

# Thread



## Minggu ke 5



**Zaid Romegar Mair, ST., M.Cs**

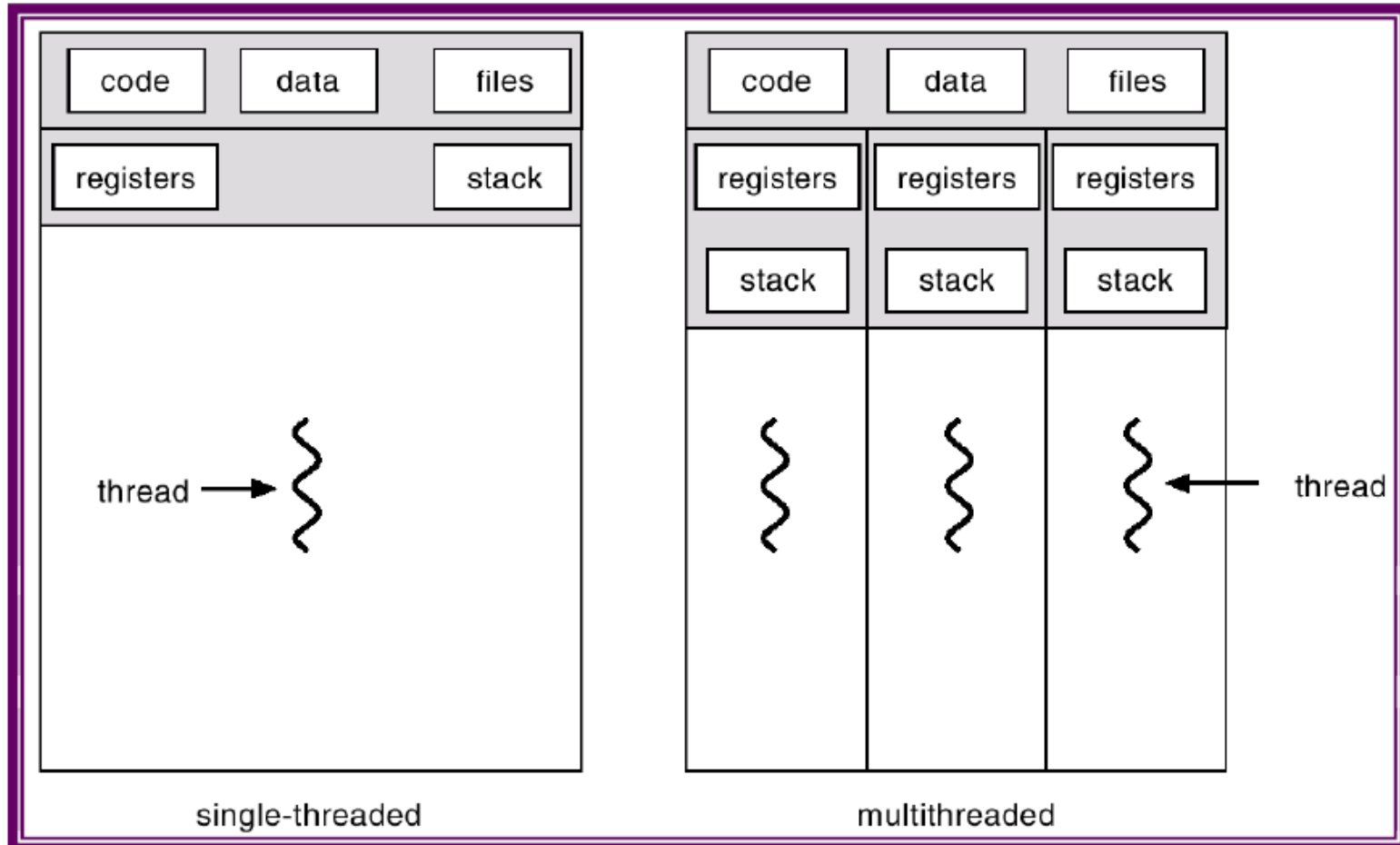
PROGRAM STUDI TEKNIK INFORMATIKA  
Jl. Kolonel Wahid Udin Lk. I Kel. Kayuara, Sekayu 30711  
web:[www.polsky.ac.id](http://www.polsky.ac.id) mail: [polsky@polsky.ac.id](mailto:polsky@polsky.ac.id)  
Tel. / Fax.: +62 714 321099

