

Multiprocessing

1. Processes

- i. top shows you details of active processes. The processes are sorted by CPU usage by default. Sort them by memory usage.

```

thinesh@thinesh-VirtualBox: ~
top - 14:50:38 up 2 min,  1 user,  load average: 1.53, 1.04, 0.42
Tasks: 187 total,   4 running, 183 sleeping,   0 stopped,   0 zombie
%Cpu(s):  2.3 us,  2.0 sy,   0.0 ni, 95.3 id,   0.3 wa,   0.0 hi,   0.0 si,   0.0 st
MiB Mem :  3936.4 total,  2617.6 free,   668.7 used,   650.0 buff/cache
MiB Swap:   472.5 total,   472.5 free,    0.0 used.  3039.1 avail Mem

  PID USER      PR  NI   VIRT   RES   SHR  S  %CPU  %MEM    TIME+  COMMAND
  967 root        20   0 504620  79200 64368 S   0.3   2.0   0:00.58 packagek+
    1 root        20   0 101644  10916  7992 S   0.0   0.3   0:01.03 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_+
    5 root        20   0     0     0     0 I   0.0   0.0   0:00.03 kworker/+
    6 root         0 -20     0     0     0 I   0.0   0.0   0:00.00 kworker/+
    7 root        20   0     0     0     0 I   0.0   0.0   0:00.05 kworker/+
    8 root        20   0     0     0     0 I   0.0   0.0   0:00.03 kworker/+
    9 root         0 -20     0     0     0 I   0.0   0.0   0:00.00 mm_percp+
   10 root        20   0     0     0     0 R   0.0   0.0   0:00.14 ksoftirq+
   11 root        20   0     0     0     0 R   0.0   0.0   0:00.24 rcu_sched
   12 root        rt    0     0     0     0 S   0.0   0.0   0:00.00 migratio+
   13 root       -51   0     0     0     0 S   0.0   0.0   0:00.00 idle_inj+
   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_task+
   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 khungtas+
   20 root        20   0     0     0     0 S   0.0   0.0   0:00.00 oom_reap+
   21 root         0 -20     0     0     0 I   0.0   0.0   0:00.00 writeback

```

Sort them by memory usage.

top -o %MEM

```
thinesh@thinesh-VirtualBox: ~  
Thunderbird Mail  
top - 14:52:42 up 4 min, 1 user, load average: 0.49, 0.82, 0.42  
Tasks: 177 total, 2 running, 175 sleeping, 0 stopped, 0 zombie  
%Cpu(s): 6.8 us, 2.7 sy, 0.0 ni, 90.5 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st  
MiB Mem : 3936.4 total, 2391.1 free, 712.3 used, 833.1 buff/cache  
MiB Swap: 472.5 total, 472.5 free, 0.0 used. 2989.3 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1414	thinesh	20	0	2291508	250660	108676	S	3.6	6.2	0:06.11	gnome-sh+
1867	thinesh	20	0	770604	143184	40848	S	0.0	3.6	0:02.91	gnome-so+
1664	thinesh	20	0	798736	61956	47908	S	0.0	1.5	0:00.20	evolutio+
1212	thinesh	20	0	218584	54012	35932	R	3.0	1.3	0:01.17	Xorg
967	root	20	0	443232	50672	14268	S	0.0	1.3	0:04.82	packagek+
1847	thinesh	20	0	969092	47776	34908	S	2.6	1.2	0:00.85	gnome-te+
1903	root	20	0	404216	40552	21788	S	0.0	1.0	0:00.28	fwupd
1529	thinesh	20	0	731172	35104	30424	S	0.0	0.9	0:00.08	evolutio+
1458	thinesh	20	0	548376	34972	29340	S	0.0	0.9	0:00.07	goa-daem+
1869	thinesh	20	0	507984	32728	23844	S	0.0	0.8	0:00.20	update-n+
1511	thinesh	20	0	289708	32540	19804	S	0.0	0.8	0:01.07	ibus-ext+
1599	thinesh	20	0	701724	31396	21712	S	0.0	0.8	0:00.27	gsd-medi+
1546	thinesh	20	0	689396	30856	27068	S	0.0	0.8	0:00.05	evolutio+
474	root	20	0	652432	30508	14320	S	0.0	0.8	0:01.03	snappd
1586	thinesh	20	0	360644	30404	20576	S	0.0	0.8	0:00.24	gsd-xset+
1449	thinesh	20	0	487820	30360	25884	S	0.0	0.8	0:00.07	evolutio+
1577	thinesh	20	0	360684	30288	20728	S	0.0	0.8	0:00.25	gsd-power
1576	thinesh	20	0	582280	30240	20700	S	0.0	0.8	0:00.26	gsd-color
1388	thinesh	20	0	507544	29596	20164	S	0.0	0.7	0:00.24	xdg-desk+
1611	thinesh	20	0	360136	29496	20152	S	0.0	0.7	0:00.24	gsd-keyb+
1513	thinesh	20	0	212192	29400	20192	S	0.0	0.7	0:00.24	ibus-x11
1570	thinesh	20	0	359632	29212	19744	S	0.0	0.7	0:00.24	gsd-wacom

- ii. Run ps with the following options: -a, -x, -u, -w. What is the name of the process with PID 1?

Ps -a → list all the processes

