

Modul VI

Manajemen Proses

SASARAN

Setelah mempelajari materi dalam bab ini, mahasiswa diharapkan mampu:

1. Memahami konsep proses pada sistem operasi Linux.
2. Menampilkan beberapa cara menampilkan hubungan proses parent dan child.
3. Menampilkan status proses dengan beberapa format berbeda.
4. Melakukan pengontrolan proses pada shell.
5. Memahami penjadwalan prioritas.

PRAKTIKUM

1. Login sebagai user.
2. Download program C++ untuk menampilkan bilangan prima yang bernama primes.
3. Lakukan percobaan-percobaan di bawah ini kemudian analisa hasil percobaan.
4. Selesaikan soal-soal latihan.

Percobaan 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
seed@Attacker_Riama(127.0.0.1):~$ ps
  PID TTY          TIME CMD
 3810 tty6      00:00:00 bash
 3926 tty6      00:00:00 ps
```

Analisis :

TIME untuk menampilkan waktu yang diperlukan untuk mengakses instruksi
CMD digunakan untuk menampilkan instruksi

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
```

```
seed@Attacker_Riama(127.0.0.1):~$ ps -u
Warning: bad ps syntax, perhaps a bogus '-'? See http://procps.sf.net/faq.html
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
seed      3810  0.3  0.6  10012  6396 tty6      S    22:38   0:00 -bash
seed      3991  0.0  0.1   4944  1148 tty6      R+   22:39   0:00 ps -u
```

Analisis :

USER merupakan informasi user yang sedang/ dalam proses saat itu

PID merupakan informasi nomor identitas dari proses

CPU mempresentasikan waktu yang digunakan CPU dalam proses

MEM mempresentasikan sistem memori yang digunakan dalam proses

RSS memberikan informasi mengenai jumlah memori yang digunakan

START memberikan informasi waktu proses diaktifkan

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

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

```
2918 ?      00:00:00 unity-applicati
2920 ?      00:00:00 unity-files-dae
2922 ?      00:00:00 unity-music-dae
2924 ?      00:00:00 unity-lens-vide
2960 ?      00:00:00 zeitgeist-daemo
2966 ?      00:00:00 zeitgeist-fts
2977 ?      00:00:00 zeitgeist-datah
2982 ?      00:00:00 cat
3006 ?      00:00:00 unity-musicstor
3007 ?      00:00:00 unity-scope-uid
3038 ?      00:00:00 gdu-notificatio
3059 ?      00:00:00 telepathy-indic
3066 ?      00:00:00 mission-control
3071 ?      00:00:00 goa-daemon
3077 ?      00:00:00 gnome-screensav
3123 ?      00:00:00 update-notifier
3145 ?      00:00:04 update-manager
3239 ?      00:00:00 deja-dup-monito
3810 tty6    00:00:00 bash
4219 ?      00:00:00 notify-osd
4528 tty1    00:00:00 bash
4626 tty1    00:00:00 ps
```

Analisis :

Perintah ps -u <user> digunakan untuk mencari proses spesifik yang dijalankan dan terbatas pada proses yang dijalankan pengguna

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

```
$ ps -a
```

```
$ ps -au
```

```

seed@Attacker_Riana(127.0.0.1):~$ ps -a
  PID TTY          TIME CMD
 3810 tty6      00:00:00 bash
 4528 tty1      00:00:00 bash
 4902 tty1      00:00:00 ps
seed@Attacker_Riana(127.0.0.1):~$ ps -au
Warning: bad ps syntax, perhaps a bogus '-'? See http://procps.sf.net/faq.html
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         789  0.0  0.0   4628   852 tty4      Ss+  22:25   0:00 /sbin/getty -8
root         797  0.0  0.0   4628   852 tty5      Ss+  22:25   0:00 /sbin/getty -8
root         810  0.0  0.0   4628   856 tty2      Ss+  22:25   0:00 /sbin/getty -8
root         811  0.0  0.0   4628   860 tty3      Ss+  22:25   0:00 /sbin/getty -8
root         816  0.0  0.1   3732   1904 tty6      Ss   22:25   0:00 /bin/login --
root         909  0.2  3.3  64452 34356 tty7      Ss+  22:25   0:04 /usr/bin/X :0 -
seed        3810  0.0  0.6  10012   6396 tty6      S+   22:38   0:00 -bash
root        4498  0.0  0.1   3732   1908 tty1      Ss   22:47   0:00 /bin/login --
seed        4528  0.1  0.6  10012   6392 tty1      S    22:48   0:00 -bash
seed        4903  0.0  0.1   4944   1160 tty1      R+   22:52   0:00 ps -au

```

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

Percobaan 2 : Menampilkan Hubungan Proses Parent dan Child

1. Pindah ke *command line terminal* (tty2) dengan menekan **Ctrl+Alt+F2** dan login ke terminal sebagai user.
2. 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

```

2922 ?          00:00:00  unity-music-dae
2924 ?          00:00:00  unity-lens-vide
2960 ?          00:00:00  zeitgeist-daemo
2966 ?          00:00:00  zeitgeist-fts
2982 ?          00:00:00      cat
2977 ?          00:00:00  zeitgeist-datah
3006 ?          00:00:00  unity-musicstor
3007 ?          00:00:00  unity-scope-vid
3066 ?          00:00:00  mission-control
3071 ?          00:00:00  goa-daemon
3141 ?          00:00:00  system-service-
3145 ?          00:00:04  update-manager
4219 ?          00:00:00  notify-osd
4498 tty1       00:00:00  login
4528 tty1       00:00:00  bash
4780 ?          00:00:00  anacron

