

LAPORAN PRAKTIKUM SISTEM OPERASI

Praktikum 4 PROSES DAN MANAJEMEN PROSES

Oleh:

**FARID AZIZ WICAKSONO
NIM. 1841720094**



**JURUSAN TEKNOLOGI INFORMASI
PROGRAM STUDI TEKNIK INFORMATIKA
POLITEKNIK NEGERI MALANG
FEBRUARI 2019**

Percobaan 5: MENGHENTIKAN DAN MEMULAI KEMBALI JOB

\$yes > /dev/ null hasil outputnya:

```
farid@farid-VirtualBox:/home$ yes > /dev/null
^Z
[2]+  Stopped                  yes > /dev/null
```

analisnya :

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

2. \$ fg

hasil ouputnya :

```
farid@farid-VirtualBox:/home$ fg
yes > /dev/null
^Z
[2]+  Stopped                  yes > /dev/null
```

analisnya :

Dalam perintah \$ fg dipergunakan untuk merestart proses atau dengan kata lain job yang di suspend/di berhentikan untuk sementara.

3. \$bg

hasil outputnya :

```
farid@farid-VirtualBox:/home$ bg
[2]+ yes > /dev/null &
```

analisnya :

kita mensuspend job “yes >/dev?null” pada screenshot di atas. Kemudian memindahkannya ke background proses, yang berarti proses atau job tersebut tidak berhenti,akan tetapi terus berjalan di belakang layar.

4.\$ yes &

hasil outputnya:

```
y
y
y
y
y
y
y
^Z
[3]+  Stopped                  yes
```

analisnya:

perintah \$ yes & ini berjalan pada job background yang kita buat tadi.kita dapat menghentikan prosesnya, dengan kita harus memindahkan poses ini menjadi foreground dengan menggunakan Ctrl +Z.

5. Perintah \$ fg %<nomor job> dan \$bg %<nomor job>

```
farid@farid-VirtualBox:/home$ fg &1
[3] 19010
bash: fg: no job control
1: command not found
[3]- Exit 1          fg
```

analisnya :

Perintah \$ fg %<nomor job> dan \$bg %<nomor job> disini berguna untuk menjalankan banyak job dalam satu waktu atau multitasking. Dalam hal ini, job – job yang sedang tidak dipakai dikelompokkan dalam background atau dimasukkan ke proses background, dan untuk mengenalinya kita menggunakan nomor job yang unik (tidak ada job yang memiliki nomor yang sama).

6. Perintah \$ ps -fae dan \$ kill -9 <nomor PID>

```
farid@farid-VirtualBox:/home$ ps -fae
  UID      PID  PPID  C  STIME TTY          TIME CMD
root         1      0  0  20:19 ?        00:00:03 /sbin/init splash
root         2      0  0  20:19 ?        00:00:00 [kthreadd]
root         3      2  0  20:19 ?        00:00:00 [rcu_gp]
root         4      2  0  20:19 ?        00:00:00 [rcu_par_gp]
root         6      2  0  20:19 ?        00:00:00 [kworker/0:0H-kblockd]
root         8      2  0  20:19 ?        00:00:00 [mm_percpu_wq]
root         9      2  0  20:19 ?        00:00:00 [ksoftirqd/0]
root        10      2  0  20:19 ?        00:00:01 [rcu_sched]
root        11      2  0  20:19 ?        00:00:00 [rcu_bh]
root        12      2  0  20:19 ?        00:00:00 [migration/0]
root        13      2  0  20:19 ?        00:00:00 [watchdog/0]
root        14      2  0  20:19 ?        00:00:00 [cpuhp/0]
root        15      2  0  20:19 ?        00:00:00 [kdevtmpfs]
root        16      2  0  20:19 ?        00:00:00 [netns]
root        17      2  0  20:19 ?        00:00:00 [rcu_tasks_kthre]
root        18      2  0  20:19 ?        00:00:00 [kauditd]
root        19      2  0  20:19 ?        00:00:00 [khungtaskd]
root        20      2  0  20:19 ?        00:00:00 [oom_reaper]
root        21      2  0  20:19 ?        00:00:00 [writeback]
```

Output dari \$ kill -9 17660

Analisnya :

Perintah \$ ps -fae digunakan untuk menampilkan secara lengkap seluruh proses yang sedang berjalan beserta detailnya. Sedangkan perintah \$ kill -9 <nomor PID> digunakan untuk menghentikan atau terminate suatu proses dengan memasukkan nomor PID sebagai pengenalan prosesnya.

Percobaan 6:

Login sebagai root, buka tiga terminal dan ketikkan perintah \$ PS1="\w:" di setiap windownya.
Tampilan outputnya:

```
farid@farid-VirtualBox:/home$ PS1="\w:"  
/w:pwd  
/home
```

Analisisnya:

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

Sedangkan perintah \$ pwd digunakan untuk mengetahui lokasi aktif kita sekarang.