# Thread



- Suatu unit dasar dari CPU utilization yang berisi program counter, kumpulan register, dan ruang stack disebut dengan thread atau lightweight process(LWP).
- Thread akan bekerjasama dengan thread yang lainnya dalam hal penggunaan bagian kode, bagian data, dan resource sistem operasi, seperti open file dan sinyal secara kolektif yang sering disebut dengan task
- Pada multi proses, setiap proses secara terpisah melakukan operasi tidak bekerja sama dengan proses lain, setiap proses mempunyai program counter, stack register dan ruang alamat sendiri.
- Satu proses multi thread menggunakan resource lebih sedikit daripada multi proses, termasuk memory, open file dan penjadwalan CPU.
- Thread memiliki status: ready, blocked, running dan terminated, dan hanya satu thread yang aktif dalam satu waktu. Thread dapat membuat child thread. Jika satu thread dalam keadaan blocked, maka thread yang lainnya dapat dijalankan. Namun, tidak saling bebas, Sebab semua thread dapat mengakses setiap alamat dalam satu task, thread dapat membaca dan menulisi stack dari thread yang lainnya.

# Single Thread dan Multithread



# Keuntungan Thread



- **Respons/Tanggapan** yaitu Multithreading mengizinkan program untuk berjalan terus walaupun pada bagian program tersebut di block atau sedang menjalankan operasi yang lama.
  - contoh, multithread web browser dapat mengizinkan pengguna berinteraksi dengan suatu thread ketika suatu gambar sedang diload oleh thread yang lain.
- **Pembagian resource** yaitu Secara default, thread membagi memori dan sumber daya dari proses. Keuntungan dari pembagian kode adalah aplikasi mempunyai perbedaan aktifitas thread dengan alokasi memori yang sama.
- **Ekonomis** Mengalokasikan memori dan sumber daya untuk membuat proses adalah sangat mahal. Alternatifnya, karena thread membagi sumber daya dari proses, ini lebih ekonomis untuk membuat Threads.
- **Pemberdayaan arsitektur multiprosesor** Keuntungann dari multithreading dapat ditingkatkan dengan arsitektur multiprosesor, dimana setiap thread dapat jalan secara parallel pada prosesor yang berbeda. Begitu juga dengan arsitektur prosesor tunggal.

# User Thread dan Kernel Thread



- Thread terdiri dari dua bentuk yaitu user thread dan kernel thread.
- User thread adalah thread yang diatur dengan menggunakan pustaka user level thread. User thread didukung oleh kernel dan diimplementasikan oleh thread library ditingkat pengguna. Library mendukung untuk pembentukan thread, penjadwalan, dan manajemen yang tidak didukung oleh kernel.
- Kernel thread adalah thread yang didukung oleh Kernel. Contoh sistem yang menggunakan kernel thread adalah Windows 95/98/NT/2000, Solaris, Tru64 UNIX, BeOS dan Linux. Kernel Thread didukung secara langsung oleh sistem operasi: pembentukan thread, penjadwalan, dan manajemen dilakukan oleh kernel dalam ruang kernel.

