perintah `chown`. Beberapa hal yang perlu diperhatikan adalah bahwa hanya root yang dapat menggunakan perintah `chown`, account biasa tidak diperbolehkan meskipun ia adalah user owner file yang ingin diubah.

```
$ls -l
$chown [user_Owner_Baru] f1
$chown [user_Owner_Baru:grup_Owner_Baru] f2
$chown [user_Owner_Baru:grup_Owner_Baru] f1 f3
$ls -l
```

2# Tulis kesimpulan anda pada form praktikum.



# MODUL PRAKTIKUM SISTEM OPERASI

## PRAKTIKUM II

6. Mengubah group hanya dapat dilakukan oleh pemilik file/direktori tersebut dengan perintah `chgrp`

Melihat nama grup yang terdaftar:

```
$grep [nama_group_yang_dicari] /etc/group
```

```
#chgrp [group_owner_baru] fl
```

```
$ls -l
```

### **Percobaan 2 : Umask (*User file-creation mode mask*)**

1. Menentukan ijin akses awal pada saat file atau direktori dibuat

```
$touch myfile
```

```
$ls -l myfile
```

2. Melihat nilai umask

```
$umask atau $umask -S
```

3. Modifikasi nilai umask

```
$umask 027
```

```
$umask
```

```
$touch newfile
```

```
$mkdir mydir
```

```
$ls -l
```

**(bandingkan ijin akses file myfile dengan newfile)**

```
$umask 077
```

```
$touch xfile
```

```
$mkdir xdir
```

```
$ls -l
```

**(analisa nilai ijin akses file xfile dan direktori xdir)**

**3# Tulis kesimpulan anda pada form praktikum.**

# MODUL PRAKTIKUM SISTEM OPERASI

## PRAKTIKUM II

### Percobaan 3 : Link

#### 1. Hard link

```
$echo "Hallo apa kabar" > halo.txt
$ls -l
$ln halo.txt hardlink
$ls -l
$cat hardlink
$mkdir newdir
$ln hardlink newdir/halo.juga
$cat newdir/halo.juga
$ls -l newdir/halo.juga
```

**4# Tulis analisa anda pada form praktikum.**

#### 2. Soft Link (Symbolic link)

```
$ln -s halo.txt softlink
$ls -l
$cat softlink
```

**5# Tulis analisa anda pada form praktikum.**

#### 3. Menghapus link

```
$rm halo.txt
$cat hardlink
$cat softlink
```

**6# Tulis kesimpulan anda pada form praktikum.**