```

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

\$ ps -ef

```

2966 ?          S1      0:00 /usr/lib/zeitgeist/zeitgeist-fts
2982 ?          S        0:00  \_ /bin/cat
2977 ?          S1      0:00 zeitgeist-datahub
3006 ?          S1      0:00 /usr/lib/unity-lens-music/unity-musicstore-daemon
3007 ?          S1      0:00 /usr/bin/python /usr/lib/unity-scope-video-remote/uni
3066 ?          S1      0:00 /usr/lib/telepathy/mission-control-5
3071 ?          S1      0:00 /usr/lib/gnome-online-accounts/goa-daemon
3141 ?          S        0:00 /usr/bin/python /usr/lib/system-service/system-servic
3145 ?          SN1    0:04 /usr/bin/python /usr/bin/update-manager --no-focus-on
4219 ?          S1      0:00 /usr/lib/notify-osd/notify-osd
4498 tty1       Ss      0:00 /bin/login --
4528 tty1       S+      0:00  \_ -bash
4780 ?          Ss      0:00 anacron -s

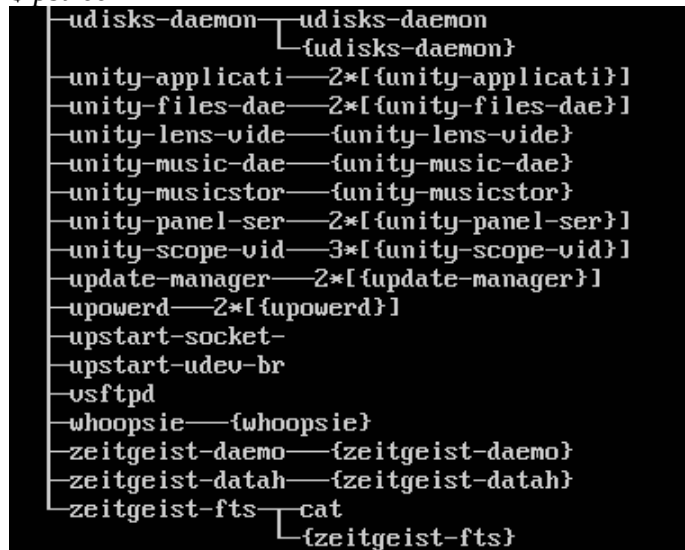
```

Analisis :

Perintah ini digunakan untuk mengetahui STAT (keadaan) dari proses yang ditandai dengan simbol S atau R.

4. Ketik **ps tree** 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 i proses bash sebagai child. Proses bash mempunyai proses child startx. Proses startx mempunyai child xinit dan seterusnya.

\$ *ps tree*



5. Ketik **ps tree | grep mingetty** dan tekan **Enter**. Akan menampilkan semua proses mingetty yang berjalan pada system yang berupa *console virtual*. Selain menampilkan semua proses, proses dikelompokkan dalam satu baris dengan suatu angka sebagai jumlah proses yang berjalan.

\$ *ps tree | grep mingetty*

```
seed@Attacker_Riama(127.0.0.1):~$ ps tree | grep mingetty
seed@Attacker_Riama(127.0.0.1):~$
```

6. Untuk melihat semua PID untuk proses gunakan opsi **-p**.

```
$ pstree -p
```

```
unity-panel-ser(2833)─{unity-panel-ser}(2834)
                    │{unity-panel-ser}(2835)
unity-scope-vid(3007)─{unity-scope-vid}(3023)
                    │{unity-scope-vid}(3025)
update-manager(3145)─{update-manager}(3147)
                    │{update-manager}(3148)
upowerd(1642)─{upowerd}(1646)
              │{upowerd}(1651)
upstart-socket-(599)
upstart-udev-br(348)
vsftpd(846)
whoopsie(852)─{whoopsie}(860)
zeitgeist-daemo(2960)─{zeitgeist-daemo}(2961)
zeitgeist-datah(2977)─{zeitgeist-datah}(2983)
zeitgeist-fts(2966)─cat(2982)
                  │{zeitgeist-fts}(2981)
```

7. Untuk menampilkan proses dan ancestor yang tercetak tebal gunakan opsi **-h**.

```
$ pstree -h
```

```
unity-panel-ser─2*[{unity-panel-ser}]
unity-scope-vid─3*[{unity-scope-vid}]
update-manager─2*[{update-manager}]
upowerd─2*[{upowerd}]
upstart-socket-
upstart-udev-br
vsftpd
whoopsie─{whoopsie}
zeitgeist-daemo─{zeitgeist-daemo}
zeitgeist-datah─{zeitgeist-datah}
zeitgeist-fts─cat
               │{zeitgeist-fts}
```

Percobaan 3 : 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
```

```
PID TTY      TIME CMD
  1 ?        00:00:00 init
  2 ?        00:00:00 kthreadd
  3 ?        00:00:00 ksoftirqd/0
  5 ?        00:00:00 kworker/u:0
  6 ?        00:00:00 migration/0
  7 ?        00:00:00 watchdog/0
  8 ?        00:00:00 cpuset
  9 ?        00:00:00 khelper
 10 ?        00:00:00 kdevmcfs
 11 ?        00:00:00 netns
 12 ?        00:00:00 sync_supers
 13 ?        00:00:00 bdi-default
 14 ?        00:00:00 kintegrityd
 15 ?        00:00:00 kblockd
 16 ?        00:00:00 ata_sff
 17 ?        00:00:00 khubd
 18 ?        00:00:00 md
 22 ?        00:00:00 khungtaskd
 23 ?        00:00:00 kswapd0
 24 ?        00:00:00 ksmd
 25 ?        00:00:00 khugepaged
 26 ?        00:00:00 fsnotify_mark
 27 ?        00:00:00 ecryptfs-kthrea
 28 ?        00:00:00 crypto
 37 ?        00:00:00 kthrotld
 40 ?        00:00:00 scsi_eh_0
 41 ?        00:00:00 scsi_eh_1
 42 ?        00:00:00 kworker/u:3
--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**