# Model Multithreading

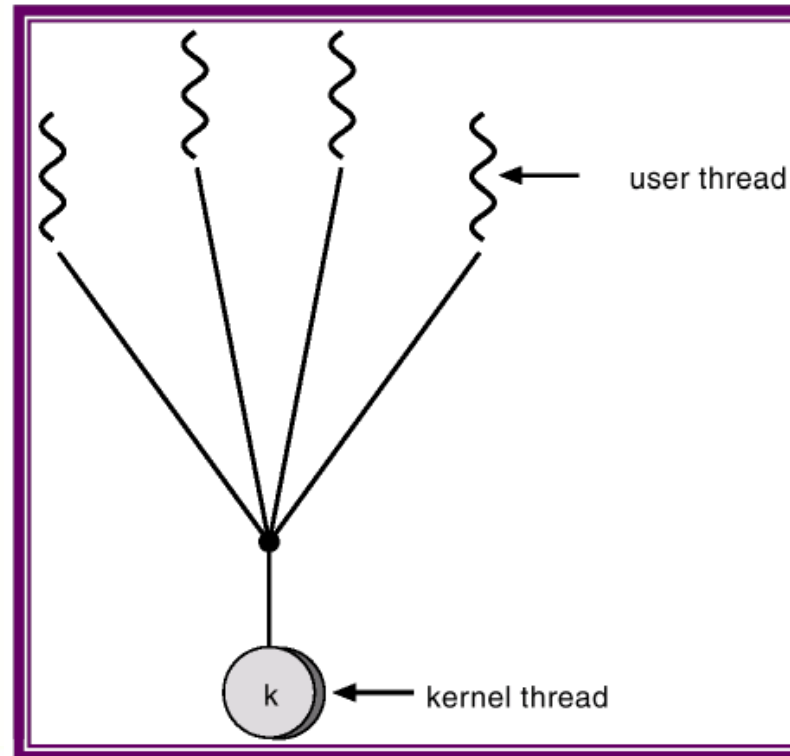


- Many to one
- One to one
- Many to many

# Many to One



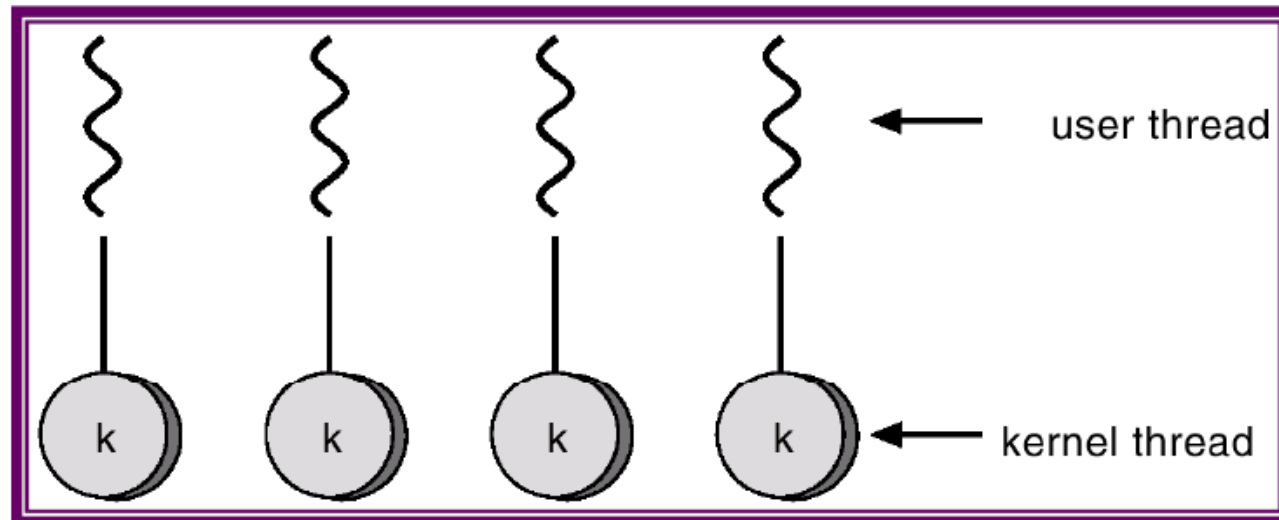
- beberapa user level thread dipetakan ke satu kernel thread dan digunakan pada sistem yang tidak mendukung kernel threads



# One-to-One



- One-to-One, setiap user-level thread dipetakan ke kernel thread
- Contoh : Windows 95/98/NT/2000 dan OS/2.

