

BAB10: FILESYSTEM, DISK & DEVICE

Oleh: Flory Katriena

::: Block & Character Device

Ketika pertama kali bisa login ke sistem Linux, saya sedikit bingung karena tidak menemukan drive A:, B: atau hard disk C: maupun D:. Mungkin bagi newbie yang lain hal membingungkan ini bisa saja terjadi. Bagi mereka yang telah lama mengenal sistem operasi DOS /Windows ini merupakan sesuatu yang sungguh berbeda, di lingkungan Linux dan juga UNIX (Syafudin, haydin@arjuna.csc.ui.ac.id), semua *device* atau *hardware device* di komputer kita akan dianggap sebagai file. Jadi itulah yang menyebabkan drive-drive tersebut tidak tampak. File-file khusus yang merupakan simbol untuk semua *device* tersebut disimpan dalam direktori /dev. *Device-device* tersebut dapat berupa **block device** ataupun **character device**.

Character device itu serangkaian karakter yang dibaca berurut misalnya serangkaian karakter yang kita tekan dari keyboard atau serangkaian data *byte* yang dikirim berurutan melalui *serial line*. Sedangkan *block device* atau sering disebut *random access device* adalah alat yang dapat menyimpan data dan menawarkan akses keseluruhan bagian secara acak. Contoh *block device* adalah hard disk dan disket. Dengan menggunakan hard disk kita dapat mengakses data yang tersimpan secara acak sedangkan bila menggunakan *serial line* misalnya, kita hanya dapat mengakses data sesuai urutannya.

Masing-masing *device node* memiliki **major** dan **minor number device**. *Major number* memberitahu kernel, *device* apa yang mewakili *node* tersebut. Misalnya *major number* 3 adalah IDE *hard disk* dan *major number* 8 adalah SCSI *disk*. *Minor number* mewakili nomor partisi, *minor number* 0 adalah keseluruhan *hard disk* yang pertama dan *minor number* 1 hingga 15 adalah partisi yang mungkin bisa dibuat di *hard disk* tersebut. Contohnya, saya memiliki 2 buah IDE *hard disk* maka *hard disk* pertama adalah /dev/hda dan hard disk kedua adalah /dev/hdb. Jika masing-masing *hard disk* tersebut memiliki 2 buah partisi maka saya memiliki /dev/hda1, /dev/hda2, /dev/hdb1 dan /dev/hdb2.

Di bawah ini isi dari direktori /dev di komputer saya. Perhatikan huruf **b** di bagian paling kiri, menunjukkan bahwa itu adalah *block device* sedangkan *character device* akan ditunjukkan dengan huruf **c**.

```
[zakaria@linux zakaria]$ ls -l --sort=none /dev/hd[ab]{?,??}
brw-rw---- 1 root    disk      3,    1 May  6 1998 /dev/hda1
brw-rw---- 1 root    disk      3,    2 May  6 1998 /dev/hda2
brw-rw---- 1 root    disk      3,    3 May  6 1998 /dev/hda3
brw-rw---- 1 root    disk      3,    4 May  6 1998 /dev/hda4
brw-rw---- 1 root    disk      3,    5 May  6 1998 /dev/hda5
brw-rw---- 1 root    disk      3,    6 May  6 1998 /dev/hda6
brw-rw---- 1 root    disk      3,    7 May  6 1998 /dev/hda7
brw-rw---- 1 root    disk      3,    8 May  6 1998 /dev/hda8
brw-rw---- 1 root    disk      3,    9 May  6 1998 /dev/hda9
brw-rw---- 1 root    disk      3,   65 May  6 1998 /dev/hdb1
brw-rw---- 1 root    disk      3,   66 May  6 1998 /dev/hdb2
brw-rw---- 1 root    disk      3,   67 May  6 1998 /dev/hdb3
brw-rw---- 1 root    disk      3,   68 May  6 1998 /dev/hdb4
brw-rw---- 1 root    disk      3,   69 May  6 1998 /dev/hdb5
brw-rw---- 1 root    disk      3,   70 May  6 1998 /dev/hdb6
brw-rw---- 1 root    disk      3,   71 May  6 1998 /dev/hdb7
brw-rw---- 1 root    disk      3,   72 May  6 1998 /dev/hdb8
brw-rw---- 1 root    disk      3,   73 May  6 1998 /dev/hdb9
brw-rw---- 1 root    disk      3,   10 May  6 1998 /dev/hda10
```

::: Filesystem

Untuk mengorganisasi file-file pada *device* diperlukan suatu metode yang disebut dengan **filesystem**. Jika Anda mengenal **FAT** selama ini di sistem operasi Windows maka Anda akan mengenal beberapa metode *filesystem* di Linux, seperti **ext fs**, **ext2 fs** atau **xia fs** dll. Saat ini *ext2 fs* adalah *filesystem* yang banyak digunakan untuk Linux karena terkenal sangat **efisien**. Meskipun demikian Red Hat Linux tetap menyediakan dukungan terhadap filesystem lain seperti msdos yang sudah *built in* di kernel atau dalam bentuk modul seperti *vfat* (Windows95 *native fs*), *ext*, *umsdos* dan sebagainya.

