

# **Praktikum Sistem Operasi Proses dan Manajemen Proses**



**Minggu ke 7-8**



**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

# Konsep Proses Pada Sistem Operasi Linux



- Proses adalah program yang sedang dieksekusi. Setiap kali menggunakan utilitas sistem atau program aplikasi dari shell, satu atau lebih proses "child" akan dibuat oleh shell sesuai perintah yang diberikan. Setiap kali instruksi diberikan pada Linux shell, maka kernel akan menciptakan sebuah proses-id.
- Proses ini disebut juga dengan terminology Unix sebagai sebuah Job. Proses Id (PID) dimulai dari 0, yaitu proses INIT, kemudian diikuti oleh proses berikutnya (terdaftar pada `/etc/inittab`).

# Tipe Proses



- **Foreground** : Proses yang diciptakan oleh pemakai langsung pada terminal (interaktif, dialog)
- **Batch** : Proses yang dikumpulkan dan dijalankan secara sekuensial (satu persatu). Proses Batch tidak diasosiasikan (berinteraksi) dengan terminal
- **Daemon** : Proses yang menunggu permintaan (request) dari proses lainnya dan menjalankan tugas sesuai dengan permintaan tersebut. Bila tidak ada request, maka program ini akan berada dalam kondisi “idle” dan tidak menggunakan waktu hitung CPU. Umumnya nama proses daemon di UNIX berakhiran d, misalnya inetd, named, popd dll.

## Tipe Proses Cont...



- Contoh Daemon : Apache Web Server HTTP daemon. Daemon ini bekerja pada background dan menerima request HTTP pada port tertentu (biasanya 80 atau 8080) dan memberikan respon terhadap request tersebut, berdasarkan tipe dari request tersebut.

# Sinyal



- Proses dapat mengirim dan menerima sinyal dari dan ke proses lainnya. Proses mengirim sinyal melalui instruksi “kill” dengan format

kill [-nomor sinyal] PID

- Nomor sinyal : 1 s/d maksimum nomor sinyal yang didefinisikan system.
- Standar nomor sinyal yang terpenting adalah :

| No Sinyal | Nama    | Deskripsi                                                                             |
|-----------|---------|---------------------------------------------------------------------------------------|
| 1         | SIGHUP  | Hangup, sinyal dikirim bila proses terputus, misalnya melalui putusnya hubungan modem |
| 2         | SIGINT  | Sinyal interrupt, melalui ^C                                                          |
| 3         | SIGQUIT | Sinyal Quit, melalui ^\                                                               |
| 9         | SIGKILL | Sinyal Kill, menghentikan proses                                                      |
| 15        | SIGTERM | Sinyal terminasi software                                                             |

# Mengirim Sinyal



- Mengirim sinyal adalah satu alat komunikasi antar proses, yaitu memberitahukan proses yang sedang berjalan bahwa ada sesuatu yang harus dikendalikan. Berdasarkan sinyal yang dikirim ini maka proses dapat bereaksi dan administrator/programmer dapat menentukan reaksi tersebut. Mengirim sinyal menggunakan instruksi
  - `kill [-nomor sinyal] PID`
- Sebelum mengirim sinyal PID proses yang akan dikirim harus diketahui terlebih dahulu.

# Mengontrol Proses Pada Shell



- Shell menyediakan fasilitas job control yang memungkinkan mengontrol beberapa job atau proses yang sedang berjalan pada waktu yang sama. Misalnya bila melakukan pengeditan file teks dan ingin melakukan interrupt pengeditan untuk mengerjakan hal lainnya. Bila selesai, dapat kembali (*switch*) ke editor dan melakukan pengeditan file teks kembali.
- Job bekerja pada **foreground** atau **background**. Pada *foreground* hanya diperuntukkan untuk satu job pada satu waktu. Job pada *foreground* akan mengontrol shell - menerima input dari keyboard dan mengirim output ke layar. Job pada background tidak menerima input dari terminal, biasanya berjalan tanpa memerlukan interaksi.

# Mengontrol Proses Pada Shell Cont...



- Job pada *foreground* kemungkinan dihentikan sementara (suspend), dengan menekan [Ctrl-Z]. Job yang dihentikan sementara dapat dijalankan kembali pada *foreground* atau *background* sesuai keperluan dengan menekan "fg" atau "bg". Sebagai catatan, menghentikan job sementara sangat berbeda dengan melakukan interrupt job (biasanya menggunakan [Ctrl-C]), dimana job yang diinterrupt akan dimatikan secara permanen dan tidak dapat dijalankan lagi