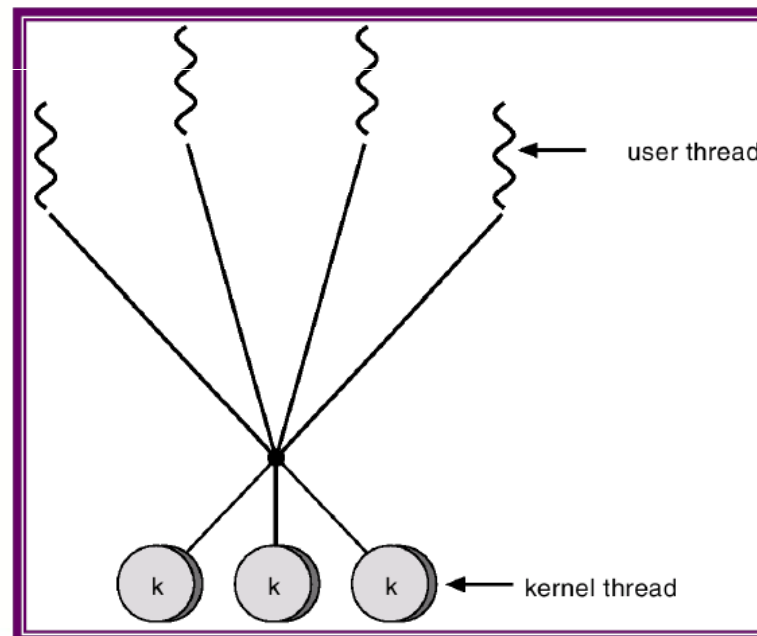




# Many-to-Many



- Many-to-Many, user level thread dipetakan ke beberapa kernel threads
- contohnya Solaris 2 dan Windows NT/2000 dengan *ThreadFiber package*