PID	TTY	STAT	TIME	COMMAND
1	?	Ss	0:00	/sbin/init
2	?	S	0:00	[kthreadd]
3	?	S	0:00	[ksoftirqd/0]
5	?	S	0:00	[kworker/u:0]
6	?	S	0:00	[migration/0]
7	?	S	0:00	[watchdog/0]
8	?	S<	0:00	[cpuset]
9	?	S<	0:00	[khelper]
10	?	S	0:00	[kdevtmpfs]
11	?	S<	0:00	[netns]
12	?	S	0:00	[sync_supers]
13	?	S	0:00	[bdi-default]
14	?	S<	0:00	[kintegrityd]
15	?	S<	0:00	[kblockd]
16	?	S<	0:00	[ata_sff]
17	?	S	0:00	[khubd]
18	?	S<	0:00	[md]
22	?	S	0:00	[khungtaskd]
23	?	S	0:00	[kswapd0]
24	?	SN	0:00	[ksmd]
25	?	SN	0:00	[khugepaged]
26	?	S	0:00	[fsnotify_mark]
27	?	S	0:00	[ecryptfs-kthrea]
28	?	S<	0:00	[crypto]
37	?	S<	0:00	[kthrotld]
40	?	S	0:00	[scsi_eh_0]
41	?	S	0:00	[scsi_eh_1]
42	?	S	0:00	[kworker/u:3]

More--

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

4. Ketik **ps -e f | more** dan tekan **Enter**. Opsi **-e f** akan menampilkan semua proses dalam format daftar penuh.

\$ **ps -e f | more**

PID	TTY	STAT	TIME	COMMAND
2	?	S	0:00	[kthreadd]
3	?	S	0:00	_ [ksoftirqd/0]
5	?	S	0:00	_ [kworker/u:0]
6	?	S	0:00	_ [migration/0]
7	?	S	0:00	_ [watchdog/0]
8	?	S<	0:00	_ [cpuset]
9	?	S<	0:00	_ [khelper]
10	?	S	0:00	_ [kdevtmpfs]
11	?	S<	0:00	_ [netns]
12	?	S	0:00	_ [sync_supers]
13	?	S	0:00	_ [bdi-default]
14	?	S<	0:00	_ [kintegrityd]
15	?	S<	0:00	_ [kblockd]

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

5. 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**

3. Blokirkan standar output ke /dev/null

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

Untuk menghentikannya gunakan **Ctrl-C**.

```
seed@Attacker_Riama(127.0.0.1):~$ yes > /dev/null  
^C  
seed@Attacker_Riama(127.0.0.1):~$
```

4. 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.

```
seed@Attacker_Riama(127.0.0.1):~$ yes > /dev/null &  
[1] 6932  
seed@Attacker_Riama(127.0.0.1):~$
```

5. Untuk melihat status proses gunakan perintah `jobs`.

```
$ jobs
```

```
seed@Attacker_Riama(127.0.0.1):~$ jobs  
[1]+  Running                  yes > /dev/null &  
seed@Attacker_Riama(127.0.0.1):~$
```

6. 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
```

7. Lihat status job setelah diterminasi

```
$ jobs
```



```
seed@Attacker_Riama(127.0.0.1):~$ kill %1
seed@Attacker_Riama(127.0.0.1):~$ jobs
[1]+  Terminated                  yes > /dev/null
```

Percobaan 5 : 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**.

```
seed@Attacker_Riama(127.0.0.1):~$ yes > /dev/null
^Z
[1]+  Stopped                  yes > /dev/null
```

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

```
$ fg
seed@Attacker_Riama(127.0.0.1):~$ fg
yes > /dev/null
^Z
[1]+  Stopped                  yes > /dev/null
```

3. 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
```

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
seed@Attacker_Riama(127.0.0.1):~$ bg
[1]+  yes > /dev/null &
seed@Attacker_Riama(127.0.0.1):~$ fg
yes > /dev/null
^Z
[1]+  Stopped                  yes > /dev/null
```

4. 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.

Analisis : Sudah dicoba, muncul tulisan [Stopped], kemudian proses berlanjut kembali

5. Apabila ingin menjalankan banyak job dalam satu waktu, letakkan job pada *foreground* atau *background* dengan memberikan job ID

\$ fg %2 atau \$%2

\$ bg %2

6. tekan **fg** dan tekan **Enter**, kemudian dilanjutkan dengan **Ctrl -Z** untuk menghentikan sementara.
7. Lihat job dengan perintah **ps -fae** dan tekan **Enter**. Kemudian hentikan proses dengan perintah **kill**.

\$ ps -fae

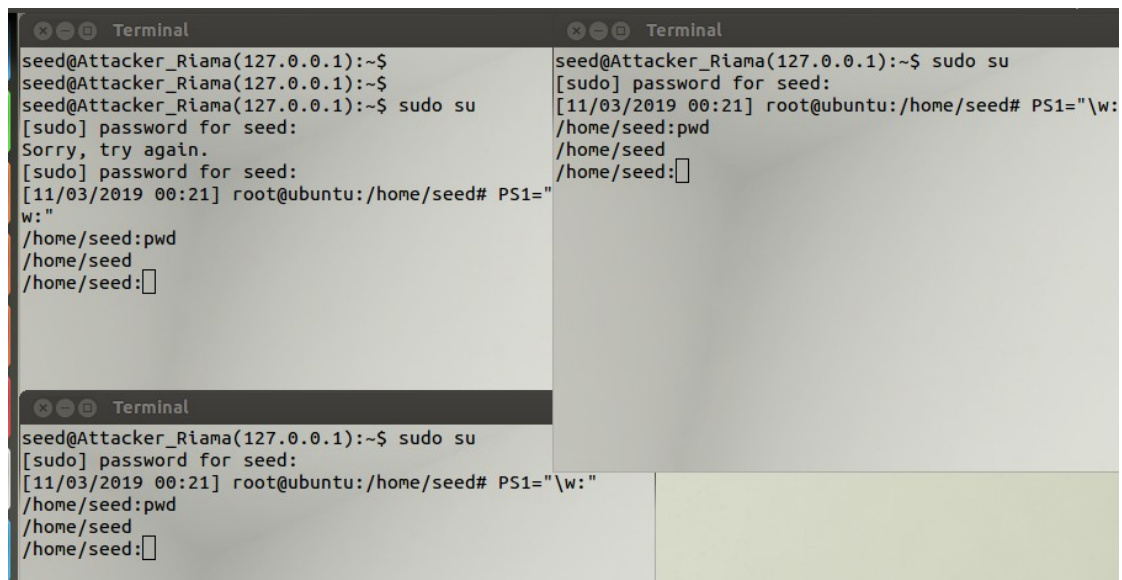
```
seed      2753   1585   0 23:43 tty1      00:00:00 -bash
seed      2875     1   0 23:43 ?           00:00:00 /usr/bin/python /usr/lib/unity
seed      2876     1   0 23:43 ?           00:00:00 /usr/lib/unity-lens-music/uni
seed      2909   2332   0 23:44 ?           00:00:00 /usr/lib/gnome-disk-utility/g
seed      2912   2332   0 23:44 ?           00:00:00 telepathy-indicator
seed      2919     1   0 23:44 ?           00:00:00 /usr/lib/telepathy/mission-co
seed      2924     1   0 23:44 ?           00:00:00 /usr/lib/gnome-online-account
seed      2930   2332   0 23:44 ?           00:00:00 gnome-screensaver
seed      2939   2753   2 23:44 tty1      00:00:11 yes
seed      2941     1   0 23:44 ?           00:00:00 /usr/lib/notify-osd/notify-os
seed      2947   2332   0 23:44 ?           00:00:00 update-notifier
root      2969     1   0 23:44 ?           00:00:00 /usr/bin/python /usr/lib/syste
seed      2983   2753   92 23:45 tty1      00:05:13 yes
seed      2984   2332   0 23:45 ?           00:00:00 /usr/lib/deja-dup/deja-dup/de
root      2992     2   0 23:46 ?           00:00:00 [kworker/0:3]
root      2995     2   0 23:46 ?           00:00:00 [kworker/0:4]
seed      3016    778   0 23:49 tty2      00:00:00 -bash
seed      3114   3016   4 23:49 tty2      00:00:03 yes
seed      3123   3016   0 23:51 tty2      00:00:00 ps -fae
```

\$ kill -9 <NomorPID>

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

Percobaan 6 : Percobaan dengan Penjadwalan Prioritas

1. Login sebagai root.
2. Buka 3 terminal, tampilkan pada screen yang sama.
3. Pada setiap terminal, ketik **PS1 = "\w:"** diikuti **Enter**. **\w** menampilkan path pada direktori home.
4. Karena login sebagai root, maka akan ditampilkan **~:** pada setiap terminal. Untuk setiap terminal ketik **pwd** dan tekan **Enter** untuk melihat bahwa Anda sedang berada pada direktori **/root**.