Buka satu lagi terminal, ketikkan perintah \$ top pada terminal tersebut maka program top akan muncul, kemudian ketik 'i', lalu ketik 'lmt' dan amati hasilnya Tampilan outputnya:

Memasukan perintah \$ top.

```
top - 21:41:23 up 4:51, 1 user, load average: 2,42, 2,09, 1,35  
Tasks: 210 total, 3 running, 175 sleeping, 1 stopped, 0 zombie  
%Cpu(s): 40,7 us, 59,3 sy, 0,0 ni, 0,0 id, 0,0 wa, 0,0 hi, 0,0 si, 0,0 st  
KiB Mem : 2041316 total, 99760 free, 1320900 used, 620656 buff/cache  
KiB Swap: 483800 total, 483020 free, 780 used. 583900 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
2816	besta	20	0	14580	780	716	R	44,7	0,0	11:19.57	yes
2834	besta	20	0	14580	820	756	R	44,0	0,0	5:43.38	yes
1142	besta	20	0	3051652	371328	94552	S	7,9	18,2	8:57.24	gnome-shell
1017	besta	20	0	471980	133000	64812	S	2,3	6,5	1:32.60	Xorg
2879	besta	20	0	802188	38160	28200	S	1,0	1,9	0:00.62	gnome-termi+
1	root	20	0	159928	9172	6712	S	0,0	0,4	0:02.62	systemd
2	root	20	0	0	0	0	S	0,0	0,0	0:00.00	kthreadd
4	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	kworker/0:0H
Settings	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	mm_percpu_wq
7	root	20	0	0	0	0	S	0,0	0,0	0:00.16	ksoftirqd/0
8	root	20	0	0	0	0	I	0,0	0,0	0:01.10	rcu_sched

Mengetikkan perintah 'i'

```
top - 21:41:37 up 4:51, 1 user, load average: 2,32, 2,08, 1,37  
Tasks: 210 total, 4 running, 175 sleeping, 1 stopped, 0 zombie  
%Cpu(s): 39,4 us, 60,6 sy, 0,0 ni, 0,0 id, 0,0 wa, 0,0 hi, 0,0 si, 0,0 st  
KiB Mem : 2041316 total, 99752 free, 1320908 used, 620656 buff/cache  
KiB Swap: 483800 total, 483020 free, 780 used. 583892 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
2816	besta	20	0	14580	780	716	R	45,9	0,0	11:26.64	yes
2834	besta	20	0	14580	820	756	R	45,9	0,0	5:50.45	yes
1142	besta	20	0	3051652	371388	94552	S	6,2	18,2	8:57.61	gnome-shell
1017	besta	20	0	471980	133000	64812	S	1,6	6,5	1:32.66	Xorg
2879	besta	20	0	802188	38160	28200	S	0,4	1,9	0:00.64	gnome-termi+

Memasukkan perintah 'lmt'

```
farid@farid-VirtualBox:~$ lmt

Command 'lmt' not found, did you mean:

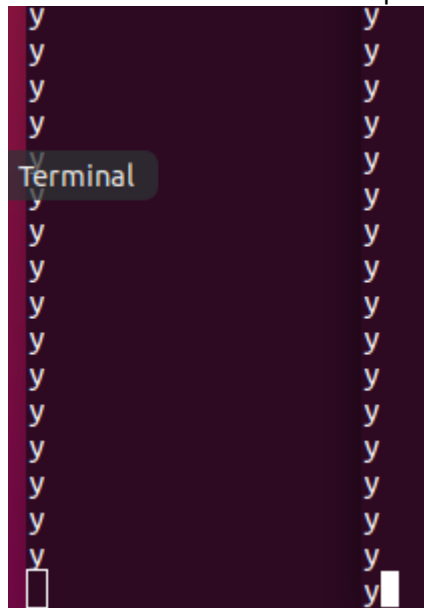
command 'smt' from snap smt (0.1)
command 'llt' from deb storebackup
command 'lft' from deb lft
command 'lft' from deb traceroute
command 'rmt' from deb tar
command 'rmt' from deb dump
command 'lbt' from deb lbt
command 'mt' from deb cpio
command 'mt' from deb mt-st
command 'fmt' from deb coreutils
command 'gmt' from deb gmt
command 'lt' from deb looptools

See 'snap info <snapname>' for additional versions.
```

Analisisnya:

Program top pada linux disini berfungsi seperti task manager pada windows, yaitu untuk mengetahui semua rincian proses yang berjalan, dan beberapa fungsi lainnya. Mengetikkan 'i' pada window top akan menampilkan proses yang sedang aktif. Ketikkan 'lmt' untuk menghilangkan atau tidak menampilkan informasi pada bagian atas dari window

Pada terminal 1 dan 2 ketikkan perintah \$ yes Tampilan outputnya:



Analisisnya :

Perintah \$ yes diatas untuk membuat proses baru.