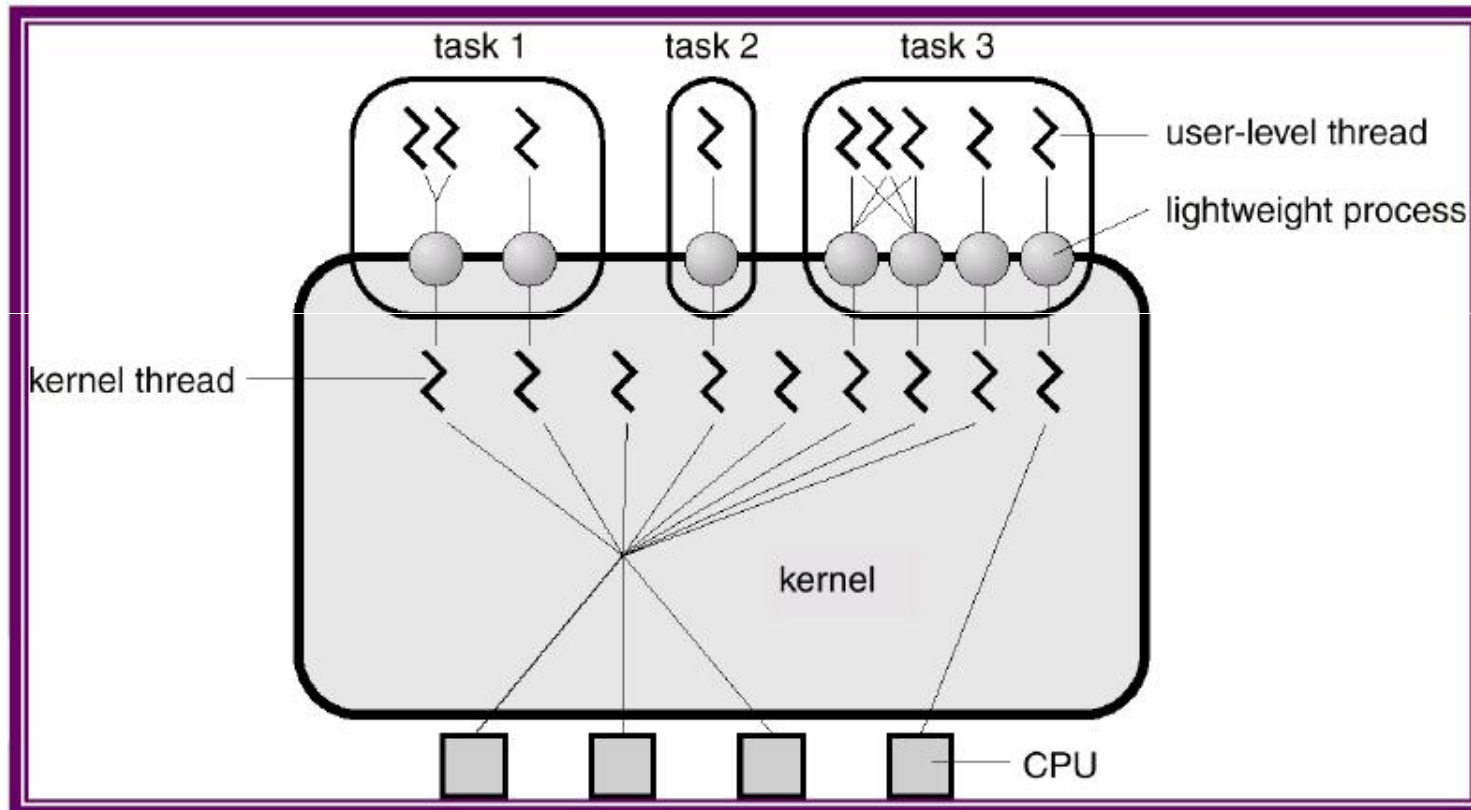


## Thread pada Solaris 2



- Solaris 2 mendukung user-level thread dan kernel thread.
- Pembuatan dan penjadwalan user level thread didukung oleh pustaka/library. kernel tidak mempunyai pengetahuan tentang user level thread. Antara user level thread dan kernel thread terdapat perantara yang disebut dengan lightweight process (LWP). Setiap task terdapat setidaknya satu LWP .
- Semua operasi pada kernel dieksekusi oleh kernel level thread. Terdapat satu kernel level thread untuk setiap LWP dan terdapat beberapa kernel level thread yang menjalankan kernel baik yang dihubungkan maupun tidak dengan LWP.

# Thread pada Solaris 2



# Thread pada Solaris 2



- Kernel thread mempunyai struktur data yang kecil dan sebuah stack.
- Perpindahan antar kernel thread tidak mengubah informasi akses memori sehingga relatif cepat. LWP berisi process control block dengan register data, informasi akutansi dan informasi memori. Perpindahan antar LWP membutuhkan tambahan pekerjaan dan relatif lambat.
- User level thread hanya memerlukan sebuah stack dan sebuah program counter, tanpa resource dari kernel. Kernel tidak dilibatkan dalam penjadwalan user level thread, sehingga perpindahan antar user level thread sangat cepat. Terdapat ratusan user level thread, tetapi semua kernel yang terlihat dalam bentuk LWP yang mendukung user-level thread.

# Thread pada Windows 2000



- Thread pada Windows 2000 mengimplementasikan pemetaan one-to-one. Setiap thread terdiri dari thread id, kumpulan register, stack user dan kernel yang terpisah serta ruang data privat.

# Thread pada Linux



- Thread pada Linux sering disebut sebagai task daripada thread.
- Pembuatan thread dilakukan dengan menggunakan system call `clone()`. `Clone()` memungkinkan task anak menggunakan ruang alamat dari task (proses) parent.

# **Praktikum Sistem Operasi Operasi File dan Struktur Direktory**



**Minggu ke 5**



**Zaid Romegar Mair, ST., M.Cs**  
**NUPN : 9902702271**

PROGRAM STUDI TEKNIK INFORMATIKA  
Jl. Kolonel Wahid Udin Lk. I Kel. Kayuara, Sekayu 30711  
web:[www.polsky.ac.id](http://www.polsky.ac.id) mail: [polsky@polsky.ac.id](mailto:polsky@polsky.ac.id)  
Tel. / Fax.: +62 714 321099