```
thinesh@thinesh-VirtualBox:~$ ps -a  
PID TTY TIME CMD  
1212 tty2 00:00:01 Xorg  
1223 tty2 00:00:00 gnome-session-b  
2705 pts/0 00:00:00 ps  
thinesh@thinesh-VirtualBox:~$
```

Ps -x

```
thinesh@thinesh-VirtualBox: ~  
thinesh@thinesh-VirtualBox:~$ ps -x  
PID TTY      STAT   TIME COMMAND  
1170 ?        Ss     0:00 /lib/systemd/systemd --user  
1172 ?        S       0:00 (sd-pam)  
1184 ?        S<sl   0:00 /usr/bin/pulseaudio --daemonize=no  
1188 ?        Sl     0:00 /usr/bin/gnome-keyring-daemon --daemonize --login  
1192 ?        Ss     0:00 /usr/bin/dbus-daemon --session --address=systemd: --  
1194 ?        Ssl    0:00 /usr/lib/gvfs/gvfsd  
1199 ?        Sl     0:00 /usr/lib/gvfs/gvfsd-fuse /run/user/1000/gvfs -f -o b  
1210 tty2     Ssl+   0:00 /usr/lib/gdm3/gdm-x-session --run-script env GNOME_S  
1212 tty2     Sl+    0:01 /usr/lib/xorg/Xorg vt2 -displayfd 3 -auth /run/user/  
1223 tty2     Sl+    0:00 /usr/lib/gnome-session/gnome-session-binary --system  
1303 ?        Ss     0:00 /usr/bin/ssh-agent /usr/bin/im-launch env GNOME_SHEL  
1340 ?        Ssl    0:00 /usr/lib/at-spi2-core/at-spi-bus-launcher  
1348 ?        S       0:00 /usr/bin/dbus-daemon --config-file=/usr/share/default  
1357 ?        Sl     0:00 /usr/lib/at-spi2-core/at-spi2-registryd --use-gnome-  
1362 ?        Ssl    0:00 /usr/libexec/xdg-desktop-portal  
1363 ?        Ssl    0:00 /usr/lib/gnome-session/gnome-session-ctl --monitor  
1372 ?        Ssl    0:00 /usr/lib/gnome-session/gnome-session-binary --system  
1376 ?        Ssl    0:00 /usr/libexec/xdg-document-portal  
1379 ?        Ssl    0:00 /usr/libexec/xdg-permission-store  
1388 ?        Ssl    0:00 /usr/libexec/xdg-desktop-portal-gtk  
1404 ?        Sl     0:00 /usr/lib/dconf/dconf-service  
1414 ?        Ssl    0:07 /usr/bin/gnome-shell  
1445 ?        Sl     0:00 /usr/lib/gnome-shell/gnome-shell-calendar-server  
1449 ?        Ssl    0:00 /usr/libexec/evolution-source-registry  
1458 ?        Sl     0:00 /usr/lib/gnome-online-accounts/goa-daemon  
1461 ?        Ssl    0:00 /usr/lib/gvfs/gvfs-udisks2-volume-monitor  
1466 ?        Ssl    0:00 /usr/lib/gvfs/gvfs-gphoto2-volume-monitor
```

Ps -u