::: Mount & Umount Command

Untuk menggunakan *filesystem* tersebut kita lebih dahulu harus me-*mount* sebuah *block device* yang memiliki *filesystem*. Perintahnya adalah sebagai berikut:

```
# mount [-t] [-o] device mount_point
```

device berupa *block device*, *mount_point* berupa sebuah direktori untuk menampilkan *filesystem*. *t* adalah *type* atau jenis *filesystem* dan *o* adalah *option*, keduanya boleh saja tidak disertakan bila Anda sudah mengkonfigurasi file */etc/fstab* yang berisi keterangan detail mengenai *device*, jenis *filesystem*, *mount point* yang digunakan dan sebagainya.

Misalnya saya akan mengakses sebuah file di disket di drive A: maka pertama kali saya harus me-*mount* dulu disk tersebut ke sebuah direktori yang sudah saya buat misalnya */mnt/floppy*:

```
# mount /dev/fd1 -t vfat /mnt/floppy
mount: block device /dev/fd1 is write-protected, mounting read-
only
```

Setelah perintah itu barulah saya bisa membaca disket di drive A: tersebut di direktori */mnt/floppy*. Misalnya dengan mengetikkan perintah *ls* maka akan ditampilkan isi disket di drive A:

```
# ls /mnt/floppy
```

Untuk membatalkan perintah *mount* digunakan perintah **umount**

```
# umount /mnt/floppy
```

Setelah perintah tersebut dieksekusi, otomatis drive A: tidak bisa digunakan, cobalah dengan mengetikkan perintah *ls* lagi. Pesan kesalahan akan ditampilkan seperti dibawah ini.

```
# ls /mnt/floppy
filesystem not mounted
```

::: mtools

Bila Anda memiliki *filesystem DOS* dan ingin menggunakannya tanpa harus melakukan *mount* maka gunakan **mtools**. Dengan cara ini Anda tidak perlu lagi melakukan *mount* bila ingin mengakses *filesystem DOS* dan partisi yang belum diformat sekalipun. (Syafrudin, haydin@arjuna.csc.ui.ac.id). Setelah menginstalasi *mtools* tersebut, Anda cukup menjalankan perintah-perintah seperti di DOS untuk mengakses *filesystem DOS* tersebut, misalnya **mdir**, **mcopy** dan sebagainya.

::: Filesystem Manager

Pada saat *boot*, kernel akan me-*mount* *root filesystem* dari *device* yang telah ditentukan dalam LILO. Selama proses *boot* tersebut semua *filesystem* yang ada dalam tabel *filesystem* atau dalam file **/etc/fstab** juga akan diperiksa apakah perlu di *mount* atau tidak.

Red Hat telah menyediakan program yang sangat mudah digunakan untuk mengkonfigurasi *filesystem* tersebut. Program ini berjalan di X dan dapat dipanggil melalui **control panel** atau cukup mengeksekusinya dengan perintah **fstool**. Selanjutnya hasil konfigurasinya juga akan disimpan dalam file */etc/fstab*.

Bila Anda tidak menginstalasi X dan tidak bisa menjalankan **fstool**, jangan kuatir file */etc/fstab* juga dapat di edit secara manual dengan program teks editor favorit Anda.

Filesystem manager menampilkan informasi mengenai nama *device*, *mount point*, jenis *filesystem*, ukuran dan sisa spasi yang masih tersedia. (Informasi ini seperti ini juga bisa diperoleh dengan perintah **df**).

Filesystem dapat di *mount* atau di *unmount* dengan tombol **Mount** dan **Unmount**. Tanda asterisk (*) didepan jenis *filesystem* menandakan *filesystem* tersebut sudah atau sedang di *mount*.

Tombol **Format** hanya berfungsi untuk *partisi hard disk*. Tombol ini akan menjalankan program **mkfs** yaitu program untuk membuat *filesystem*.

Tombol **Check** hanya berfungsi untuk *filesystem ext2* dan *minix*. Tombol ini akan menjalankan program **fsck** dan membutuhkan waktu beberapa saat untuk menampilkan hasil pemeriksaan. Bila *filesystem* digunakan secara normal dan *fsck* selalu dijalankan pada saat *boot* maka hasil *check* seringkali tidak menunjukkan kesalahan apapun.

::: Mengedit file /etc/fstab

File */etc/fstab* berupa file teks biasa, bisa dibaca dan mudah di edit dengan program editor teks favorit kita. Lakukan secara hati-hati karena kesalahan dalam memasukkan atau menghapus sesuatu menyebabkan sistim tidak bisa me-*mount* *filesystem* saat *boot* nanti.