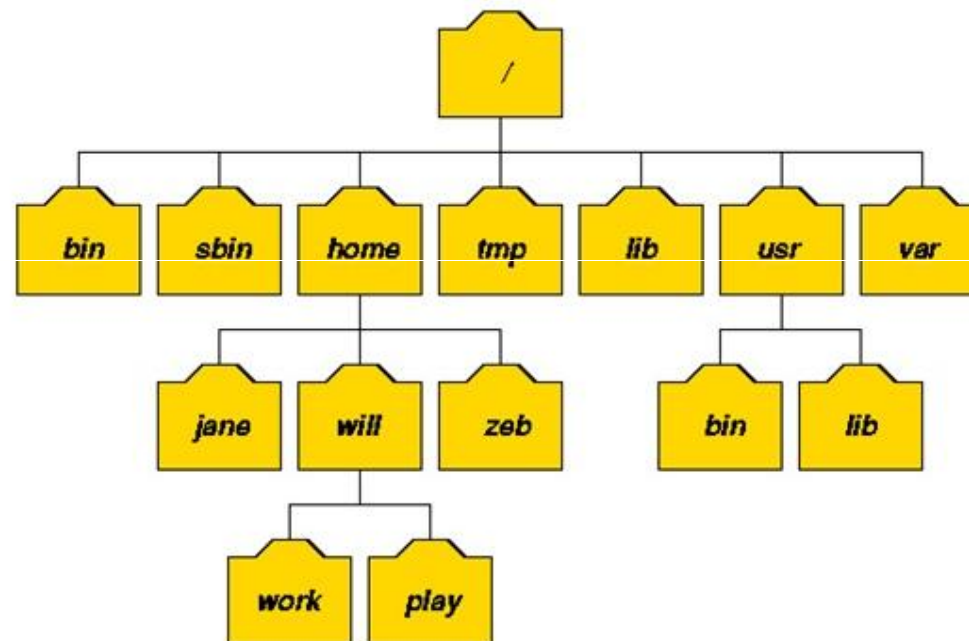
# Organisasi File



- 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 “/” .
- 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).



# Gambar Struktur Direktori pada Linux



# Direktori Standar



Setelah proses instalasi, Linux menciptakan system file yang baku, terdiri atas direktori sebagai berikut :

Direktori	Deskripsi
/etc	Berisi file administrative (konfigurasi dll) dan file executable atau script yang berguna untuk administrasi system.
/dev	Berisi file khusus yang merepresentasikan peralatan hardware seperti memori, disk, printer, tape, floppy, jaringan dll.
/bin	Berisi utilitas sistem level rendah (binary)...
/sbin	Berisi utilitas sistem untuk superuser (untuk membentuk administrasi sistem).
/usr/sbin /usr/bin	Berisi utilitas sistem dan program aplikasi level tinggi.
/usr/lib	Berisi program library yang diperlukan untuk kompilasi

	program (misalnya C). Berisi instruksi (command) misalnya untuk Print Spooler (lpadmin) dll.
/tmp	Berisi file sementara, yang pada saat Bootstrap akan dihapus (dapat digunakan oleh sembarang user).
/boot	Berisi file yang sangat penting untuk proses bootstrap. Kernel <b>vmlinuz</b> disimpan di direktori ini.
/proc	Berisi informasi tentang kernel Linux, proses dan virtual system file.
/var	Direktori variable, artinya tempat penyimpanan LOG (catatan hasil output program), file ini dapat membengkak dan perlu dimonitor perkembangannya.
/home	Berisi direktori untuk pemakai Linux (pada SCO diletakkan pada /usr)
/mnt	Direktori untuk mounting system file
/root	Home direktori untuk superuser (root)
/usr/bin/X11	Symbolic link ke /usr/X11R6/bin, program untuk X-Window
/usr/src	Source code untuk Linux
/opt	Option, direktori ini biasanya berisi aplikasi tambahan ("add-on") seperti Netscape Navigator, kde, gnome, applix dll.

# Direktori /etc



Berisi file yang berhubungan dengan administrasi system, maintenance script, konfigurasi, security dll. Hanya superuser yang boleh memodifikasi file yang berada di direktori ini. Subdirektori yang sering diakses pada direktori **/etc** antara lain :

- httpd, apache web server.
- ppp, point to point protocol untuk koneksi ke Internet.
- rc.d atau init.d, inisialisasi (startup) dan terminasi (shutdown) proses di Linux dengan konsep runlevel.
- cron.d, rincian proses yang dieksekusi dengan menggunakan jadwal(time dependent process)
- FILES, file security dan konfigurasi meliputi : *passwd, hosts, shadow, ftpaccess, inetd.conf, lilo.conf, motd, printcap, profile, resolv.conf, sendmail.cf, syslog.conf, dhcp.conf, smb.conf, fstab.*

# Direktori /dev



Konsep Unix dan Linux adalah memperlakukan peralatan hardware sama seperti penanganan file. Setiap alat mempunyai nama file yang disimpan pada direktori /dev.