```
thinesh@thinesh-VirtualBox:~$ ps -u  
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND  
thinesh  1210  0.0  0.1 175032  6564 tty2     Ssl+  14:49   0:00 /usr/lib/gdm3/  
thinesh  1212  0.6  1.3 218584 54012 tty2     Sl+   14:49   0:02 /usr/lib/xorg/  
t Ubuntu Software 0.0  0.3 203116 16000 tty2     Sl+   14:49   0:00 /usr/lib/gnome  
thinesh  1858  0.0  0.1  21732  4972 pts/0    Ss    14:50   0:00 bash  
thinesh  2782  0.0  0.0  22608  3584 pts/0    R+    14:54   0:00 ps -u
```

Ps -w

```
Rhythmbox thinesh@thinesh-VirtualBox:~$ ps -w  
PID TTY      TIME CMD  
1858 pts/0    00:00:00 bash  
2783 pts/0    00:00:00 ps
```

What is the name of the process with PID 1 :

Ps -eaf

PID 1 will be always the init process

```
thinesh@thinesh-VirtualBox:~$ ps -eaf
UID          PID    PPID  C STIME TTY          TIME CMD
root          1      0  0 14:48 ?        00:00:01 /sbin/init splash
root          2      0  0 14:48 ?        00:00:00 [kthreadd]
root          3      2  0 14:48 ?        00:00:00 [rcu_gp]
root          4      2  0 14:48 ?        00:00:00 [rcu_par_gp]
root          6      2  0 14:48 ?        00:00:00 [kworker/0:0H-kblockd]
root          7      2  0 14:48 ?        00:00:00 [kworker/0:1-events]
root          8      2  0 14:48 ?        00:00:00 [kworker/u2:0-events_power_effi
root          9      2  0 14:48 ?        00:00:00 [mm_percpu_wq]
root         10      2  0 14:48 ?        00:00:00 [ksoftirqd/0]
root         11      2  0 14:48 ?        00:00:00 [rcu_sched]
root         12      2  0 14:48 ?        00:00:00 [migration/0]
root         13      2  0 14:48 ?        00:00:00 [idle_inject/0]
root         14      2  0 14:48 ?        00:00:00 [cpuhp/0]
root         15      2  0 14:48 ?        00:00:00 [kdevtmpfs]
root         16      2  0 14:48 ?        00:00:00 [netns]
root         17      2  0 14:48 ?        00:00:00 [rcu_tasks_kthre]
root         18      2  0 14:48 ?        00:00:00 [kauditd]
```

1.1. Creating a new process

Exercise 2:

1. In what order are the messages from parent and child printed? Is the order always the same? (Code attached ex2_1.c)

```
thinesh@thinesh-VirtualBox:~$ ./ex3
this is parent
this is the child
```

I got parent process first and then the child process printed.

But always parent process first and child process printed.

This is parent

This is the child

OR

This is the child

This is parent

Here, two outputs are possible because the parent process and child process are running concurrently. So we don't know whether the OS will first give control to the parent process or the child process.

- ii. How many children will the following program spawn? Draw a diagram illustrating the parent-child relationships between processes. (Code attached to ex2_2.c)

```
GNU nano 4.3 ex2_2.c
int pid;
for(int i=0;i<3;i++){
pid=fork();

if (pid<0){
    perror("fork");
    exit(1);
}

//while(pid!=0){
//wait();
//}

if (pid==0){
puts("This is the child");
printf("my pid(child) is %d\n",getpid());
printf("my parent pid(chid) is %d\n",getppid());
wait(2);
}
else
puts("this is parent");
printf("my pid(parent) is %d\n",getpid());
printf("my parent(parent) is %d\n",getppid());
wait(2);
```

There are eight processes. The program changed for identify how many child and parent processes are created.