# Mengontrol Proses Lain



- Perintah `ps` dapat digunakan untuk menunjukkan semua proses yang sedang berjalan pada mesin (bukan hanya proses pada shell saat ini) dengan format :
  - `ps -fae` atau
  - `ps -aux`
- Beberapa versi UNIX mempunyai utilitas sistem yang disebut `top` yang menyediakan cara interaktif untuk memonitor aktifitas sistem. Statistik secara detail dengan proses yang berjalan ditampilkan dan secara terus-menerus di-*refresh*. Proses ditampilkan secara terurut dari utilitas CPU. Kunci yang berguna pada `top` adalah
  - `s` – set update frequency
  - `u` – display proses dari satu user
  - `k` – kill proses (dengan PID)
  - `q` – quit
- Utilitas untuk melakukan pengontrolan proses dapat ditemukan pada sistem UNIX adalah perintah `killall`. Perintah ini akan menghentikan proses sesuai PID atau job number proses

# Latihan



1. Login sebagai user (user sesuaikan dengan username anda)
2. Lakukan latihan-latihan berikut ini kemudian perhatikan hasilnya.
3. Selesaikan soal-soal postest.

# Latihan 1 Status Proses



1. Pindah ke *command line terminal* (tty2) dengan menekan **Ctrl+Alt+F2** dan login ke terminal sebagai user.
2. Instruksi *ps* (*process status*) digunakan untuk melihat kondisi proses yang ada. PID adalah Nomor Identitas Proses, TTY adalah nama terminal dimana proses tersebut aktif, STAT berisi S (*Sleeping*) dan R (*Running*), COMMAND merupakan instruksi yang digunakan.

\$ *ps*

3. Untuk melihat faktor/elemen lainnya, gunakan option *-u* (user). %CPU adalah presentasi CPU time yang digunakan oleh proses tersebut, %MEM adalah presentasi system memori yang digunakan proses, SIZE adalah jumlah memori yang digunakan, RSS (*Real System Storage*) adalah jumlah memori yang digunakan, START adalah kapan proses tersebut diaktifkan

\$ *ps -u*

## Latihan 2 Status Proses



1. Mencari proses yang spesifik pemakai. Proses diatas hanya terbatas pada proses milik pemakai, dimana pemakai tersebut melakukan login

```
$ ps -u <user>
```

2. Mencari proses lainnya gunakan opsi a (*all*) dan au (*all user*)

```
$ ps -a
```

```
$ ps -au
```

3. **Logout** dan tekan **Alt+F7** untuk kembali ke mode grafis

# Latihan 3 Menampilkan Hubungan Proses Parent dan Child



1. Ketik **ps -eH** dan tekan **Enter**. Opsi **e** memilih semua proses dan opsi **H** menghasilkan tampilan proses secara hierarki. Proses child muncul dibawah proses parent. Proses child ditandai dengan awalan beberapa spasi.

*\$ ps -eH*

2. Ketik **ps -e f** dan tekan **Enter**. Tampilan serupa dengan langkah 1. Opsi **-f** akan menampilkan status proses dengan karakter grafis (**\** dan **\_**)

*\$ ps -e f*

3. Ketik **pstree** dan tekan **Enter**. Akan ditampilkan semua proses pada sistem dalam bentuk hirarki parent/child. Proses parent di sebelah kiri proses child. Sebagai contoh proses init sebagai parent (*ancestor*) dari semua proses pada sistem. Beberapa child dari init mempunyai child. Proses login mempunyai proses bash sebagai child. dan seterusnya.

*\$ pstree*

## Latihan 4 Menampilkan Hubungan Proses Parent dan Child Cont...



1. Ketik **ps tree | grep login** dan tekan **Enter**. Akan menampilkan semua proses login  
*\$ ps tree | grep login* atau *\$ ps tree | grep mingetty*
2. Untuk melihat semua PID untuk proses gunakan opsi **-p**.  
*\$ ps tree -p*
3. Untuk menampilkan proses dan ancestor yang tercetak tebal gunakan opsi **-h**.  
*\$ ps tree -h*

# Latihan 5 Menampilkan Status Proses dengan Berbagai Format



1. Pindah ke *command line terminal* (tty2) dengan menekan **Ctrl+Alt+F2** dan login ke terminal sebagai user.
2. Ketik **ps -e | more** dan tekan **Enter**. Opsi **-e** menampilkan semua proses dalam bentuk 4 kolom : PID, TTY, TIME dan CMD.

```
$ ps -e | more
```

Jika halaman penuh terlihat prompt **--More--** di bagian bawah screen, tekan **q** untuk kembali ke prompt perintah.

3. Ketik **ps ax | more** dan tekan **Enter**. Opsi **a** akan menampilkan semua proses yang dihasilkan terminal (TTY). Opsi **x** menampilkan semua proses yang tidak dihasilkan terminal. Secara logika opsi ini sama dengan opsi **-e**. Terdapat 5 kolom : PID, TTY, STAT, TIME dan COMMAND.

```
$ ps ax | more
```

Jika halaman penuh terlihat prompt **--More--** di bagian bawah screen, tekan **q** untuk kembali ke prompt perintah.