Peralatan	Direktori
Floppy	/dev/fd0
Harddisk	IDE : /dev/had, /dev/hdb, /dev/hdc, /dev/hdd SCSI : /dev/sda, /dev/sdb, /dev/sdc
CDROM	SCSI : /dev/scd0, /dev/scd1 IDE : /dev/gscd, /dev/sonycd Universal : /dev/cdrom (link dari actual cdrom ide atau scsi)
Mouse	PS2 : /dev/lp0 Universal : /dev/mouse
Parallel Port	LPT1 : /dev/lp0 LPT2 : /dev/lp1
Serial Port	COM1 : /dev/ttyS0 COM2 : /dev/ttyS1 Universal : /dev/modem (link dari S0 atau S1)

## Direktori /proc



- Direktori /proc adalah direktori yang dibuat diatas RAM (Random Access Memory) dengan system file yang diatur oleh kernel. /proc berisi nomor proses dari system dan nama driver yang aktif di system.
- Semua direktori berukuran 0 (kosong) kecuali file kcore dan self. Setiap nomor yang ada pada direktori tsb merepresentasikan PID (Process ID).

# Symbolic Link



- Link adalah sebuah teknik untuk memberikan lebih dari satu nama file dengan data yang sama. Bila file asli dihapus, maka data yang baru juga terhapus. Format dari Link :
  - `ln fileAsli fileDuplikat`
- File Duplikat disebut *hard link* dimana kedua file akan muncul identik (*link count* = 2). Bila fileAsli atau file duplikat diubah perubahan akan terjadi pada file lainnya. Symbolic Link diperlukan bila file tersebut di “link” dengan direktori /file yang berada pada partisi yang berbeda. Tipe file menjadi l (link) dan file tersebut menunjuk ke tempat asal. Format : `ln -s /FULLPATH/fileAsli /FULLPATH/fileDuplikat`
- Pilihan `-s` (*shortcut*) merupakan bentuk *soft link* dimana jumlah *link count* pada file asal tidak akan berubah. Pada bentuk *soft link*, *symbolic link* dapat dilakukan pada file yang tidak ada, sedangkan pada *hard link* tidak dimungkinkan. Perbedaan lain, *symbolic link* dapat dibentuk melalui media disk atau partisi yang berbeda dengan *soft link*, tetapi pada *hard link* terbatas pada partisi disk yang sama.

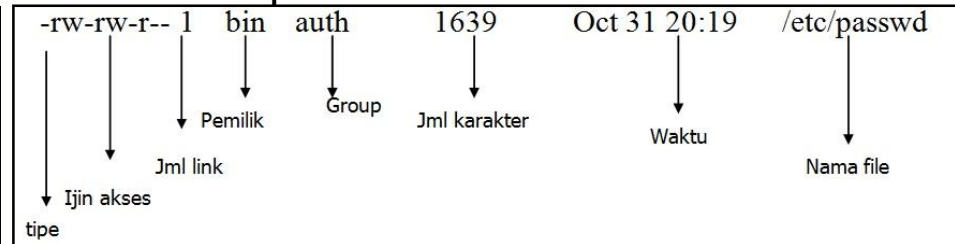
# Properti File



File mempunyai beberapa atribut, antara lain :

- Tipe file : menentukan tipe dari file

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

# Melihat Isi File



- Untuk melihat jenis file menggunakan format :
  - file *filename(s)*
- Isi file akan dilaporkan dengan deskripsi level tinggi seperti contoh berikut
  - \$ file myprog.c letter.txt webpage.html
  - myprog.c: C program text
  - letter.txt: ASCII text
  - webpage.html: HTML document text
- Perintah ini dapat digunakan secara luas untuk file yang kadang membingungkan, misalnya antara kode C++ dan Java



# Mencari File



Jika ingin melihat bagaimana pohon direktori dapat digunakan perintah

- find
  - Format : `find directory -name targetfile -print`
  - Akan melihat file yang bernama *targetfile* (bisa berupa karakter wildcard)
- which
  - Format : `which command`
- Mengetahui letak system utility
  - locate
  - Format : `locate string`
  - Akan mencari file pada semua direktori dengan lebih cepat dan ditampilkan dengan path yang penuh.