```
Terminal
seed@Attacker_Riama(127.0.0.1):~$
seed@Attacker_Riama(127.0.0.1):~$
seed@Attacker_Riama(127.0.0.1):~$ sudo su
[sudo] password for seed:
Sorry, try again.
[sudo] password for seed:
[11/03/2019 00:21] root@ubuntu:/home/seed# PS1="\w:"
w:"
/home/seed:pwd
/home/seed
/home/seed:

Terminal
seed@Attacker_Riama(127.0.0.1):~$ sudo su
[sudo] password for seed:
[11/03/2019 00:21] root@ubuntu:/home/seed# PS1="\w:"
/home/seed:pwd
/home/seed
/home/seed:

Terminal
seed@Attacker_Riama(127.0.0.1):~$ sudo su
[sudo] password for seed:
[11/03/2019 00:21] root@ubuntu:/home/seed# PS1="\w:"
/home/seed:pwd
/home/seed
/home/seed:
```

5. Buka terminal lagi (keempat), atur posisi sehingga keempat terminal terlihat pada screen.
6. Pada terminal keempat, ketik **top** dan tekan **Enter**. Maka program top akan muncul. Ketik **i**. **Top** akan menampilkan proses yang aktif. Ketik **lmt**. **Top** tidak lagi menampilkan informasi pada bagian atas dari screen. Pada percobaan ini, terminal ke empat sebagai je ndela **Top**.

```

Terminal
seed@Attacker_Rlama(127.0.0.1):~$
seed@Attacker_Rlama(127.0.0.1):~$
seed@Attacker_Rlama(127.0.0.1):~$ sudo su
[sudo] password for seed:
Sorry, try again.
[sudo] password for seed:
[11/03/2019 00:21] root@ubuntu:/home/seed# PS1="\w:"
w:~
/home/seed:pwd
/home/seed
/home/seed:~

Terminal
seed@Attacker_Rlama(127.0.0.1):~$ sudo su
[sudo] password for seed:
[11/03/2019 00:21] root@ubuntu:/home/seed# PS1="\w:"
/home/seed:pwd
/home/seed
/home/seed:~

Terminal
seed@Attacker_Rlama(127.0.0.1):~$ top
top - 00:23:49 up 29 min, 6 users, load average: 0.24, 0.24, 0.15
top - 00:23:49 up 29 min, 6 users, load average: 0.24, 0.24, 0.15
top - 00:23:49 up 29 min, 6 users, load average: 0.24, 0.24, 0.15
top - 00:23:49 up 29 min, 6 users, load average: 0.24, 0.24, 0.15
top - 00:23:49 up 29 min, 6 users, load average: 0.24, 0.24, 0.15
top - 00:23:49 up 29 min, 6 users, load average: 0.24, 0.24, 0.15
top - 00:23:51 up 29 min, 6 users, load average: 0.22, 0.24, 0.15
Tasks: 173 total, 3 running, 170 sleeping, 0 stopped, 0 zombie
Cpu(s): 50.0%us, 50.0%sy, 0.0%ni, 0.0%id, 0.0%wa, 0.0%hi, 0.0%st,
Mem: 1026952k total, 783728k used, 243224k free, 68988k buffe
Swap: 2094076k total, 0k used, 2094076k free, 364284k cache

