

Bab 2

Mengontrol dan memonitor proses

Ketika suatu sistem linux dijalankan, dan berbagai program dieksekusi, hal ini memerlukan suatu manajemen dalam mengontrol dan memonitor setiap proses. Proses di linux tercatat dalam bentuk bilangan angka (PID), sehingga setiap proses dapat diamati dari memory yang dipakai, user yang menjalankannya (*process owner*), lamanya proses tersebut di jalankan dan lain sebagainya. Proses yang berlangsung dapat berupa *daemon*, aplikasi pengguna, utilitas dan lain lain.

Proses yang dijalankan di console, shell, *xterm*, dan lain-lainnya disebut sebagai *foreground process*. sedangkan *background process* adalah proses yang dijalankan secara *background* dan biasanya di jalankan dengan menambahkan "&" pada akhir dari baris perintah.

contoh :

```
$ ping 192.168.0.1 &
```

Perintah diatas akan mengakibatkan mesin akan melakukan ping ke sistem dengan ip 192.168.0.1 secara kontinyu dan bekerja di background (di belakang layar) Untuk memonitor proses-proses yang dijalankan oleh system dapat di gunakan perintah *ps* dan *top*. Sedangkan untuk mengontrol proses dalam dilakukan dengan utilitas *kill*, yang akan mengirim signal kepada suatu proses.

ps

Utilitas *ps* ini digunakan untuk melihat proses apa saja yang sedang dijalankan system. option yang umum digunakan adalah "-aux", untuk option lengkapnya dapat dilihat dengan mengeksekusi "*ps --help*". Sintaks dari utilitas ini:

```
$ ps [option]
```

contoh :

```
antoro@drutz:~$ ps aux
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.4  1020    64 ?        S    10:12   0:04 init [2]
root         2  0.0  0.0      0     0 ?        SW   10:12   0:00 [kflushd]
root         3  0.0  0.0      0     0 ?        SW   10:12   0:00 [kupdate]
root         4  0.0  0.0      0     0 ?        SW   10:12   0:00 [kpiod]
root         5  0.0  0.0      0     0 ?        SW   10:12   0:05 [kswapd]
root       149  0.0  1.5  1352   216 ?        S    10:15   0:00 /sbin/syslogd
root       151  0.0  1.2  1396   164 ?        S    10:15   0:04 /sbin/klogd
root       155  0.0  5.4  2156   740 ?        S    10:15   0:02 /usr/sbin/named
disappea  159  0.0  0.0   1748     0 ?        SW   10:15   0:00 [safe_mysqld]
root       160  0.1 11.5  7440  1572 ?        SN   10:15   0:29 perl ./infobot
root       176  0.0  2.7   1060   368 ttyS0    S    10:15   0:01 /usr/sbin/gpm an-
toro@drutz:~$
```

Keterangan :

USER : pemilik proses
 PID : proses ID (Nomor Proses)
 %CPU : penggunaan CPU oleh proses
 %MEM : penggunaan Memory (dalam KB) oleh proses
 VSZ : penggunaan virtual memory dalam proses
 RSS : penggunaan memory yang resident oleh proses
 TTY : terminal (tty), tempat asal proses tsb dijalankan
 STAT : keadaan suatu proses
 - R : proses dalam kondisi *Running*
 - S : proses dalam kondisi *Sleeping*
 - I : proses dalam kondisi *idle*
 - Z : proses *Zombie*
 - D : uninterruptible sleep (biasanya IO)
 - < : proses dengan prioritas tinggi
 - N : proses dengan prioritas rendah
 - dan lain lain
 START : saat proses dijalankan
 TIME : laman proses dijalankan
 COMMAND : Baris perintah yang dijalankan

top

Utilitas ini secara mendasar fungsinya sama seperti `ps`, hanya saja `top` lebih aktual karena setelah kita eksekusi, `top` akan update proses yang dijalankan secara berkala. Juga `top` lebih spesifik dalam menjelaskan keadaan sistem. Kelemahannya adalah proses `top` ini lebih memakan banyak memory. Sintaks dari utilitas ini :

```
$ top
```

contoh :

```

antoro@drutz:~$ top
 3:37pm up 5:24, 3 users, load average: 0.05, 0.06, 0.00
34 processes: 33 sleeping, 1 running, 0 zombie, 0 stopped
CPU states: 3.6% user, 9.9% system, 0.0% nice, 86.3% idle
Mem: 13624K av, 12564K used, 1060K free, 8340K shrd, 380K buff
Swap: 62460K av, 11700K used, 50760K free, 6012K cached

  PID USER   PRI  NI  SIZE  RSS SHARE STAT   LIB %CPU %MEM   TIME COMMAND
  689 antoro   14    0 1308 1308   700 R       0  8.0  9.6   0:22 top
  176 root      6    0  372  368   316 S       0  4.3  2.7   0:02 gpm
    1 root      0    0  108   64    48 S       0  0.0  0.4   0:04 init
    2 root      0    0     0     0     0 SW      0  0.0  0.0   0:00 kflushd
    3 root      0    0     0     0     0 SW      0  0.0  0.0   0:00 kupdate
    4 root      0    0     0     0     0 SW      0  0.0  0.0   0:00 kpiod
    5 root      0    0     0     0     0 SW      0  0.0  0.0   0:05 kswapd
  149 root      0    0  264  216   152 S       0  0.0  1.5   0:00 syslogd
  151 root      0    0  576  164   136 S       0  0.0  1.2   0:04 klogd
.... dst

```

Dari dua perintah `ps` dan `top` diatas dikenal sebutan proses zombie. Proses zombie adalah suatu proses (biasanya proses anak) yang telah mati/dimatikan tetapi tetap ada. Hal ini dikarenakan proses ibunya (parent process) tidak mematikan proses anaknya dengan sempurna. Proses zombie ini akan mati apabila parent prosesnya mati.

kill

Setiap proses yang dijalankan di linux dapat dideteksi dan kemudian di matikan, seperti telah dijelaskan pada **Modul Linux Dasar** dalam perintah-perintah dasar Linux.