Amati dua proses yes melalui Terminal Top. Kemudian buka terminal 3, ketikkan perintah \$ renice 19 <PID terminal 1> (disini PID nya = 2546), kemudian tekan Enter. Tampilan outputnya:

Analisisnya:

Dua proses yang sama akan menggunakan sumber daya yang sama (mendekati), misalkandua program \$ yes diatas, keduanya menggunakan sumber daya sama besarnya, namun PID dari kedua proses akan berbeda. Perintah \$ renice 17809 berarti mengganti penjadwalan prioritasdari proses ke 19. Setelah prioritasnya diubah menjadi 19, proses 17809 berjalan lebih lambat dari proses 17810.

Pilih terminal top dan tekan 'r', masukkan 2546 tekan Enter, kemudian ketik -19 lalu Enter.

```
top - 21:44:10 up 4:54, 1 user, load average: 2,96, 2,37, 1,58
Tasks: 210 total, 3 running, 175 sleeping, 1 stopped, 0 zombie
%Cpu(s): 37,9 us, 62,1 sy, 0,0 ni, 0,0 id, 0,0 wa, 0,0 hi, 0,0 si, 0,0 st
KiB Mem : 2041316 total, 96124 free, 1322872 used, 622320 buff/cache
KiB Swap: 483800 total, 483020 free, 780 used. 581928 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
2816	besta	20	0	14580	780	716	R	49,7	0,0	12:30.96	yes
2834	besta	20	0	14580	820	756	R	49,0	0,0	6:54.77	yes
1142	besta	20	0	3052536	372668	94552	S	0,7	18,3	9:10.69	gnome-shell
1268	besta	20	0	668940	23300	18232	S	0,3	1,1	0:01.63	gsd-color
1	root	20	0	159928	9172	6712	S	0,0	0,4	0:02.66	systemd
2	root	20	0	0	0	0	S	0,0	0,0	0:00.00	kthreadd
4	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	kworker/0:0H
6	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	mm_percpu_wq
7	root	20	0	0	0	0	S	0,0	0,0	0:00.17	ksoftirqd/0
8	root	20	0	0	0	0	I	0,0	0,0	0:01.13	rcu_sched
9	root	20	0	0	0	0	I	0,0	0,0	0:00.00	rcu_bh
10	root	rt	0	0	0	0	S	0,0	0,0	0:00.00	migration/0
11	root	rt	0	0	0	0	S	0,0	0,0	0:00.06	watchdog/0
12	root	20	0	0	0	0	S	0,0	0,0	0:00.00	rcu_sched

Tekan r dan ketik 2456

```
top - 21:44:28 up 4:54, 1 user, load average: 2,69, 2,34, 1,59
Tasks: 210 total, 3 running, 175 sleeping, 1 stopped, 0 zombie
%Cpu(s): 37,3 us, 62,7 sy, 0,0 ni, 0,0 id, 0,0 wa, 0,0 hi, 0,0 si, 0,0 st
KiB Mem : 2041316 total, 96000 free, 1322996 used, 622320 buff/cache
KiB Swap: 483800 total, 483020 free, 780 used. 581804 avail Mem
Renice PID 2546 to value
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
2834	besta	20	0	14580	820	756	R	47,0	0,0	7:03.59	yes
2816	besta	20	0	14580	780	716	R	46,7	0,0	12:39.76	yes
1142	besta	20	0	3052536	372688	94552	S	4,6	18,3	9:11.03	gnome-shell
1017	besta	20	0	471980	133000	64812	S	1,3	6,5	1:35.04	Xorg
2879	besta	20	0	803576	39464	28200	S	0,7	1,9	0:05.80	gnome-termi+
1	root	20	0	159928	9172	6712	S	0,3	0,4	0:02.67	systemd
2	root	20	0	0	0	0	S	0,0	0,0	0:00.00	kthreadd
4	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	kworker/0:0H
6	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	mm_percpu_wq
7	root	20	0	0	0	0	S	0,0	0,0	0:00.17	ksoftirqd/0

Ketik -19

```
top - 21:45:44 up 4:55, 1 user, load average: 2,27, 2,28, 1,63
Tasks: 209 total, 5 running, 173 sleeping, 1 stopped, 0 zombie
%Cpu(s): 39,4 us, 60,6 sy, 0,0 ni, 0,0 id, 0,0 wa, 0,0 hi, 0,0 si, 0,0 st
KiB Mem : 2041316 total, 96356 free, 1322568 used, 622392 buff/cache
KiB Swap: 483800 total, 483020 free, 780 used. 582212 avail Mem
```

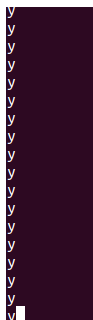
PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
2834	besta	20	0	14580	820	756	R	44,7	0,0	7:40.30	yes
2816	besta	20	0	14580	780	716	R	44,4	0,0	13:16.47	yes
1142	besta	20	0	3052536	372008	94552	R	7,6	18,2	9:12.34	gnome-shell
1017	besta	20	0	471980	133000	64812	S	2,0	6,5	1:35.25	Xorg
2879	besta	20	0	803576	39504	28200	S	1,0	1,9	0:05.90	gnome-termi+
2950	besta	20	0	51288	4092	3372	R	0,7	0,2	0:00.11	top
1162	besta	20	0	377856	9872	7960	S	0,3	0,5	0:06.52	ibus-daemon
1383	besta	20	0	220948	7956	7256	S	0,3	0,4	0:02.26	ibus-engine+