  PID USER      PR  NI  VIRT  RES  SHR  S %CPU  %MEM    TIME+  COMMAND
 1043 root        20   0  103m  58m  12m  R 23.2   5.9   0:15.45 Xorg
  462 seed       20   0  144m  12m  9.9m  S  2.0   1.3   0:03.44 metacity
2426 seed       20   0  9744 1444  972  S 23.2   0.1   1:29.84 VBoxClie
    2 root        20   0    0    0    0  S  0.0   0.0   0:00.00 kthreadd
 5127 seed       20   0 99244  23m  10m  R 23.2   2.3   0:01.40 gnome-ter

top - 00:25:31 up 31 min, 6 users, load average: 0.29, 0.26, 0.16
Tasks: 173 total, 1 running, 172 sleeping, 0 stopped, 0 zombie
Cpu(s): 0.0%us, 46.2%sy, 0.0%ni, 53.8%id, 0.0%wa, 0.0%hi, 0.0%st,
Mem: 1026952k total, 785092k used, 241860k free, 69044k buffe
Swap: 2094076k total, 0k used, 2094076k free, 365956k cache

  PID USER      PR  NI  VIRT  RES  SHR  S %CPU  %MEM    TIME+  COMMAND
 6110 seed       20   0  2852 1168  876  R  0.0   0.1   0:00.40 top

  PID USER      PR  NI  VIRT  RES  SHR  S %CPU  %MEM    TIME+  COMMAND
 6110 seed       20   0  2852 1168  876  R  0.3   0.1   0:00.62 top

```

7. Pada terminal 1, bukalah program executable C++ dengan mengetik program **yes** dan tekan **Enter**.
8. Ulangi langkah 7 untuk terminal 2.
9. Jendela **Top** akan menampilkan dua program **yes** sebagai proses yang berjalan. Nilai **%CPU** sama pada keduanya. Hal ini berarti kedua proses mengkonsumsi waktu proses yang sama dan berjalan sama cepat. **PID** dari kedua proses akan berbeda, misalnya 3148 dan 3149. Kemudian gunakan terminal 3 (yang tidak menjalankan **primes** maupun Jendela **Top**) dan ketik **renice 19 <PID terimnal 1>** (contoh : **renice 19 3148**) dan diikuti **Enter**. Hal ini berarti mengganti penjadwalan prioritas dari proses ke 19.

```

D USER      PR  NI  VIRT  RES  SHR  S  %CPU  %MEM    TIME+  COMMAND
7 seed      20   0 99912  23m  10m  R  51.6   2.4    0:59.16  gnome-ter
7 root      20   0  4204   280  228  R  11.3   0.0    0:12.23  yes
8 root      20   0  4204   284  228  R  11.3   0.0    0:11.16  yes
6110 seed   20   0  2852  1168  876  R   0.0   0.1    0:01.25  top

/home/seed:renice 19 6347
6347 (process ID) old priority 0, new priority 19
/home/seed:

```

10. Tunggu beberapa saat sampai program top berubah dan terlihat pada jendela **Top**. Pada kolom STAT memperlihatkan N untuk proses 3148. Hal ini berarti bahwa penjadwalan prioritas untuk proses 3148 lebih besar (lebih lambat) dari 0. Proses 3149 berjalan lebih cepat.
11. Program top juga mempunyai fungsi yang sama dengan program renice. Pilih Jendela **Top** dan tekan **r**. Program top terdapat prompt **PID to renice:** tekan **3148** (ingat bahwa Anda harus mengganti 3148 dengan PID Anda sendiri) dan tekan **Enter**. Program top memberikan prompt **Renice PID 3148 to value:** tekan **-19** dan tekan **Enter**.
12. Tunggu beberapa saat sampai top berubah dan lihat nilai %CPU pada kedua proses. Sekarang proses 3148 lebih cepat dari proses 3149. Kolom status menunjukkan < pada proses 3148 yang menunjukkan penjadwalan prioritas lebih rendah (lebih cepat) dari nilai 0.

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
5127	seed	20	0	99.3m	25m	10m	R	53.9	2.6	9:14.74	gnome-terminal
45	root	20	0	0	0	0	S	15.6	0.0	0:51.59	kworker/0:2
6347	root	1	-19	4204	280	228	S	11.3	0.0	1:53.96	yes
6348	root	20	0	4204	284	228	R	11.0	0.0	1:54.02	yes
1043	root	20	0	103m	58m	13m	S	4.0	5.8	1:03.46	Xorg

13. Pilih terminal 3 (yang sedang tidak menjalankan yes atau program top) dan ketik **nice -n -10 yes** dan tekan **Enter**. Tunggu beberapa saat agar program top berubah dan akan terlihat proses primes ketiga. Misalnya PID nya 4107. Opsi -10 berada pada kolom NI (penjadwalan prioritas).

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
5127	seed	20	0	99.7m	25m	10m	R	57.9	2.6	10:23.30	gnome-terminal
45	root	20	0	0	0	0	S	13.3	0.0	1:03.67	kworker/0:2
6348	root	20	0	4204	284	228	S	11.6	0.0	2:07.89	yes
6347	root	1	-19	4204	280	228	S	6.0	0.0	2:06.19	yes
6469	root	10	-10	4204	280	228	S	5.7	0.0	0:01.72	yes
1043	root	20	0	103m	58m	13m	S	3.3	5.8	1:08.01	Xorg
2426	seed	20	0	9744	1444	972	S	2.7	0.1	9:11.01	VBoxClient
2462	seed	20	0	144m	12m	9.9m	S	0.3	1.3	0:07.23	metacity
6465	root	20	0	2852	1180	876	R	0.3	0.1	0:00.21	top
1	root	20	0	3668	2004	1288	S	0.0	0.2	0:00.44	init

14. Jangan menggunakan mouse dan keyboard selama 10 detik. Program top menampilkan proses yang aktif selain program yes. Maka akan terlihat proses top terdaftar tetapi %CPU kecil (dibawah 1.0) dan konsisten. Juga terlihat proses berhubungan dengan dekstop grafis seperti X, panel dll.

15. Pindahkan mouse sehingga kursor berubah pada screen dan lihat apa yang terjadi dengan tampilan top. Proses tambahan akan muncul dan nilai %CPU berubah sebagai bagian grafis yang bekerja. Satu alasan adalah bahwa proses 4107 berjalan pada penjadwalan prioritas tinggi. Pilih jendela **Top**, ketik **r. PID to renice** : muncul prompt. Ketik **4107** (ubahlah 4107 dengan PID Anda) dan tekan **Enter**. **Renice PID 4107 to value:** muncul prompt. Ketik **0** dan tekan **Enter**. Sekarang pindahkan mouse ke sekeliling screen. Lihat perubahannya.

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
5127	seed	20	0	100m	26m	10m	R	57.0	2.6	13:36.88	gnome-terminal
3512	root	20	0	0	0	0	S	13.2	0.0	2:14.63	kworker/0:0
6348	root	20	0	4204	284	228	S	11.9	0.0	2:46.10	yes
6347	root	1	-19	4204	280	228	S	5.6	0.0	2:25.84	yes
6469	root	10	-10	4204	280	228	S	5.3	0.0	0:21.32	yes
2426	seed	20	0	9744	1444	972	S	3.3	0.1	9:20.40	VBoxClient
1043	root	20	0	104m	58m	13m	S	3.0	5.9	1:18.82	Xorq

16. Tutup semua terminal window.
17. Logout dan login kembali sebagai user.

LATIHAN

1. Masuk ke tty2 dengan **Ctrl+Alt+F2**. Ketik **ps -au** dan tekan **Enter**. Kemudian perhatikan keluaran sebagai berikut :

