

LAPORAN RESMI

PRAKTIKUM 5

“ Proses Dan Management Proses 2 ”



NAMA : FERI AFRIANTO

KELAS : D4 Teknik Informatika A

NRP : 2110191007

Politeknik Elektronika Negeri Surabaya

PRAKTIKUM 5

“Proses Dan Management Proses 2”

1. PERCOBAAN

1. Percobaan 1 : Menghentikan dan Memulai Kembali Job.

1) Perintah \$ yes > /dev /null

Analisis : Perintah \$ yes > /dev /null di pergunakan untuk memulai job, job dapat dihentikan sementara atau sering di sebut (suspend) dengan menggunakan Ctrl – Z.

2) Perintah fg.

Analisis : Perintah fg digunakan untuk memulai kembali job yang telah disuspend pada foreground.

3) Perintah \$bg

Analisis : Perintah ini berfungsi untuk mensuspend job yes >/dev/null. Kemudian memindahkannya ke background proses, yang berarti proses atau job tersebut tidak berhenti, akan tetapi terus berjalan di belakang layar. Job yang berada pada background tidak bisa langsung dihentikan dengan Ctrl + Z. Untuk menghentikannya kita harus memindahkan job tersebut ke foreground lalu mensuspendnya.

4) Perintah \$ yes &

Analisis : Perintah \$ yes & ini berjalan pada job background yang kita buat tadi. Kita dapat menghentikan prosesnya dengan cara kita mengetik fg lalu enter agar proses pindah ke foreground, kemudian dilanjutkan menekan Ctrl + Z agar berhenti sementara.

5) Perintah \$ fg %2 dan \$bg %2

Analisis : Perintah ini berguna untuk menjalankan banyak job dalam waktu bersamaan. Job – job yang sedang tidak dipakai dimasukkan ke proses background, dan untuk mengenalinya kita menggunakan nomor job ID karena tidak ada job yang memiliki nomor ID sama.

6) Tekan fg dan tekan Enter, kemudian dilanjutkan dengan Ctrl -Z untuk menghentikan sementara.

Analisis : Pada saat kita menekan fg, muncul hasil dari perintah \$yes & tadi. Jadi fungsi perintah fg disini adalah untuk memanggil proses yang berjalan di background. Dan kita bisa menghentikannya dengan menekan Ctrl + Z.

7) Perintah \$ ps -fae dan \$ kill -9 <nomor PID>

Analisis : Perintah \$ ps -fae digunakan untuk menampilkan secara lengkap seluruh proses yang sedang berjalan beserta detailnya termasuk proses dari perintah \$ps – fae sendiri. Sedangkan perintah \$ kill -9 <nomor PID> digunakan untuk menghentikan atau terminate suatu proses, berdasarkan nomor PID proses yang ingin dihentikan.

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

Analisis : Disini kita disuruh logout lalu kembali ke mode grafis. Untuk logout kita bisa menekan exit dan untuk kembali ke mode grafis kita bisa menekan tombol Alt + F7.

2. Percobaan 2 : Percobaan dengan Penjadwalan Prioritas.

1) Login sebagai root.

Analisis : Untuk login sebagai root, kita bisa menggunakan perintah `$sudo su` lalu enter. Kemudian masukkan password dan tekan enter.

2) Buka 3 terminal, tampilkan pada screen yang sama.

Analisis : Disini kita disuruh membuka 3 terminal kemudian menampilkannya pada screen yang sama. Caranya tinggal kecilkan satu per satu jendela terminal sehingga terlihat semua.

3) Pada setiap terminal, ketik `PS1 = "\w:"` diikuti Enter. `\w` menampilkan path pada direktori home.

Analisis : Perintah `$ PS1="\w:"` diatas bertujuan untuk masuk ke directory home dari user root.

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`.

Analisis : Bila posisi kita masih berada di `/home/<user>`, maka kita bisa gunakan perintah `$cd ~` untuk masuk ke root. Dan untuk mengeceknya kita gunakan perintah `$pwd`.

5) Buka terminal lagi (keempat), atur posisi sehingga keempat terminal terlihat pada screen.