Analisisnya:

Program top juga memiliki fungsi yang sama dengan program renice. Jadi kita dapat mengubah konfigurasi proses melalui program renice maupun top. Setelah prioritrs proses 17809 diubah menjadi -19, proses 17809 menjadi lebih cepat daripada 17810

Dari kedua praktik diatas didapati bahwa semakin kecil prioritas dari suatu proses, akan semakin cepat proses tersebut berjalan.

Pilih terminal 3, masukkan perintah \$ nice -n -10 yes Tampilan output:



Analisisnya:

Perintah \$ nice -n -10 yes merupakan untuk membuat proses baru, dan opsi -10 merupakan penentuan prioritas dari proses tersebut. Dan kita ketahui juga bahwa PID nya adalah 17809.

Jangan menggunakan mouse atau keyboard selama 10 detik, amati yang terjadi pada proses top melalui terminal top. Kemudian pindahkan mouse sehingga pointer akan berubah posisi, amati yang terjadi pada proses top melalui terminal top.

Tampilan outputnya:

Proses top ketika mouse & keyboard diam

```
top - 21:47:56 up 4:57, 1 user, load average: 2,34, 2,31, 1,74
Tasks: 212 total, 3 running, 175 sleeping, 3 stopped, 0 zombie
%Cpu(s): 42,7 us, 57,3 sy, 0,0 ni, 0,0 id, 0,0 wa, 0,0 hi, 0,0 si, 0,0 st
KiB Mem : 2041316 total, 95372 free, 1323100 used, 622844 buff/cache
KiB Swap: 483800 total, 483020 free, 780 used. 581700 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
2816	besta	20	0	14580	780	716	R	43,0	0,0	14:16.37	yes
2816	besta	20	0	14580	820	756	R	43,0	0,0	8:40.19	yes
1142	besta	20	0	3052536	371676	94552	S	11,6	18,2	9:18.52	gnome-shell
1017	besta	20	0	471980	133000	64812	S	1,7	6,5	1:36.16	Xorg
32	root	20	0	0	0	0	I	0,3	0,0	0:04.77	kworker/0:1
2879	besta	20	0	803576	39504	28200	S	0,3	1,9	0:08.87	gnome-termi+
2959	besta	20	0	51292	4124	3404	R	0,3	0,2	0:00.04	top
1	root	20	0	159928	9172	6712	S	0,0	0,4	0:02.68	systemd
2	root	20	0	0	0	0	S	0,0	0,0	0:00.00	kthreadd

Proses top ketika mouse digerakkan

```
%Cpu(s): 35,4 us, 64,6 sy, 0,0 ni, 0,0 id, 0,0 wa, 0,0 hi, 0,0 si, 0,0 st
KiB Mem : 2041316 total, 95248 free, 1323224 used, 622844 buff/cache
KiB Swap: 483800 total, 483020 free, 780 used. 581576 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
2816	besta	20	0	14580	780	716	R	49,3	0,0	14:22.18	yes
2834	besta	20	0	14580	820	756	R	49,3	0,0	8:46.00	yes
1142	besta	20	0	3052536	371764	94552	S	0,7	18,2	9:18.86	gnome-shell
1017	besta	20	0	471980	133000	64812	S	0,3	6,5	1:36.20	Xorg
2959	besta	20	0	51292	4124	3404	R	0,3	0,2	0:00.08	top
1	root	20	0	159928	9172	6712	S	0,0	0,4	0:02.68	systemd
2	root	20	0	0	0	0	S	0,0	0,0	0:00.00	kthreadd
4	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	kworker/0:0H
6	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	mm_percpu_wq
7	root	20	0	0	0	0	S	0,0	0,0	0:00.17	ksoftirqd/0
8	root	20	0	0	0	0	I	0,0	0,0	0:01.15	rcu_sched

Analisisnya:

Saat mouse dan keyboard diam, penggunaan sumber daya oleh program top kecil. Namun ketika mouse mulai digerakkan atau ada perubahan posisi komponen semisal pointer, dll, proses top memakan sumber daya lebih banyak daripada saat diam tadi, salah satu alasannya adalah proses 17809 berjalan pada prioritas yang tinggi.

15. Pilih terminal top, ketikkan 'r', kemudian ketikkan 3611 tekan Enter, lalu akan muncul prompt ketik 0 dan tekan Enter. Pindahkan mouse ke sekeliling layar, lihat perbedaannya.