```
seed@Attacker_Riama(127.0.0.1):~$ ps -au
Warning: bad ps syntax, perhaps a bogus '-'? See http://procps.sf.net/faq.html
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root        771  0.0  0.0   4628   852 tty4      Ss+   Nov02    0:00 /sbin/getty -8
root        786  0.0  0.0   4628   856 tty5      Ss+   Nov02    0:00 /sbin/getty -8
root        799  0.0  0.0   4628   852 tty2      Ss+   Nov02    0:00 /sbin/getty -8
root        800  0.0  0.0   4628   852 tty3      Ss+   Nov02    0:00 /sbin/getty -8
root        806  0.0  0.0   4628   856 tty6      Ss+   Nov02    0:00 /sbin/getty -8
root       1043  1.9  3.6  67660  37760 tty7      Ss+   Nov02    1:22 /usr/bin/X :0 --
root       3028  0.0  0.1   3732   1908 tty1      Ss    Nov02    0:00 /bin/login --
seed       3042  0.0  0.6  10012  6320 tty1      S     Nov02    0:00 -bash
seed       6601  0.0  0.1   4944   1156 tty1      R+    01:04    0:00 ps -au
```

- Sebutkan nama-nama proses yang bukan root
PID 3042 CMD bash, PID 6601 CMD ps -au
- Tulis PID dan COMMAND dari proses yang paling banyak menggunakan CPU time
PID 1043 CMD /usr/bin/X :0 dengan %CPU 1,9
- Sebutkan buyut proses dan PID dari proses tersebut
PID 1043 CMD /usr/bin/X :0 --
- Sebutkan beberapa proses daemon
- Pada prompt login lakukan hal- hal sebagai berikut :
\$ csh

\$ who

\$ bash

\$ ls

\$ sh

\$ ps

```
seed@Attacker_Riama(127.0.0.1): $ csh
% who
seed      tty2      2019-11-03 20:43
seed      tty7      2019-11-03 20:43
% bash
seed@Attacker_Riama(127.0.0.1):~$ ls
a.out          kelas2.txt      passwd_input
attack_process  kelas.txt       Pictures
attack_process.c Music           Public
baru           mydir           rmdirerror.txt
baru.urut      myerror.txt     satu.txt
coba           myfile.txt      surat
cow_attack.c   mylife          target_process.sh
Desktop        openssl-1.0.1    Templates
Documents      openssl_1.0.1-4ubuntu5.11.debian.tar.gz Videos
Downloads      openssl_1.0.1-4ubuntu5.11.dsc      vulp
elggData       openssl_1.0.1.orig.tar.gz          vulp2.c
examples.desktop out              vulp.c
hello.txt      output
kelas1.txt     out.txt
seed@Attacker_Riama(127.0.0.1):~$ sh
$ ps
  PID TTY          TIME CMD
 2833 tty2      00:00:00 bash
 4259 tty2      00:00:00 csh
 4300 tty2      00:00:00 bash
 4355 tty2      00:00:00 sh
 4358 tty2      00:00:00 ps
```

Analisis : Perintah csh digunakan untuk membuat shell, perintah who untuk menampilkan user yang sedang aktif, bash untuk menampilkan file pada direktori, ls menampilkan file yang terdapat dalam direktori, sh untuk memasuki shell, dan ps untuk menampilkan kondisi proses.

- f. Sebutkan PID yang paling besar dan kemudian buat urutan proses sampai ke PPID = 1.

UID	PID	PPID	C	STIME	TTY	TIME	CMD
seed	2833	776	0	20:43	tty2	00:00:00	-bash
seed	4259	2833	0	20:47	tty2	00:00:00	-sh
seed	4300	4259	0	20:47	tty2	00:00:00	bash
seed	4355	4300	0	20:47	tty2	00:00:00	sh
seed	4359	4355	0	20:48	tty2	00:00:00	bash
seed	4425	4359	0	20:53	tty2	00:00:00	ps -f

2. Cobalah format tampilan ps dengan opsi berikut dan perhatikan hasil tampilannya :

-f daftar penuh

UID	PID	PPID	C	STIME	TTY	TIME	CMD
seed	2833	776	0	20:43	tty2	00:00:00	-bash
seed	4259	2833	0	20:47	tty2	00:00:00	-sh
seed	4300	4259	0	20:47	tty2	00:00:00	bash
seed	4355	4300	0	20:47	tty2	00:00:00	sh
seed	4359	4355	0	20:48	tty2	00:00:00	bash
seed	4425	4359	0	20:53	tty2	00:00:00	ps -f

-j format job

```
seed@Attacker_Riama(127.0.0.1):~$ ps -j
```

PID	PGID	SID	TTY	TIME	CMD
2833	2833	776	tty2	00:00:00	bash
4259	4259	776	tty2	00:00:00	csh
4300	4300	776	tty2	00:00:00	bash
4355	4355	776	tty2	00:00:00	sh
4359	4359	776	tty2	00:00:00	bash
4432	4432	776	tty2	00:00:00	ps

j format job control