Mirip dengan tampilan program *fstool* diatas, saat dibuka file */etc/fstab* menampilkan kolom-kolom yang berisi informasi nama *device*, *mount point*, jenis *filesystem*, option, dump dan fsckorder. Kolom-kolom tersebut masing-masing dipisahkan dengan tabs atau spasi.

Kolom dump berhubungan dengan program **dump**, lihat manual page program **dumb**. Kolom terakhir fsckorder berhubungan dengan program **fsck**. Urutan pengecekan *filesystem* saat *boot* ditunjukkan oleh nomor dikolom ini. Pengecekan *filesystem* dapat dilakukan secara paralel

dengan memberikan nomor urut yang sama tapi *filesystem root* selalu harus nomor 1. *Filesystem* dengan *fsckorder 0* tidak akan dicek saat *boot*.

::: Filesystem Support

Linux memiliki dukungan terhadap beberapa *filesystem* lain sehingga kita dapat menggunakan atau mengakses *filesystem* yang berbeda tanpa harus melakukan konversi lebih dulu. Berikut ini adalah beberapa *filesystem* yang bisa di dukung Linux sejak kernel 2.0.30 di keluarkan: (Anda dapat mengkonfigurasi dukungan *filesystem* ini saat konfigurasi kernel)

Minix	Merupakan <i>filesystem</i> Linux yang pertama dan saat ini masih banyak digunakan untuk <i>boot disk</i> dan beberapa format disket.
Extended fs	Tidak banyak yang menggunakannya lagi dan sebaiknya tidak perlu di kompilasi dalam kernel.
Second Extended fs	Saat ini merupakan <i>filesystem</i> default untuk Linux dan seharusnya di kompilasi dalam kernel karena <i>filesystem root</i> tidak bisa bekerja bila berupa modul.
xiafs filesystem	Diperkenalkan bersamaan dengan second extended fs dan dimaksudkan untuk menggantikan extended fs. Saat ini jarang digunakan dan sebaiknya tidak di kompilasi dalam kernel kecuali Anda membutuhkannya.
DOS FAT fs	Pada dasarnya bukan merupakan sebuah <i>filesystem</i> tapi merupakan dasar bagi <i>filesystem</i> berbasis FAT lainnya seperti MS-DOS FAT, VFAT (Windows95) atau umsdos.
MS-DOS FAT fs	Jika Anda menginginkan Linux dapat mengakses sistem berbasis DOS maka Anda dapat mengkompilasi dalam kernel. Sangat berguna untuk komputer yang memiliki sistem <i>dual-boot</i> .
VFAT (Windows95) fs	Merupakan peningkatan dari MS-DOS FAT fs dan mendukung format <i>long filename</i> . Ini juga berguna untuk komputer yang memiliki sistem <i>dual-boot</i> .
umsdos	Dukungan ini dibutuhkan bila ingin menjalankan Linux diatas partisi DOS tapi sepertinya RedHat Linux tidak akan bisa berjalan pada sebuah <i>filesystem</i> umsdos.

/proc	<i>Filesystem</i> ini dipakai oleh kernel untuk menyediakan informasi mengenai sistem kepada <i>user</i> program, seperti ps , top , xload , free atau netstat . <i>Filesystem</i> ini memang seharusnya selalu ada dan walaupun ada isinya tapi tidak akan mengisi spasi hard disk.
NFS	Dibutuhkan bila Anda akan mengakses <i>remote filesystem</i> dan untuk sebuah <i>server</i> dukungan ini merupakan suatu keharusan.
SMB	Sangat berguna bila Anda menginginkan memiliki akses langsung ke Windows95 atau NT.
NCP	Dukungan terhadap NetWare melalui protokol NCP.
ISO9660	Bila Anda memiliki CD-ROM sebaiknya memanfaatkan dukungan ini karena banyak sekali CD-ROM yang ada dipasaran menggunakan format ISO9660 ini.
OS/2 HPFS	Mendukung <i>filesystem</i> OS/2 dan HPFS. Linux hanya dapat membaca <i>filesystem</i> tersebut (<i>read-only</i>).
System V & Coherent	Merupakan <i>filesystem</i> UNIX generasi lama dan dukungan ini hanya diperlukan bila Anda memiliki data-data lama dan ingin memindahkannya ke <i>filesystem</i> yang baru.
Amiga FFS	Masih berupa <i>experimental code</i> dan sebaiknya gunakan dengan hati-hati.
UFS	<i>Filesystem</i> ini digunakan oleh beberapa sistem UNIX, Solaris dan SunOS(4.2). Linux hanya dapat membaca <i>filesystem</i> ini.

Sumber dari www.linux.or.id