Analisis : Setelah kita membuka terminal ke empat, kecilkan lagi ukurannya agar posisinya terlihat semua di layar.

6) Perintah `top`

Analisis : Perintah `top` disini digunakan untuk mengetahui semua rincian proses yang berjalan, dan beberapa fungsi lainnya. Mengetikkan `'i'` pada window `top` akan menampilkan proses yang sedang aktif. Mengetikkan `'lmt'` untuk menghilangkan atau tidak menampilkan informasi pada bagian atas dari tampilan `top`.

7) Perintah `$ yes`

Analisis : Karena pada percobaan ini menggunakan program sendiri yang perintah eksekusinya adalah `./a.out`, maka yang muncul adalah angka ganjil antara angka 1 sampai 10, dan itu diulang-ulang terus tidak berhenti.

8) Ulangi langkah 7 untuk terminal 2.

Analisis : Disini kita disuruh mengulangi langkah nomor 7 namun pada terminal kedua. Dan hasilnya sama yaitu angka ganjil yang diulang terus menerus. Dan mulai dari sini OS Linux yang kita gunakan menjadi lemot.

- 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 terminal 1> (contoh : renice 19 3148) dan diikuti Enter. Hal ini berarti mengganti penjadwalan prioritas dari proses ke 19.**

Analisis : Disini PID dua program yang telah kita jalankan tadi adalah 2356 dan 2357. Dua proses yang sama akan menggunakan sumber daya yang sama (mendekati), misalkan dua program yang memiliki PID 2356 dan 2357, keduanya menggunakan sumber daya yang hampir sama besarnya, namun PID dari kedua proses akan berbeda. Perintah \$ renice 19 2356 berarti mengganti penjadwalan prioritas dari proses ke 19, dan NI berubah yang tadinya 0 menjadi 19.

- 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.**

Analisis : Setelah prioritasnya diubah menjadi 19, proses 2356 berjalan lebih lambat dari proses 2357. Ini disebabkan proses 2356 mendapatkan penjadwalan prioritas lebih besar dari proses 2357.

- 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 dan tekan Enter. Program top memberikan prompt Renice PID 3148 to value: tekan -19 dan tekan Enter.**

Analisis : Program top juga memiliki fungsi yang sama dengan program renice. Jadi kita dapat mengubah konfigurasi proses melalui program renice maupun top.

- 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.**

Analisis : Setelah prioritas proses 2356 diubah menjadi -19, proses 2356 menjadi lebih cepat daripada 2357.

- 13) **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).**

Analisis : Disini kita menggunakan program sendiri. Perintah \$ nice -n -10 ./a.out digunakan untuk membuat proses baru, dan opsi -10 merupakan penentuan prioritas dari proses tersebut. PID dari proses yang kita buat adalah 2381, dan disana terlihat pada kolom NI terdapat angka -10.

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.

Analisis : Saat mouse diam muncul beberapa proses lain, namun dengan penggunaan %CPU kecil (penggunaannya memang di bawah 1.0). Dan itu juga konsisten tidak berubah-ubah.

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 4655 to value: muncul prompt. Ketik 0 dan tekan Enter. Sekarang pindahkan mouse ke sekeliling screen. Lihat perubahannya.

Analisis : Saat kita memindahkan kursor mouse, beberapa proses yang muncul tadi penggunaan %CPU nya berubah semua yaitu cenderung bertambah. Dan juga muncul lagi proses lain dengan penggunaan %CPU lumayan banyak.

Kita dapat mengubah konfigurasi suatu proses melalui perintah – perintah yang terdapat pada proses top. Dan ternyata setelah proses 2381 diubah prioritasnya menjadi 0, penggunaan sumber daya oleh terminal top menjadi lebih stabil (tidak banyak perubahan) walaupun mouse digerakkan ke sekeliling screen.

16)Tutup semua terminal window.