```
seed@Attacker_Riama(127.0.0.1):~$ ps j
```

PPID	PID	PGID	SID	TTY	TPGID	STAT	UID	TIME	COMMAND
776	2833	2833	776	tty2	4437	S	1000	0:00	-bash
2833	4259	4259	776	tty2	4437	S	1000	0:00	-sh
4259	4300	4300	776	tty2	4437	S	1000	0:00	bash
4300	4355	4355	776	tty2	4437	S	1000	0:00	sh
4355	4359	4359	776	tty2	4437	S	1000	0:00	bash
4359	4437	4437	776	tty2	4437	R+	1000	0:00	ps j

l daftar memanjang

```
seed@Attacker_Riama(127.0.0.1):~$ ps l
```

F	UID	PID	PPID	PRI	NI	USZ	RSS	WCHAN	STAT	TTY	TIME	COMMAND
4	1000	2833	776	20	0	10012	6400	wait	S	tty2	0:00	-bash
0	1000	4259	2833	20	0	4384	568	sigsus	S	tty2	0:00	-sh
0	1000	4300	4259	20	0	7320	3704	wait	S	tty2	0:00	bash
0	1000	4355	4300	20	0	2232	548	wait	S	tty2	0:00	sh
0	1000	4359	4355	20	0	7320	3700	wait	S	tty2	0:00	bash
0	1000	4438	4359	20	0	4704	704	-	R+	tty2	0:00	ps l

s format sinyal

```
seed@Attacker_Riama(127.0.0.1):~$ ps s
```

UID	PID	PENDING	BLOCKED	IGNORED	CAUGHT	STAT	TTY	TIME	COMMAND
1000	2833	00000000	00010000	00384004	4b813efb	S	tty2	0:00	-bash
1000	4259	00000000	00000002	00384004	01810003	S	tty2	0:00	-sh
1000	4300	00000000	00010000	00384004	4b813efb	S	tty2	0:00	bash
1000	4355	00000000	00000000	00284004	00010002	S	tty2	0:00	sh
1000	4359	00000000	00010000	00384004	4b813efb	S	tty2	0:00	bash
1000	4439	00000000	00000000	00000000	73d3fef9	R+	tty2	0:00	ps s

v format virtual memory

```
seed@Attacker_Riama(127.0.0.1):~$ ps v
```

PID	TTY	STAT	TIME	MAJFL	TRS	DRS	RSS	%MEM	COMMAND
2833	tty2	S	0:00	1	877	9134	6400	0.6	-bash
4259	tty2	S	0:00	1	128	4255	568	0.0	-sh
4300	tty2	S	0:00	0	877	6442	3704	0.3	bash
4355	tty2	S	0:00	0	92	2139	548	0.0	sh
4359	tty2	S	0:00	0	877	6442	3700	0.3	bash
4443	tty2	R+	0:00	0	82	4621	704	0.0	ps v

X format register i386

```
seed@Attacker_Riama(127.0.0.1):~$ ps X
```

PID	STACKP	ESP	EIP	TMOUT	ALARM	STAT	TTY	TIME	COMMAND
2833	bf8b5550	bf8b5088	b775e424	-	-	S	tty2	0:00	-bash
4259	bfac7670	bfac6c70	b7776424	-	-	S	tty2	0:00	-sh
4300	bfe088a0	bfe083d8	b7775424	-	-	S	tty2	0:00	bash
4355	bff3c610	bff3c1e4	b7758424	-	-	S	tty2	0:00	sh
4359	bfa80720	bfa80258	b7702424	-	-	S	tty2	0:00	bash
4444	bfde2170	bfde1af8	b771f424	-	-	R+	tty2	0:00	ps X

3. Lakukan urutan pekerjaan berikut :

- Gunakan perintah find ke seluruh direktori pada sistem, blokir output sehingga daftar direktori dialihkan ke file directories.txt dan daftar pesan error dialihkan ke file errors.txt
- Gunakan perintah sleep 5. Apa yang terjadi dengan perintah ini ?

```
seed@Attacker_Riama(127.0.0.1):~$ sleep 5
seed@Attacker_Riama(127.0.0.1):~$ sleep 5 &
[1] 4497
```

- Jalankan perintah pada *background* menggunakan &

```
seed@Attacker_Riama(127.0.0.1):~$ bg &
[2] 4499
[1] Done sleep 5
```


- d. Jalankan sleep 15 pada *foreground* , hentikan sementara dengan Ctrl- Z dan kemudian letakkan pada *background* dengan bg. Ketikkan jobs. Ketikkan ps. Kembalikan job ke *foreground* dengan perintah fg.

```
seed@Attacker_Riama(127.0.0.1):~$ sleep 15
^Z[1] Done sleep 15

[2]+ Stopped sleep 15
seed@Attacker_Riama(127.0.0.1):~$ bg
[2]+ sleep 15 &
```

- e. Jalankan sleep 15 pada *background* menggunakan & dan kemudian gunakan perintah kill untuk menghentikan proses diikuti *job number*.

```
seed@Attacker_Riama(127.0.0.1):~$ sleep 15 &
[3] 4507
[2] Done sleep 15
seed@Attacker_Riama(127.0.0.1):~$ kill 4507
```

- f. Jalankan sleep 15 pada *background* menggunakan & dan kemudian gunakan kill untuk menghentikan sementara proses. Gunakan bg untuk melanjutkan menjalankan proses.

```
seed@Attacker_Riama(127.0.0.1):~$ bg
bash: bg: job 1 already in background
seed@Attacker_Riama(127.0.0.1):~$ sleep 15
^Z
[2]+ Stopped sleep 15
seed@Attacker_Riama(127.0.0.1):~$ bg
[2]+ sleep 15 &
```

- g. Jalankan sleep 60 pada *background* 5 kali dan terminasi semua pada dengan menggunakan perintah killall.