Tampilan output:

```
top - 21:50:22 up 5:00, 1 user, load average: 2,03, 2,19, 1,78
Tasks: 214 total, 3 running, 175 sleeping, 5 stopped, 0 zombie
%Cpu(s): 37,1 us, 62,9 sy, 0,0 ni, 0,0 id, 0,0 wa, 0,0 hi, 0,0 si, 0,0 st
KiB Mem : 2041316 total, 93496 free, 1324968 used, 622852 buff/cache
KiB Swap: 483800 total, 483020 free, 780 used. 579832 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
2816	besta	20	0	14580	780	716	R	48,0	0,0	15:27.30	yes
2834	besta	20	0	14580	820	756	R	48,0	0,0	9:51.12	yes
1142	besta	20	0	3052536	371624	94552	S	2,6	18,2	9:20.94	gnome-shell
1	root	20	0	159928	9172	6712	S	0,7	0,4	0:02.70	systemd
1017	besta	20	0	471980	133000	64812	S	0,3	6,5	1:36.53	Xorg
1335	besta	20	0	973216	60556	47468	S	0,3	3,0	0:02.34	nautilus-de+
2879	besta	20	0	803576	39680	28200	S	0,3	1,9	0:09.05	gnome-termi+
2961	besta	20	0	51292	4004	3276	R	0,3	0,2	0:00.03	top
2	root	20	0	0	0	0	S	0,0	0,0	0:00.00	kthreadd
4	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	kworker/0:0H

Sebelumnya kita lakukan :

Pertama : Ketik 'r' pada terminal top, ketik 3611 lalu Enter.

Lalu : Pada prompt berikut, masukkan 0 kemudian Enter.

Analisisnya:

Kita dapat mengubah konfigurasi suatu proses melalui perintah – perintah yang terdapat pada proses top. Dan ternyata setelah proses 17809 diubah prioritasnya menjadi 0, penggunaan sumber daya oleh terminal top menjadi lebih stabil (tidak banyak perubahan) walaupun mouse digerakkan ke sekeliling screen.

LATIHAN

Masuk ke tty2 dengan Ctrl+Alt+F2. Ketik perintah \$ ps -au kemudian tekan Enter. Ps -au

```
farid@farid-VirtualBox:~$ ps -au
```

USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
farid	1225	0.0	0.2	187900	2028	tty2	Ssl+	20:20	0:00	/usr/lib/gdm3/g
farid	1227	0.5	6.9	380964	69640	tty2	Sl+	20:20	0:47	/usr/lib/xorg/X
farid	1285	0.0	0.5	615716	5548	tty2	Sl+	20:20	0:00	/usr/lib/gnome-
farid	1503	0.1	0.4	328476	4128	tty2	Sl	20:20	0:09	ibus-daemon --x
farid	1509	0.0	0.2	254012	2116	tty2	Sl	20:20	0:00	/usr/lib/ibus/i
farid	1524	0.0	0.8	321344	8948	tty2	Sl	20:20	0:01	/usr/lib/ibus/i
farid	1528	0.0	0.4	244292	4964	tty2	Sl	20:20	0:00	/usr/lib/ibus/i
farid	1666	0.0	0.8	407448	8280	tty2	Sl+	20:20	0:00	/usr/lib/gnome-
farid	1674	0.0	0.3	280140	3036	tty2	Sl+	20:20	0:00	/usr/lib/gnome-
farid	1676	0.0	0.2	396908	2228	tty2	Sl+	20:20	0:00	/usr/lib/gnome-
farid	1677	0.0	0.2	249300	2152	tty2	Sl+	20:20	0:00	/usr/lib/gnome-
farid	1679	0.0	0.4	416392	4640	tty2	Sl+	20:20	0:00	/usr/lib/gnome-
farid	1681	0.0	0.2	288264	2244	tty2	Sl+	20:20	0:00	/usr/lib/gnome-
farid	1682	0.0	0.7	393188	8024	tty2	Sl+	20:20	0:00	/usr/lib/gnome-
farid	1687	0.0	0.1	347772	1872	tty2	Sl+	20:20	0:00	/usr/lib/gnome-

Sebutkan nama – nama proses yang bukan root.

Jawab:

Diantaranya adalah bash, -bash dan ps -au

b. Tulis PID dan COMMAND dari proses yang paling banyak menggunakan CPU time.

Jawab:

PID : 2474

COMMAND : /usr/bin/gnome-shell

c. Sebutkan buyut proses dan PID dari proses tersebut.

Jawab: Buyut proses yaitu proses yang memiliki PID terkecil yang menandakan program tersebut merupakan program yang pertama dijalankan.

Buyut proses : royyan

PID : 1331

d. Sebutkan beberapa proses daemon.

Jawab:

contoh proses daemon adalah inetd, named, popd, dan lain lainnya. Namun pada praktikum ini, tidak ada proses daemon yang berjalan.

e. Pada prompt login lakukan hal – hal sebagai berikut; \$ csh : untuk

\$ who : untuk melihat user yang aktif pada computer.

\$ bash :

\$ ls : untuk melihat isi dari directory yang aktif.

\$ sh :

\$ ps : untuk melihat proses yang sedang berjalan

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

Analisa:

Dalam hal ini kita dapat melihat seruh proses yang berjalan pada computer, menggunakan perintah ps -au.

PID terbesar dari data dibawah: 8670

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

\$ ps -f tampilan penuh tampilan outputnya :