# Mencari Text Pada File



- Untuk mencari text pada file digunakan perintah grep (*General Regular Expression Print*) dengan format perintah
  - *grep option pattern files*
- Grep akan mencari file yang bernama sesuai pattern yang diberikan dan akan menampilkan baris yang sesuai

# Latihan



- Login sebagai user.
- Bukalah Console Terminal dan lakukan latihan-latihan pada setiap slide. Perhatikan hasilnya.
- Selesaikan soal-soal posttest

# Latihan 1 Direktori



1. Melihat direktori HOME

```
$ pwd
```

```
$ echo $HOME
```

2. Melihat direktori aktual dan parent direktori

```
$ pwd
```

```
$ cd .
```

```
$ pwd
```

```
$ cd ..
```

```
$ pwd
```

```
$ cd
```

3. Lakukan instalasi secara online dengan share resource antara host dan guest untuk koneksi internet dan berikan perintah sudo apt-get install [nama\_paket/tree] bagi yang guest OS nya belum terinclude paket tree.

4. Membuat satu direktori, lebih dari satu direktori atau sub direktori

```
$ pwd
```

```
$ mkdir A B C A/D A/E B/F  
A/D/A
```

```
$ ls -l
```

```
$ ls -l A
```

```
$ ls -l A/D
```

# Latihan 2 Direktori



1. Menghapus satu atau lebih direktori hanya dapat dilakukan pada direktori kosong dan hanya dapat dihapus oleh pemiliknya kecuali bila diberikan ijin aksesnya

```
$ rmdir B
```

(Terdapat pesan error, mengapa ?)

```
$ ls -l B
```

```
$ rmdir B/F B
```

```
$ ls -l B
```

(Terdapat pesan error, mengapa ?)

2. Navigasi direktori dengan instruksi `cd` untuk pindah dari satu direktori ke direktori lain.

```
$ pwd
```

```
$ ls -l
```

```
$ cd A
```

```
$ pwd
```

```
$ cd ..
```

```
$ pwd
```

```
$ cd /home/<user>/C
```

```
$ pwd
```

```
$ cd /<user>/C (Terdapat pesan error, mengapa ?)
```

```
$ pwd
```

# Latihan 3 Manipulasi File



1. Perintah `cp` untuk mengkopi file atau seluruh direktori

```
$ cat > contoh
```

Membuat sebuah file

[Ctrl-d]

```
$ cp contoh contoh1
```

```
$ ls -l
```

```
$ cp contoh A
```

```
$ ls -l A
```

```
$ cp contoh contoh1 A/D
```

```
$ ls -l A/D
```

2. Perintah `mv` untuk memindah file

```
$ mv contoh contoh2
```

```
$ ls -l
```

```
$ mv contoh1 contoh2 A/D
```

```
$ ls -l A/D
```

```
$ mv contoh contoh1 C
```

```
$ ls -l C
```

3 Perintah `rm` untuk menghapus file

```
$ rm contoh2
```

```
$ ls -l
```

```
$ rm -i contoh
```

```
$ rm -rf A C
```

```
$ ls -l
```

# Latihan 4 Symbolic Link



Membuat shortcut (file link)

```
$ echo "Hallo apa khabar" > halo.txt
```

```
$ ls -l
```

```
$ ln halo.txt z
```

```
$ ls -l
```

```
$ cat z
```

```
$ mkdir mydir
```

```
$ ln z mydir/halo.juga
```

```
$ cat mydir/halo.juga
```

```
$ ln -s z bye.txt
```

```
$ ls -l bye.txt
```

```
$ cat bye.txt
```

## Latihan 5 Melihat Isi File



```
$ ls -l
```

```
$ file halo.txt
```

```
$ file bye.txt
```



# Latihan 6 Mencari File



## Perintah find

```
$ find /home -name *.txt -print > myerror.txt
```

```
$ cat myerror.txt
```

```
$ find . -name *.txt -exec wc -l '{}' ';'
```

## Perintah which

```
$ which ls
```

## Perintah locate

```
$ locate "*.txt"
```

## Latihan 7 Mencari Text Pada File



```
$ grep Hallo *.txt
```

# Posttest



Cobalah dan urutkan perintah berikut :

```
$ cd  
$ pwd  
$ ls -al  
$ cd .  
$ pwd  
$ cd ..  
$ pwd  
$ ls -al  
$ cd ..
```

1

```
$ $ pwd  
$ ls -al  
$ cd /etc  
$ ls -al | more  
$ cat passwd  
$ cd -  
$ pwd
```

2