```
seed@Attacker_Riama(127.0.0.1):~$ sleep 60 &
[3] 4521
[1] Done sleep 15
[2] Done sleep 15
seed@Attacker_Riama(127.0.0.1):~$ sleep 60 &
[4] 4522
seed@Attacker_Riama(127.0.0.1):~$ sleep 60 &
[5] 4523
seed@Attacker_Riama(127.0.0.1):~$ sleep 60 &
[6] 4525
seed@Attacker_Riama(127.0.0.1):~$ sleep 60 &
[7] 4526
seed@Attacker_Riama(127.0.0.1):~$ killall sleep
[3] Terminated sleep 60
[4] Terminated sleep 60
[5] Terminated sleep 60
[6]- Terminated sleep 60
[7]+ Terminated sleep 60
```

- h. Gunakan perintah ps, w dan top untuk menunjukkan semua proses yang sedang dieksekusi.

```
[1]+ Stopped top
seed@Attacker_Riama(127.0.0.1):~$ ps
  PID TTY          TIME CMD
 2833 tty2      00:00:00 bash
 4259 tty2      00:00:00 csh
 4300 tty2      00:00:00 bash
 4355 tty2      00:00:00 sh
 4359 tty2      00:00:00 bash
 4532 tty2      00:00:00 top
 4533 tty2      00:00:00 ps
seed@Attacker_Riama(127.0.0.1):~$ w
 21:19:07 up 36 min,  2 users,  load average: 0.00, 0.01, 0.06
USER      TTY      FROM              LOGIN@   IDLE   JCPU   PCPU   WHAT
seed      tty2                    20:43    3.00s  0.97s  0.00s  w
seed      tty7                    20:43   35:58  7.07s  0.16s  gnome-session -
```

```

top - 21:20:22 up 37 min, 2 users, load average: 0.00, 0.01, 0.05
Tasks: 164 total, 1 running, 162 sleeping, 1 stopped, 0 zombie
Cpu(s):  0.0%us,  0.0%sy,  0.0%ni,100.0%id,  0.0%wa,  0.0%hi,  0.0%si,  0.0%st
Mem:   1026952k total,  951284k used,    75668k free,   156796k buffers
Swap:  2094076k total,    0k used,   2094076k free,   454380k cached

```

PID	USER	PR	NI	UIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
4536	seed	20	0	2852	1172	880	R	0.7	0.1	0:00.04	top
20	root	20	0	0	0	0	S	0.3	0.0	0:03.78	kworker/0:1
347	messageb	20	0	4188	1968	836	S	0.3	0.2	0:01.63	dbus-daemon
997	root	20	0	38972	8548	4612	S	0.3	0.8	0:00.21	apache2
1292	root	20	0	9844	1308	956	S	0.3	0.1	0:01.10	UBoxService
2426	seed	20	0	9744	1392	928	S	0.3	0.1	0:07.85	UBoxClient
2472	seed	20	0	255m	50m	29m	S	0.3	5.1	0:03.59	unity-2d-shell
1	root	20	0	3652	2052	1288	S	0.0	0.2	0:00.45	init
2	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kthreadd
3	root	20	0	0	0	0	S	0.0	0.0	0:00.65	ksoftirqd/0
5	root	20	0	0	0	0	S	0.0	0.0	0:00.31	kworker/u:0
6	root	RT	0	0	0	0	S	0.0	0.0	0:00.00	migration/0
7	root	RT	0	0	0	0	S	0.0	0.0	0:00.03	watchdog/0
8	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	cpuset
9	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	khelper
10	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kdeutmpfs
11	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	netns

- i. Gunakan perintah `ps -aeH` untuk menampilkan hierarki proses. Carilah `init` proses.

Apakah Anda bisa identifikasi sistem daemon yang penting ?

Dapatkan Anda identifikasi shell dan subproses ?

2665	?	00:00:00	gufsd-burn
2673	?	00:00:00	gufsd-metadata
2689	?	00:00:14	hud-service
2695	?	00:00:00	unity-lens-vid
2696	?	00:00:00	unity-music-dae
2697	?	00:00:00	unity-files-dae
2698	?	00:00:00	unity-applicati
2733	?	00:00:00	zeitgeist-daemo
2739	?	00:00:00	zeitgeist-fts
2743	?	00:00:00	cat
2741	?	00:00:00	zeitgeist-datah
2770	?	00:00:00	unity-scope-vid
2771	?	00:00:00	unity-musicstor
2815	?	00:00:00	mission-control
2819	?	00:00:00	goa-daemon
2944	?	00:00:01	notify-osd
2976	?	00:00:00	system-service-
3163	?	00:00:00	at-spi-bus-laun
4269	?	00:00:04	update-manager

- j. Kombinasikan `ps -fae` dan `grep`, apa yang Anda lihat ?
- k. Jalankan proses `sleep 300` pada *background*. Log off komputer dan log in kembali. Lihat daftar semua proses yang berjalan. Apa yang terjadi pada proses `sleep` ?

```

seed@Attacker_Riama(127.0.0.1):~$ ps -fae | grep
Usage: grep [OPTION]... PATTERN [FILE]...
Try 'grep --help' for more information.
seed@Attacker_Riama(127.0.0.1):~$ sleep 300 &
[31] 4542

```

Analisis :

Perintah `bsleep` digunakan untuk mengentikan proses job pada terminal selama `n` detik. Perintah `sleep` dengan tambahan “&” digunakan untuk menjalankan kembali job. Perintah `kill` digunakan untuk menghentikan proses pada job. Perintah `killall sleep` digunakan untuk menghentikan semua job. Perintah `w` digunakan untuk menunjukkan user aktif. Perintah `top` digunakan untuk menunjukkan semua proses yang dieksekusi.

LAPORAN RESMI

1. Analisa hasil percobaan yang Anda lakukan.
2. Kerjakan latihan diatas dan analisa hasil tampilannya.
3. Berikan kesimpulan dari praktikum ini. Kaitkan dengan definisi Deadlock. Percobaan mana yang dapat digunakan untuk mencegah deadlock? Kaitkan dengan definisi dan penyebab deadlock