```
farid@farid-VirtualBox:~$ ps -f
UID          PID    PPID    C  STIME TTY          TIME CMD
farid       19124  19050    0  22:51 pts/0        00:00:00 bash
farid       19154  19124    0  22:54 pts/0        00:00:00 ps -f
```

analisnya :

h. Hasil dari daftar penuh berupa UID,PID,PPID,C,STIME,TTY,TIME,CMD. Yang dapat kita lihat pada gambar di atas.

\$ ps -j format job. Tampilan output :

```
farid@farid-VirtualBox:~$ ps -j
  PID  PGID   SID  TTY          TIME CMD
19124 19124 19124 pts/0        00:00:00 bash
19156 19156 19124 pts/0        00:00:00 ps
```

analisnya :

Berisi PID, PGID, SID, TTY, TIME, CMD.Dan hasilnya berupa format job (biasanya hanya dipenuhi oleh angka, karena lebih cenderung menampilkan ID dimana proses sedang bekerja).

\$ ps j format job control Tampilan outputnya:

```
farid@farid-VirtualBox:~$ ps -j
  PID  PGID   SID  TTY          TIME CMD
19124 19124 19124 pts/0        00:00:00 bash
19157 19157 19124 pts/0        00:00:00 ps
```

analisnya :

berupa format job yang lebih mendetail hasilnya.

\$ ps l berupa daftar memanjang Tampilan outputnya :

```
farid@farid-VirtualBox:~$ ps l
F  UID    PID  PPID  PRI  NI   VSZ   RSS  WCHAN  STAT  TTY          TIME COMMAND
4  1000   1225  1099   20    0  187900  2028 poll_s Ssl+  tty2        0:00 /usr/lib/gd
4  1000   1227  1225   20    0  380964 69640 ep_pol Sl+   tty2        0:48 /usr/lib/xo
0  1000   1285  1225   20    0  615716 5548 poll_s Sl+   tty2        0:00 /usr/lib/gn
0  1000   1503    1   20    0  328476 4128 poll_s Sl    tty2        0:09 ibus-daemon
0  1000   1509  1503   20    0  254012 2116 poll_s Sl    tty2        0:00 /usr/lib/ib
0  1000   1524  1503   20    0  321344 8948 poll_s Sl    tty2        0:01 /usr/lib/ib
0  1000   1528    1   20    0  244292 4964 poll_s Sl    tty2        0:00 /usr/lib/ib
0  1000   1666  1285   20    0  407448 8280 poll_s Sl+   tty2        0:00 /usr/lib/gn
0  1000   1674  1285   20    0  280140 3036 poll_s Sl+   tty2        0:00 /usr/lib/gn
0  1000   1676  1285   20    0  396908 2228 poll_s Sl+   tty2        0:00 /usr/lib/gn
0  1000   1677  1285   20    0  249300 2152 poll_s Sl+   tty2        0:00 /usr/lib/gn
0  1000   1679  1285   20    0  416392 4640 poll_s Sl+   tty2        0:00 /usr/lib/gn
0  1000   1681  1285   20    0  288264 2244 poll_s Sl+   tty2        0:00 /usr/lib/gn
```

analisan :

Daftar yang memanjang berisi F, UID, PID,PPID,PRI,NI,VSZ,RS,S,WCHAN,STAT,TTY,TIME COMMAND.

\$ ps s format sinyal

Tampilan outputnya :

```
farid@farid-VirtualBox:~$ ps s
  UID    PID  PENDING   BLOCKED   IGNORED   CAUGHT  STAT  TTY      TIME COMMAND
 1000  1225  00000000  00000000  00001000 <80014000 Ssl+  tty2    0:00 /usr/li
 1000  1227  00000000  00000000  00001000 <c18066ef SL+   tty2    0:48 /usr/li
 1000  1285  00000000  00000000  00001000 <80014a02 SL+   tty2    0:00 /usr/li
 1000  1503  00000000  00000000  01001000 <80010000 SL   tty2    0:09 ibus-da
 1000  1509  00000000  00000000  01001000 <80000000 SL   tty2    0:00 /usr/li
 1000  1524  00000000  00000000  01001000 <80000000 SL   tty2    0:01 /usr/li
 1000  1528  00000000  00000000  01001000 <80004002 SL   tty2    0:00 /usr/li
 1000  1666  00000000  00000000  00001000 <80000000 SL+   tty2    0:00 /usr/li
 1000  1674  00000000  00000000  00001000 <80000000 SL+   tty2    0:00 /usr/li
 1000  1676  00000000  00000000  00001000 <80000000 SL+   tty2    0:00 /usr/li
 1000  1677  00000000  00000000  00001000 <80000000 SL+   tty2    0:00 /usr/li
 1000  1679  00000000  00000000  00001000 <80000000 SL+   tty2    0:00 /usr/li
 1000  1681  00000000  00000000  00001000 <80000000 SL+   tty2    0:00 /usr/li
 1000  1682  00000000  00000000  00001000 <80000000 SL+   tty2    0:00 /usr/li
 1000  1687  00000000  00000000  00001000 <80000000 SL+   tty2    0:00 /usr/li
```

analisisnya:

biasanya angka yang muncul banyak, berisi UID, PID, PENDING, BLOCKED, IGNORED, CAUGHT, STAT, TTY, TIME, COMMAND. Dan ini berupa format sinyal.

\$ ps v format virtual memory Tampilan outputnya:

