

Operating Systems

Spring 2018

S01

Professor : Philippe Cudré-Mauroux
Assistant : Ines Arous

Submitted by Sylvain Julmy

Exercice 2

a)

In kernel mode, the executing code has a complete access to the hardware. There is no restriction about it. The code can do almost anything from executing any CPU instruction or accessing any memory address.

In user mode, the executing code has to use system call in order to access low-level functionality like memory allocation. The executing code memory is allocated by the system and the code don't have any access to the memory of the other running code.

Having two separate mode is usefull in designing an operating system for the following reason :

- The critical code which are purely system related are running in kernel mode for efficiency. Those code have to be sure and idealy certified and verified.
- Any non-system related code run in user mode, because any failure of the code would be recoverable due to the memory management by the system.

b)

Only reading the time-of-day clock should be allowed in user mode as well as in kernel mode. All other instructions should be only allowed in kernel mode for security.

c)

Advantages :

- Increase the hardware utilisation.
- Decrease capital and operating cost.
- Run program in a VM which maybe canno't run on the physical one due to operating system compatibility.
- We can save the state of a VM.
- VM are isolated from the host machine. In case of malware, the host is safe.

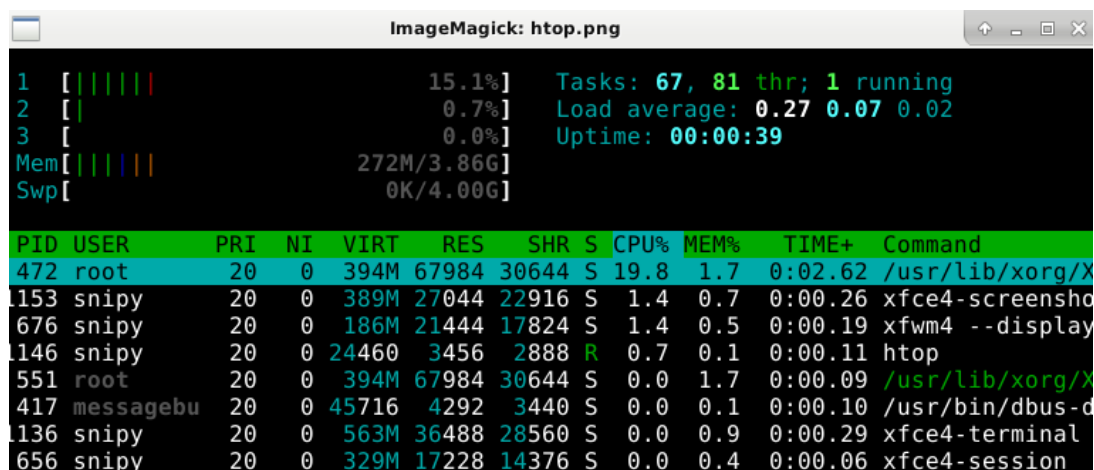
Disadvantage :

- The host has to be accessible in order for all the VM to be accessible.
- The VM don't have a direct access to the hardware.
- Increased memory and processor usage as part of overhead introduced by the VM.
- When multiple virtual machines are simultaneously running on a host computer, each virtual machine may introduce an unstable performance, which depends on the workload on the system by other running virtual machines.

Exercice 3

a)

Using the command `htop`, we can visualize the number of CPU of the computer and the memory usage of the system.



```
1 [|||||] 15.1% Tasks: 67, 81 thr: 1 running
2 [|] 0.7% Load average: 0.27 0.07 0.02
3 [|] 0.0% Uptime: 00:00:39
Mem[|||||] 272M/3.86G
Swp[|] 0K/4.00G
```

PID	USER	PRI	NI	VIRT	RES	SHR	S	CPU%	MEM%	TIME+	Command
472	root	20	0	394M	67984	30644	S	19.8	1.7	0:02.62	/usr/lib/xorg/X
153	snipy	20	0	389M	27044	22916	S	1.4	0.7	0:00.26	xfce4-screensho
676	snipy	20	0	186M	21444	17824	S	1.4	0.5	0:00.19	xfwm4 --display
146	snipy	20	0	24460	3456	2888	R	0.7	0.1	0:00.11	htop
551	root	20	0	394M	67984	30644	S	0.0	1.7	0:00.09	/usr/lib/xorg/X
417	messagebu	20	0	45716	4292	3440	S	0.0	0.1	0:00.10	/usr/bin/dbus-d
136	snipy	20	0	563M	36488	28560	S	0.0	0.9	0:00.29	xfce4-terminal
656	snipy	20	0	329M	17228	14376	S	0.0	0.4	0:00.06	xfce4-session

The command simply show so information about the current system state and all the active process.

Number of CPU : 3

How much memory : 3.86G

Fraction of the memory used : 272M/3.86G

b)

In order to construct the complete tree of the processes, we can use `htop` too. The command has an option which construct the tree for us :

```
TIME+ Command
0:00.65 /sbin/init
0:01.44 └─ xfce4-terminal
0:00.02 │   └─ bash
0:01.00 │       └─ htop
0:00.00 │   └─ xfce4-terminal
0:00.00 └─ xfce4-terminal
0:00.19 └─ /usr/bin/pulseaudio --start --log-target=syslog
0:00.12 │   └─ /usr/bin/pulseaudio --start --log-target=syslog
0:00.04 │       └─ /usr/bin/pulseaudio --start --log-target=syslog
0:00.01 └─ /usr/lib/udisks2/udisksd --no-debug
0:00.00 │   └─ /usr/lib/udisks2/udisksd --no-debug
0:00.00 │       └─ /usr/lib/udisks2/udisksd --no-debug
0:00.00 │           └─ /usr/lib/udisks2/udisksd --no-debug
0:00.02 └─ xfce4-power-manager --restart --sm-client-id 2f51c081f-92cc-
0:00.00 │   └─ xfce4-power-manager --restart --sm-client-id 2f51c081f-92
0:00.00 │       └─ xfce4-power-manager --restart --sm-client-id 2f51c081f-92
0:00.02 └─ /usr/lib/upower/upowerd
0:00.00 │   └─ /usr/lib/upower/upowerd
0:00.00 │       └─ /usr/lib/upower/upowerd
0:00.03 └─ xfsettingsd --display :0.0 --sm-client-id 2ef417c9f-4b32-4df
0:00.00 │   └─ xfsettingsd --display :0.0 --sm-client-id 2ef417c9f-4b32-
0:00.00 │       └─ xfsettingsd --display :0.0 --sm-client-id 2ef417c9f-4b32-
0:00.02 └─ /lib/systemd/systemd --user
```

The init process is `/sbin/init` and it's the one that start all the other processes.