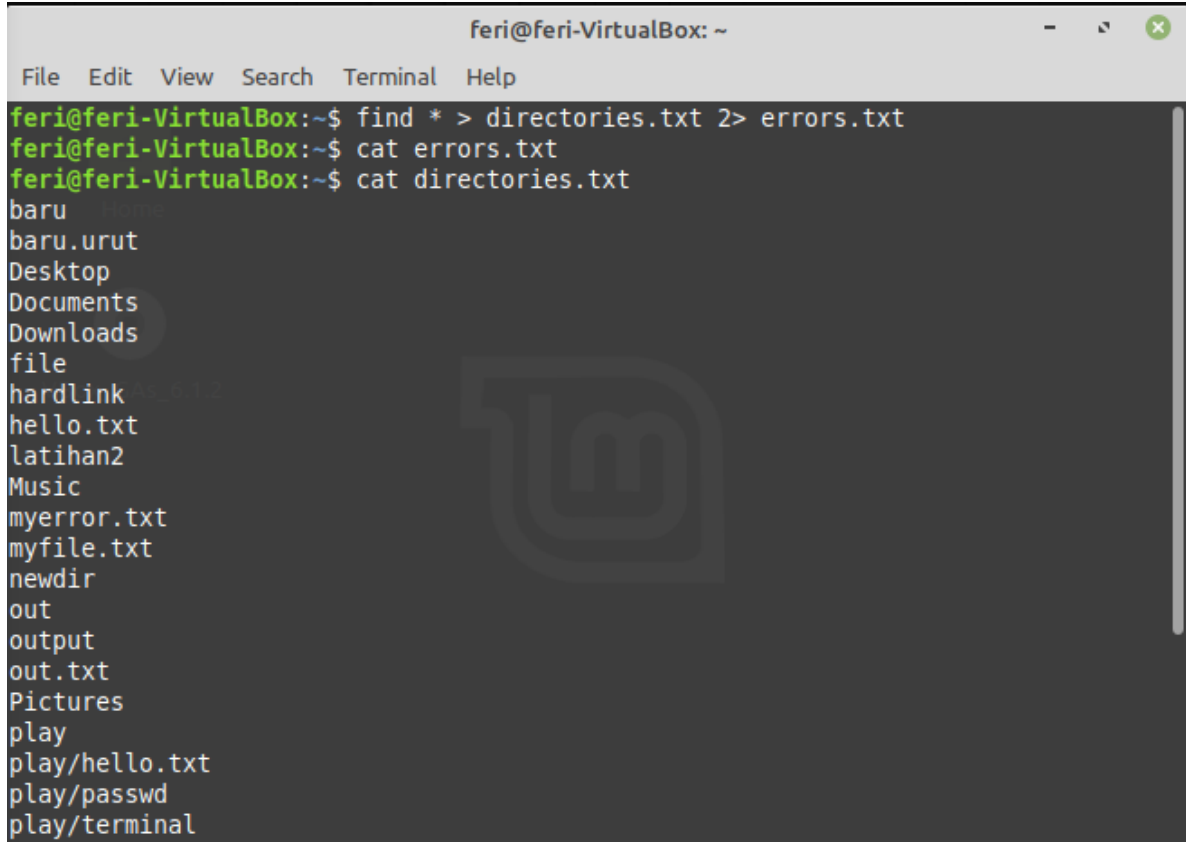
Analisis : Disini kita disuruh menutup semua window terminal. Dan hasilnya setelah semua terminal tertutup, linux menjadi tidak lemot lagi.

17)Logout dan login kembali sebagai user.

Analisis : Saat kita menutup terminal, otomatis juga kita logout. Dan saat kita membuka kembali terminal, otomatis kita langsung login sebagai user biasa.