## Latihan 6 Menampilkan Status Proses dengan Berbagai Format Cont...



1. Ketik **ps -ef | more** dan tekan **Enter**. Opsi **-ef** akan menampilkan semua proses dalam format daftar penuh.

*\$ ps ef | more*

Jika halaman penuh terlihat prompt **--More--** di bagian bawah screen, tekan **q** untuk kembali ke prompt perintah.

2. Ketik **ps -eo pid, cmd | more** dan tekan **Enter**. Opsi **-eo** akan menampilkan semua proses dalam format sesuai definisi user yaitu terdiri dari kolom PID dan CMD.

*\$ ps -eo pid,cmd | more*

Jika halaman penuh terlihat prompt **--More--** di bagian bawah screen, tekan **q** untuk kembali ke prompt perintah.



## Latihan 7 Menampilkan Status Proses dengan Berbagai Format Cont...



1. Ketik **ps -eo pid,ppid,%mem,cmd | more** dan tekan **Enter**. Akan menampilkan kolom PID, PPID dan %MEM. PPID adalah proses ID dari proses parent. %MEM menampilkan persentasi memory system yang digunakan proses. Jika proses hanya menggunakan sedikit memory system akan ditampilkan 0.

```
$ ps -eo pid,ppid,%mem,cmd | more
```

2. **Logout** dan tekan **Alt+F7** untuk kembali ke mode grafis

# Latihan 8 Mengontrol Proses Pada Shell



1. Gunakan perintah `yes` yang mengirim output `y` yang tidak pernah berhenti

`$ yes` → Untuk menghentikannya gunakan **Ctrl-C**.

2. Belokkan standart output ke `/dev/null`

`$ yes > /dev/null` → Untuk menghentikannya gunakan **Ctrl-C**.

3. Salah satu cara agar perintah `yes` tetap dijalankan tetapi shell tetap digunakan untuk hal yang lain dengan meletakkan proses pada *background* dengan menambahkan karakter `&` pada akhir perintah.

`$ yes > /dev/null &`

Angka dalam "[ ]" merupakan **job number** diikuti PID.

4. Untuk melihat status proses gunakan perintah `jobs`. → `$ jobs`

# Latihan 9 Mengontrol Proses Pada Shell



1. Untuk menghentikan job, gunakan perintah kill diikuti *job number* atau PID proses. Untuk identifikasi job number, diikuti prefix dengan karakter "%".

`$ kill %<nomor job>` contoh : `kill %1`

2. Lihat status job setelah diterminasi

`$ jobs`

# Latihan 10 Menghentikan dan Memulai Kembali Job



1. Cara lain meletakkan job pada *background* dengan memulai job secara normal (pada *foreground*), **stop** job dan memulai lagi pada *background*

```
$ yes > /dev/null
```

Hentikan sementara job (*suspend*), bukan menghentikannya (*terminate*), tetapi menghentikan sementara job sampai di restart. Untuk menghentikan sementara job gunakan **Ctrl-Z**.

2. Untuk restart job pada *foreground*, gunakan perintah

```
fg.
```

```
$ fg
```

# Latihan 11 Menghentikan dan Memulai Kembali Job Cont...



1. Shell akan menampilkan nama perintah yang diletakkan di *foreground*. Stop job lagi dengan **Ctrl-Z**. Kemudian gunakan perintah `bg` untuk meletakkan job pada *background*.

`$ bg`

2. Job tidak bisa dihentikan dengan **Ctrl-Z** karena job berada pada *background*. Untuk menghentikannya, letakkan job pada *foreground* dengan `fg` dan kemudian hentikan sementara dengan **Ctrl-Z**.

`$ fg`

# Latihan 12 Menghentikan dan Memulai Kembali Job Cont...



1. Job pada *background* dapat digunakan untuk menampilkan teks pada terminal, dimana dapat diabaikan jika mencoba mengerjakan job lain. → `$ yes &`  
Untuk menghentikannya tidak dapat menggunakan **Ctrl-C**. Job harus dipindah ke *foreground*, baru dihentikan dengan cara tekan **fg** dan tekan **Enter**, kemudian dilanjutkan dengan **Ctrl-Z** untuk menghentikan sementara.
2. Apabila ingin menjalankan banyak job dalam satu waktu, letakkan job pada *foreground* atau *background* dengan memberikan job ID
  1. `$ fg %2` atau `$ %2`
  2. `$ bg %2`

(Kemudian ketikkan perintah `$yes` dilanjutkan dengan `Ctrl-Z`)

## Latihan 13 Menghentikan dan Memulai Kembali Job Cont...



1. Ketikkan perintah `$jobs` untuk melihat jobs yang sudah dikerjakan
2. Lihat job dengan perintah `ps -fae` dan tekan **Enter**. Kemudian hentikan proses dengan perintah `kill`.

```
$ ps -fae
```

```
$ kill -9 <NomorPID>
```