```
farid@farid-VirtualBox:~$ ps v
  PID  TTY      STAT  TIME  MAJFL  TRS   DRS   RSS  %MEM  COMMAND
 1225  tty2     Ssl+  0:00   61     58  187841 2028  0.2  /usr/lib/gdm3/gdm-x-ses
 1227  tty2     SL+   0:49  1657  1828  379135 69640  6.9  /usr/lib/xorg/Xorg vt2
 1285  tty2     SL+   0:00   188   197  615518 5548  0.5  /usr/lib/gnome-session/
 1503  tty2     SL    0:09   22   123  328352 4128  0.4  ibus-daemon --xim --pan
 1509  tty2     SL    0:00    3   12  253999 2116  0.2  /usr/lib/ibus/ibus-dcon
 1524  tty2     SL    0:01  293  110  321233 8948  0.8  /usr/lib/ibus/ibus-exte
 1528  tty2     SL    0:00   166   73  244218 4964  0.4  /usr/lib/ibus/ibus-x11
 1666  tty2     SL+   0:00   275   62  407385 8280  0.8  /usr/lib/gnome-settings
 1674  tty2     SL+   0:00   108   31  280108 3036  0.3  /usr/lib/gnome-settings
 1676  tty2     SL+   0:00   20   26  396881 2228  0.2  /usr/lib/gnome-settings
 1677  tty2     SL+   0:00   19   13  249286 2152  0.2  /usr/lib/gnome-settings
```

analisisnya:

dari sini kita dapat melihat hasilnya berupa format virtual memory.

\$ps X

Tampilan outputnya:

```
farid@farid-VirtualBox:~$ ps x
  PID  TTY      STAT  TIME  COMMAND
 1151  ?        Ss     0:00  /lib/systemd/systemd --user
 1166  ?        S       0:00  (sd-pam)
 1219  ?        SLL    0:00  /usr/bin/gnome-keyring-daemon --daemonize --login
 1225  tty2     Ssl+   0:00  /usr/lib/gdm3/gdm-x-session --run-script env GNOME_SH
 1227  tty2     SL+    0:49  /usr/lib/xorg/Xorg vt2 -displayfd 3 -auth /run/user/1
 1280  ?        Ss     0:00  /usr/bin/dbus-daemon --session --address=systemd: --n
 1285  tty2     SL+    0:00  /usr/lib/gnome-session/gnome-session-binary --session
 1412  ?        Ss     0:00  /usr/bin/ssh-agent /usr/bin/im-launch env GNOME_SHELL
 1415  ?        Ssl    0:00  /usr/lib/gvfs/gvfsd
 1420  ?        SL     0:00  /usr/lib/gvfs/gvfsd-fuse /run/user/1000/gvfs -f -o bi
 1429  ?        Ssl    0:00  /usr/lib/at-spi2-core/at-spi-bus-launcher
 1434  ?        S       0:00  /usr/bin/dbus-daemon --config-file=/usr/share/default
```

analisisnya:

dari sini kita dapat melihat hasilnya , berisi PID, STACKP, ECP, EIP, TMOUT, ALARM, STAT, TTY, TIME, COMMAND.

Lakukan urutan pekerjaan berikut:

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
Tampilan outputnya:

```
farid@farid-VirtualBox:~$ find >directories.txt | grep error > errors.txt
```

Analisisnya :

Dengan mengetikkan perintah:

\$ find > directories.txt | grep error >errors.txt maka akan muncul tampilan seperti di atas.

b. Gunakan perintah sleep 5. Apa yang terjadi dengan perintah ini?

Tampilan outputnya:

```
farid@farid-VirtualBox:~$ sleep 5
```

Analisisnya :

Dalam hal ini proses dari terminal akan terhenti selama 5 detik lamanya.

Jalankan perintah Background dengan perintah &. Tampilan outputnya:

```
farid@farid-VirtualBox:~$ sleep 5 &  
[1] 19170
```

Analisisnya :

proses tersebut akan masuk kedalam proses background, dengan memasukkan option & dalam suatu perintah proses.

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

Tampilan outputnya:

```
farid@farid-VirtualBox:~$ sleep 15
```

Analisisnya:

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