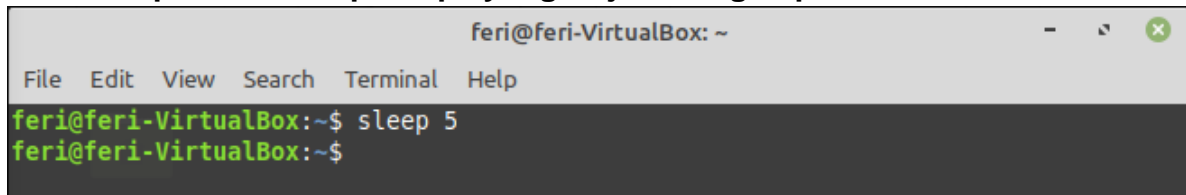
2. Latihan

1. Gunakan perintah `find` ke seluruh direktori pada system, belokkan output sehingga daftar direktori dialihkan ke file `directories.txt` dan daftar pesan error dialihkan ke file `errors.txt`.



```
feri@feri-VirtualBox: ~  
File Edit View Search Terminal Help  
feri@feri-VirtualBox:~$ find * > directories.txt 2> errors.txt  
feri@feri-VirtualBox:~$ cat errors.txt  
feri@feri-VirtualBox:~$ cat directories.txt  
baru  
baru.urut  
Desktop  
Documents  
Downloads  
file  
hardlink  
hello.txt  
latihan2  
Music  
myerror.txt  
myfile.txt  
newdir  
out  
output  
out.txt  
Pictures  
play  
play/hello.txt  
play/passwd  
play/terminal
```

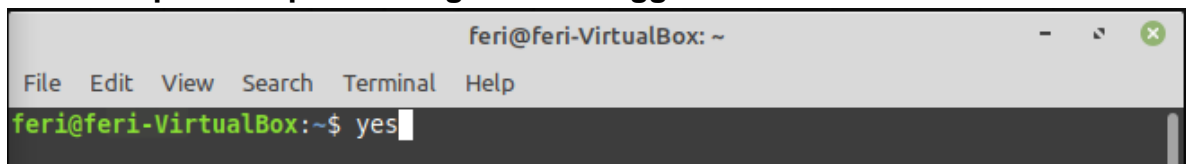
2. Gunakan perintah `sleep 5`. Apa yang terjadi dengan perintah ini ?



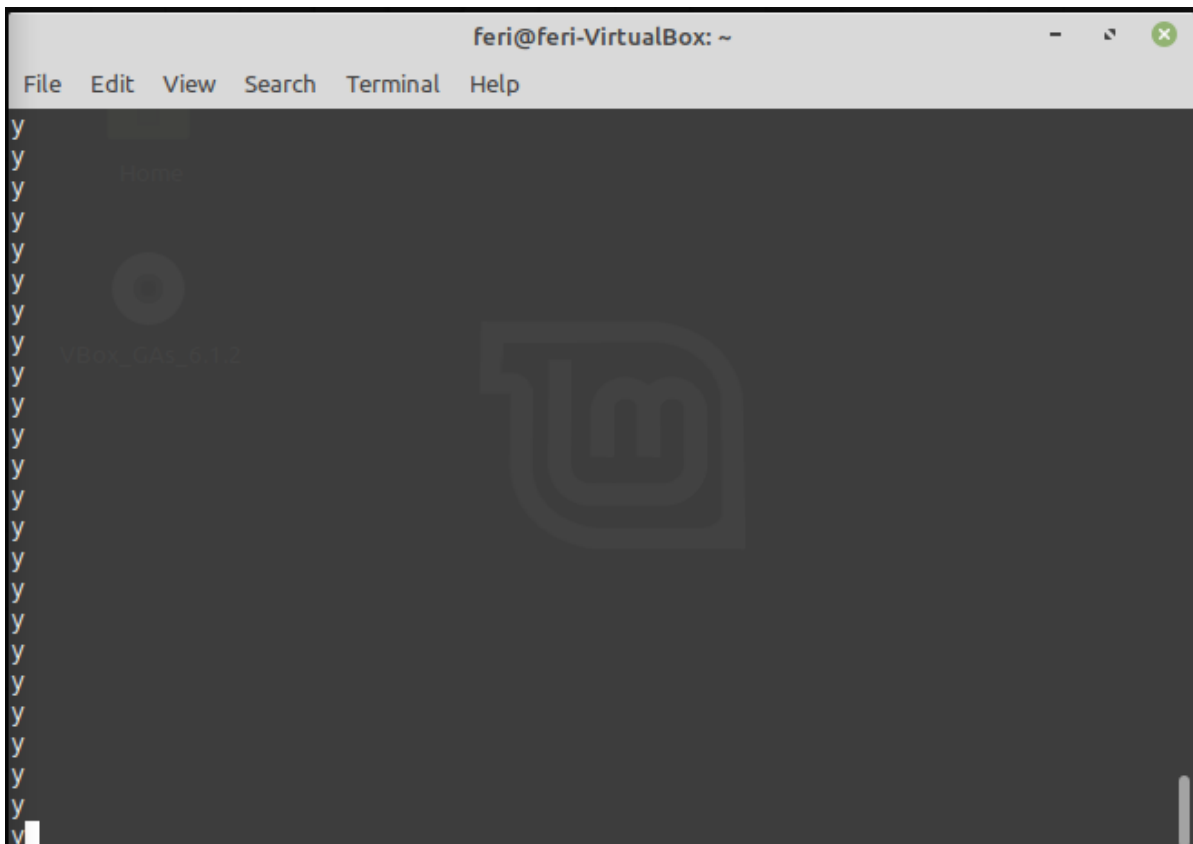
```
feri@feri-VirtualBox: ~  
File Edit View Search Terminal Help  
feri@feri-VirtualBox:~$ sleep 5  
feri@feri-VirtualBox:~$
```

`Sleep 5` menunjukkan perintah untuk idle sleep terminal hingga 5 detik.

3. Jalankan perintah pada background menggunakan `&`



```
feri@feri-VirtualBox: ~  
File Edit View Search Terminal Help  
feri@feri-VirtualBox:~$ yes
```



4. 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.

```
feri@feri-VirtualBox: ~
File Edit View Search Terminal Help

feri@feri-VirtualBox:~$ sleep 15
^Z
[1]+  Stopped                  sleep 15
feri@feri-VirtualBox:~$ fg
sleep 15
^Z
[1]+  Stopped                  sleep 15
feri@feri-VirtualBox:~$ bg
[1]+ sleep 15 &
feri@feri-VirtualBox:~$ jobs
[1]+  Done                     sleep 15
feri@feri-VirtualBox:~$ ps
  PID TTY          TIME CMD
 2054 pts/0    00:00:00 bash
 2063 pts/0    00:00:00 ps
feri@feri-VirtualBox:~$ fg
bash: fg: current: no such job
feri@feri-VirtualBox:~$
```

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

```
feri@feri-VirtualBox: ~  
File Edit View Search Terminal Help  
feri@feri-VirtualBox:~$ sleep 15&  
[1] 2065  
feri@feri-VirtualBox:~$ jobs  
[1]+  Running                  sleep 15 &  
feri@feri-VirtualBox:~$ kill %1  
bash: kill: (2065) - No such process  
[1]+  Done                      sleep 15  
feri@feri-VirtualBox:~$ jobs  
feri@feri-VirtualBox:~$
```

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

```
feri@feri-VirtualBox: ~  
File Edit View Search Terminal Help  
feri@feri-VirtualBox:~$ sleep 15&  
[1] 2081  
feri@feri-VirtualBox:~$ kill -STOP 2081  
feri@feri-VirtualBox:~$ bg  
[1]+ sleep 15 &  
feri@feri-VirtualBox:~$
```


7. Jalankan sleep 60 pada background 5 kali dan terminasi semua pada dengan menggunakan perintah killall.

```
feri@feri-VirtualBox: ~  
File Edit View Search Terminal Help  
feri@feri-VirtualBox:~$ sleep 60  
^Z  
[1]+  Stopped                  sleep 60  
feri@feri-VirtualBox:~$ sleep 60  
^Z  
[2]+  Stopped                  sleep 60  
feri@feri-VirtualBox:~$ sleep 60  
^Z  
[3]+  Stopped                  sleep 60  
feri@feri-VirtualBox:~$ sleep 60  
^Z  
[4]+  Stopped                  sleep 60  
feri@feri-VirtualBox:~$ sleep 60  
^Z  
[5]+  Stopped                  sleep 60  
feri@feri-VirtualBox:~$ jobs  
[1]  Stopped                  sleep 60  
[2]  Stopped                  sleep 60  
[3]  Stopped                  sleep 60  
[4]- Stopped                  sleep 60  
[5]+ Stopped                  sleep 60  
feri@feri-VirtualBox:~$ killall -9 sleep  
[1]  Killed                  sleep 60  
[2]  Killed                  sleep 60  
[3]  Killed                  sleep 60  
[4]- Killed                  sleep 60  
[5]+ Killed                  sleep 60  
feri@feri-VirtualBox:~$
```