```
farid@farid-VirtualBox:~$ sleep 15 &  
[1] 19172
```

Analisisnya:

untuk menghentikan proses, menggunakan perintah kill [PID].

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

Tampilan outputnya:

```
farid@farid-VirtualBox:~$ sleep 15  
[1]+  Done                  sleep 5  
farid@farid-VirtualBox:~$ sleep 15 &  
[1] 19172
```

Analisisnya:

Untuk menghentikan sementara proses menggunakan perintah kill, harus memasukkan perintah kill -1 [job number].

Jalankan sleep 60 pada background 5 kali dan terminasi semuanya menggunakan perintah killall.

Tampilan outputnya:

```
farid@farid-VirtualBox:~$ sleep 60 &
[2] 19174
[1] Done sleep 15
farid@farid-VirtualBox:~$ sleep 60 &
[3] 19175
farid@farid-VirtualBox:~$ sleep 60 &
[4] 19176
farid@farid-VirtualBox:~$ sleep 60 &
[5] 19177
farid@farid-VirtualBox:~$ sleep 60 &
[6] 19178
farid@farid-VirtualBox:~$ killall sleep
[2] Terminated sleep 60
[3] Terminated sleep 60
[4] Terminated sleep 60
[5]- Terminated sleep 60
[6]+ Terminated sleep 60
```

Analisisnya:

Perintah killall sleep diatas memerintahkan pada system untuk menghentikan semua proses yang dalam namanya terdapat kata "sleep".

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

Tampilan outputnya:

```
farid@farid-VirtualBox:~$ ps
  PID TTY          TIME CMD
 19124 pts/0    00:00:00 bash
 19181 pts/0    00:00:00 ps
farid@farid-VirtualBox:~$ w
 23:04:18 up  2:44,  1 user,  load average: 0,10, 0,13, 0,18
USER      TTY      FROM              LOGIN@   IDLE   JCPU   PCPU WHAT
farid     :0        :0                19:17   ?xdm?  48.85s  0.01s /usr/lib/gdm3/g
farid@farid-VirtualBox:~$ top

top - 23:04:28 up  2:44,  1 user,  load average: 0,09, 0,12, 0,18
Tasks: 167 total,  1 running, 166 sleeping,  0 stopped,  0 zombie
%Cpu(s):  5,4 us,  2,7 sy,  0,0 ni, 91,9 id,  0,0 wa,  0,0 hi,  0,0 si,  0,0 st
MiB Mem :  985,3 total,  90,7 free,  538,5 used,  356,1 buff/cache
MiB Swap:  472,5 total,  284,0 free,  188,5 used.  267,6 avail Mem

  PID USER      PR  NI   VIRT   RES   SHR  S  %CPU  %MEM    TIME+  COMMAND
 1227 farid     20   0  381204 69640 30452 S   3,6   6,9   0:53.64 Xorg
18821 farid     20   0 2676348 275540 113736 S   2,6  27,3   1:34.47 gnome-she+
```

Ps : perintah untuk melihat proses yang sedang berjalan.

w : perintah untuk melihat proses beserta rincian lengkap dari semua program yang berjalan.

i. Gunakan perintah `ps -aeH` untuk menampilkan hierarki proses. Carilah init proses. Apakah anda bisa identifikasi system daemon yang penting? Dapatkan anda identifikasi shell dan subprocesses?

Tampilan outputnya:

Perintah `ps -aeH`.

```
farid@farid-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
    6 ?            00:00:00 kworker/0:0H-kblockd
    8 ?            00:00:00 mm_percpu_wq
    9 ?            00:00:00 ksoftirqd/0
   10 ?            00:00:01   rcu_sched
   11 ?            00:00:00   rcu_bh
   12 ?            00:00:00 migration/0
   13 ?            00:00:00 watchdog/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
```

Shell : pada screen daemon diatas, yang merupakan shell adalah dbus, avahi, account, dan rtkit.

Subproses : pada screen daemon diatas, dbus, ibus, syn, dan bamf merupakan subprocesses

Analisisnya :

`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.

j. Kombinasikan `ps -fae` dan `grep`, apa yang anda lihat?

Tampilan outputnya:

```
farid@farid-VirtualBox:~$ ps -fae | grep
Usage: grep [OPTION]... PATTERN [FILE]...
Try 'grep --help' for more information.
```

Analisisnya:

`ps -fae | grep terminal`, berarti `ps -fae` hanya mengarahkan pada semua proses yang bernama terminal

k. Jalankan proses sleep 300 pada background. Log off computer dan log in kembali. Lihat daftar semua proses yang berjalan. Apa yang terjadi pada proses sleep?

Tampilan outputnya:

```
farid@farid-VirtualBox:~$ sleep 300 &  
[1] 19264
```

Analisisnya:

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.

KESIMPULAN

Dalam hal ini, saya tidak masuk ke tty2, dikarenakan tidak memungkinkan untuk mengambil gambar atau screenshotnya, jadi disini saya menggunakan terminal.

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.