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

```
feri@feri-VirtualBox: ~  
File Edit View Search Terminal Help  
feri@feri-VirtualBox:~$ ps  
  PID TTY          TIME CMD  
 2054 pts/0    00:00:00 bash  
 2094 pts/0    00:00:00 ps  
feri@feri-VirtualBox:~$ w  
22:09:17 up 14 min,  1 user,  load average: 0,26, 0,91, 0,86  
USER      TTY      FROM            LOGIN@   IDLE   JCPU   PCPU WHAT  
feri      tty7     :0               21:56   13:48  14.07s  0.66s cinnamon-sessio  
feri@feri-VirtualBox:~$
```

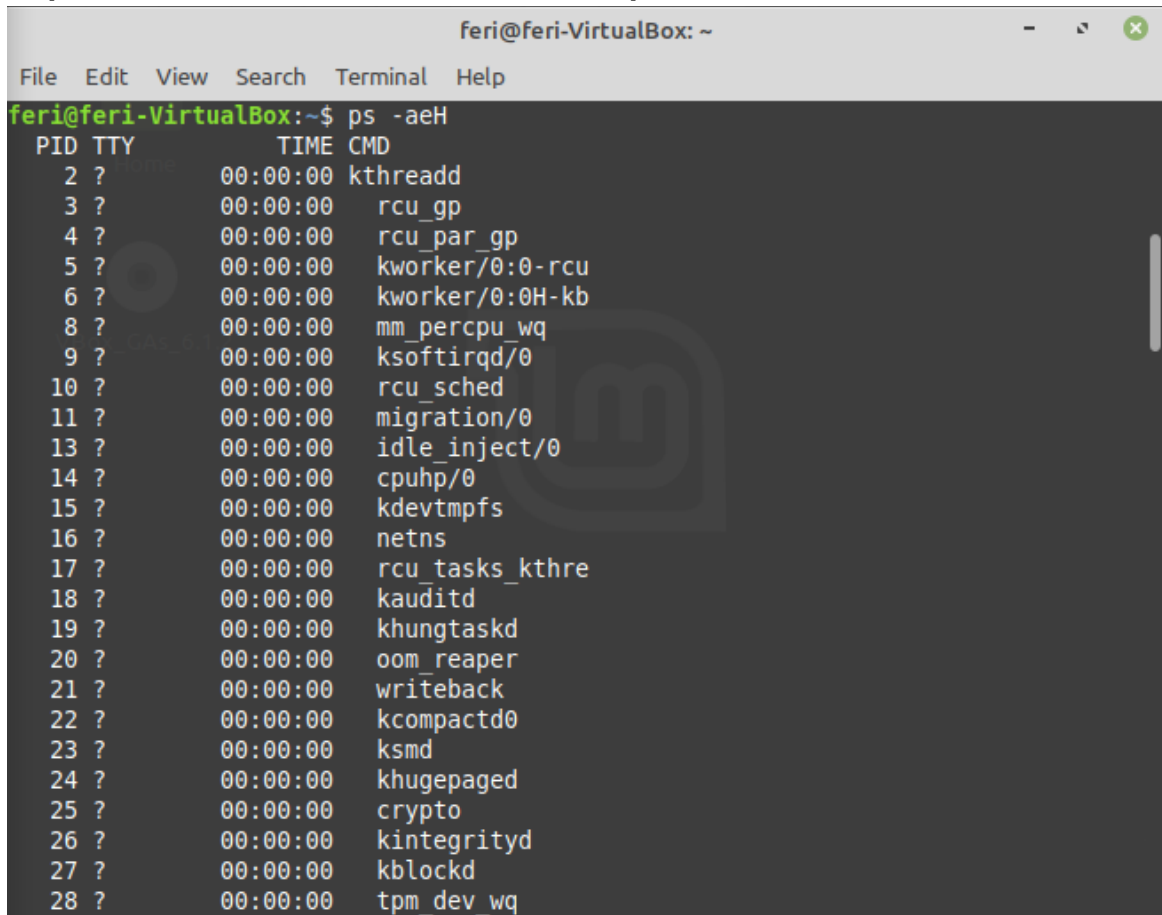
```

top - 22:09:48 up 14 min,  1 user,  load average: 0,32, 0,86, 0,84
Tasks: 152 total,   2 running, 119 sleeping,   0 stopped,   0 zombie
%Cpu(s):  4,7 us,   0,7 sy,   0,0 ni, 94,6 id,   0,0 wa,   0,0 hi,   0,0 si,   0,0 st
KiB Mem : 1006576 total,  102340 free,  540228 used,  364008 buff/cache
KiB Swap: 1459804 total, 1445468 free,   14336 used.  292808 avail Mem

```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1516	feri	20	0	2534640	149092	77144	S	4,0	14,8	2:12.72	cinnamon
1107	root	20	0	390884	62232	32028	S	0,7	6,2	0:14.29	Xorg
2047	feri	20	0	589364	35220	25912	S	0,7	3,5	0:02.17	gnome-termi+
2096	feri	20	0	41776	3820	3228	R	0,7	0,4	0:00.04	top
1	root	20	0	159924	7852	6296	S	0,0	0,8	0:03.28	systemd
2	root	20	0	0	0	0	S	0,0	0,0	0:00.00	kthreadd
3	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	rcu_gp
4	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	rcu_par_gp
5	root	20	0	0	0	0	I	0,0	0,0	0:00.01	kworker/0:0+
6	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	kworker/0:0+
8	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	mm_percpu_wq
9	root	20	0	0	0	0	S	0,0	0,0	0:00.71	ksoftirqd/0
10	root	20	0	0	0	0	R	0,0	0,0	0:00.78	rcu_sched
11	root	rt	0	0	0	0	S	0,0	0,0	0:00.00	migration/0
13	root	-51	0	0	0	0	S	0,0	0,0	0:00.00	idle_inject+
14	root	20	0	0	0	0	S	0,0	0,0	0:00.00	cpuhp/0
15	root	20	0	0	0	0	S	0,0	0,0	0:00.00	kdevtmpfs
16	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	netns
17	root	20	0	0	0	0	S	0,0	0,0	0:00.00	rcu_tasks_k+
18	root	20	0	0	0	0	S	0,0	0,0	0:00.00	kauditd
19	root	20	0	0	0	0	S	0,0	0,0	0:00.00	khunotaskd

9. Gunakan perintah `ps -aeH` untuk menampilkan hierarki proses. Carilah init proses. Apakah Anda bisa identifikasi sistem daemon yang penting ?
Dapatkan Anda identifikasi shell dan subprocess ?



```
feri@feri-VirtualBox: ~$ ps -aeH
PID TTY          TIME CMD
  2 ?           00:00:00 kthreadd
  3 ?           00:00:00 rcu_gp
  4 ?           00:00:00 rcu_par_gp
  5 ?           00:00:00 kworker/0:0-rcu
  6 ?           00:00:00 kworker/0:0H-kb
  8 ?           00:00:00 mm_percpu_wq
  9 ?           00:00:00 ksoftirqd/0
 10 ?           00:00:00 rcu_sched
 11 ?           00:00:00 migration/0
 13 ?           00:00:00 idle_inject/0
 14 ?           00:00:00 cpuhp/0
 15 ?           00:00:00 kdevtmpfs
 16 ?           00:00:00 netns
 17 ?           00:00:00 rcu_tasks_kthre
 18 ?           00:00:00 kauditd
 19 ?           00:00:00 khungtaskd
 20 ?           00:00:00 oom_reaper
 21 ?           00:00:00 writeback
 22 ?           00:00:00 kcompactd0
 23 ?           00:00:00 ksm
 24 ?           00:00:00 khugepaged
 25 ?           00:00:00 crypto
 26 ?           00:00:00 kintegrityd
 27 ?           00:00:00 kblockd
 28 ?           00:00:00 tpm_dev_wq
```

Ps -aeH hanya mengarahkan pada proses yang dimasukan setelah grep, contohnya `ps -aeH | grep terminal`, berarti `ps -aeH` hanya mengarahkan pada semua proses yang bernama terminal. Init proses atau disebut juga sistem daemon adalah `kthreadd` dan `systemd` yang sangat penting untuk sistem operasi Linux. Identifikasi shell dan subprocess.

10. Kombinasikan ps -fae dan grep, apa yang Anda lihat ?

```
feri@feri-VirtualBox: ~  
File Edit View Search Terminal Help  
feri@feri-VirtualBox:~$ ps -fae  
UID      PID  PPID  C  STIME TTY      TIME CMD  
root      1      0  0  21:55 ?        00:00:03 /sbin/init splash  
root      2      0  0  21:55 ?        00:00:00 [kthreadd]  
root      3      2  0  21:55 ?        00:00:00 [rcu_gp]  
root      4      2  0  21:55 ?        00:00:00 [rcu_par_gp]  
root      5      2  0  21:55 ?        00:00:00 [kworker/0:0-rcu]  
root      6      2  0  21:55 ?        00:00:00 [kworker/0:0H-kb]  
root      8      2  0  21:55 ?        00:00:00 [mm_percpu_wq]  
root      9      2  0  21:55 ?        00:00:00 [ksoftirqd/0]  
root     10      2  0  21:55 ?        00:00:00 [rcu_sched]  
root     11      2  0  21:55 ?        00:00:00 [migration/0]  
root     13      2  0  21:55 ?        00:00:00 [idle_inject/0]  
root     14      2  0  21:55 ?        00:00:00 [cpuhp/0]  
root     15      2  0  21:55 ?        00:00:00 [kdevtmpfs]  
root     16      2  0  21:55 ?        00:00:00 [netns]  
root     17      2  0  21:55 ?        00:00:00 [rcu_tasks_kthre]  
root     18      2  0  21:55 ?        00:00:00 [kauditd]  
root     19      2  0  21:55 ?        00:00:00 [khungtaskd]  
root     20      2  0  21:55 ?        00:00:00 [oom_reaper]  
root     21      2  0  21:55 ?        00:00:00 [writeback]  
root     22      2  0  21:55 ?        00:00:00 [kcompactd0]  
root     23      2  0  21:55 ?        00:00:00 [ksmd]  
root     24      2  0  21:55 ?        00:00:00 [khugepaged]  
root     25      2  0  21:55 ?        00:00:00 [crypto]  
root     26      2  0  21:55 ?        00:00:00 [kintegrityd]  
root     27      2  0  21:55 ?        00:00:00 [kblockd]
```

11. 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 ?

```
feri@feri-VirtualBox: ~  
File Edit View Search Terminal Help  
feri@feri-VirtualBox:~$ sleep 300&  
[1] 1689  
feri@feri-VirtualBox:~$ SS
```

Pada saat melakukan Log Off, semua proses yang dijalankan oleh user tersebut akan berhenti, sehingga program sleep 300 juga berhenti. Dan ketika melakukan Log in ulang, system hanya akan menjalankan program – program dasar dan program start up.

3. KESIMPULAN

Pada Praktikum ini saya tidak menggunakan terminal tty2, dikarenakan ada beberapa perintah yang tidak bisa terscroll up dan stuck di bawah hasil perintah. Setiap kali instruksi atau perintah yang diberikan pada Shell Linux, kernel secara otomatis akan menciptakan proses- id. Proses ini disebut juga dalam terminologi UNIX sebagai JOB.

Proses - proses sistem terbagi dalam tiga tipe utama, yaitu :

- ✓ Interactive : Diprakarsai oleh sebuah shell dan berjalan dalam foreground dan background.
- ✓ Batch : Secara tipikal merupakan sebuah seri dari proses-proses yang dijadwalkan untuk dieksekusi pada suatu waktu tertentu.
- ✓ Daemon : Secara tipikal diinisialisasi saat boot untuk membentuk fungsi-fungsi sistem yang dibutuhkan, seperti LPD, NFS dan DNS

Dari praktikum diatas dapat disimpulkan bahwa kita dapat mengubah tampilan system, berbasis GUI (Graphical Unit Interface) ataupun TUI (Text Unit Interface). Semua proses atau perintah dapat dijalankan baik dengan basis GUI ataupun TUI.