```
fork ();    // Line 1
fork ();    // Line 2
fork ();    // Line 3

      L1      // There will be 1 child process
    /   \    // created by line 1.
  L2     L2   // There will be 2 child processes
 /  \   /  \  // created by line 2
L3  L3 L3  L3 // There will be 4 child processes
              // created by line 3
```

```
thinesh@thinesh-VirtualBox: ~  
thinesh@thinesh-VirtualBox:~$ ./ex2_2  
this is parent  
my pid(parent) is 12463  
my parent(parent) is 1858  
This is the child  
my pid(child) is 12464  
my Rhythmbox pid(chid) is 12463  
my pid(parent) is 12464  
my parent(parent) is 12463  
this is parent  
my pid(parent) is 12464  
my parent(parent) is 12463  
This is the child  
my pid(child) is 12465  
my parent pid(chid) is 12464  
my pid(parent) is 12465  
my parent(parent) is 12464  
this is parent  
my pid(parent) is 12465  
my parent(parent) is 12464  
This is the child  
my pid(child) is 12466  
my parent pid(chid) is 12465  
my pid(parent) is 12466  
my parent(parent) is 12465  
this is parent  
my pid(parent) is 12464  
my parent(parent) is 12463  
This is the child
```

1.2 Waiting for children

Modify the program in section 1.1 so that the parent always prints its message after the child. (Code attached to ex3.c)

```
thinesh@thinesh-VirtualBox:~$ gcc -o ex3 ex3.c && ./ex3  
this is child  
this is the parent  
thinesh@thinesh-VirtualBox:~$
```


1.3 Replacing the process image

1. Compile and run the above code giving it a path as an argument. How many times is the message "Program ls has terminated" printed?

(Code attached to ex4_1.c)

```
thinesh@thinesh-VirtualBox:~$ gcc -o ex3 ex4_1.c && ./ex3
Desktop  Downloads  ex2_1.c  ex2_2.c  ex3.c  Music  Public  Videos
Documents ex2_1      ex2_2    ex3      ex4_1.c Pictures Templates
```

Message was not printed even once.

2. Write a very simple shell that repeatedly prompts the user for a command and runs it with any arguments given. Make sure your shell waits until the command has completed before prompting the user for the next command. (Code attached to ex4_2.c)

```
thinesh@thinesh-VirtualBox:~$ ./ex4
e15366_simple_shell>> ls
ls
Desktop  ex2_1.c  ex3.c  ex5  exnew  Music  Videos
Documents ex2_2    ex4     ex5_1.c exnew.c Pictures
Downloads ex2_2.c  ex4_1.c ex5_2.c lab01   Public
ex2_1      ex3     ex4_2.c ex5_3.c lab01.zip Templates
Child exited
e15366_simple_shell>> dir
dir
Desktop  ex2_1.c  ex3.c  ex5  exnew  Music  Videos
Documents ex2_2    ex4     ex5_1.c exnew.c Pictures
Downloads ex2_2.c  ex4_1.c ex5_2.c lab01   Public
ex2_1      ex3     ex4_2.c ex5_3.c lab01.zip Templates
Child exited
e15366_simple_shell>> cd Lab01
cd
Lab01
Child process could not do execvp
e15366_simple_shell>> 
```

2. Multiprocess servers

Exercise 5:

1. Open three terminals and run the server in one. Use nc() to connect as two clients concurrently on port 12345. Type some text in both clients and examine the client and server outputs.

Code attached ex5_1.c

2. Suppose we modify the server parent process to call wait() on the last line above (highlighted) to wait until the child serving a client terminates. What would happen?

If we modify the server parent process to call wait() on the last line above, at the same time multiple clients can be connected but can not send message from one client to another client at the same time.

3. What happens if you terminate the the server while a client is connected, and then try to restart it? (Resolving this issue requires a signal handler.)

When client connected after the termination of the server, the client also terminated.

4. Modify this server to do the following: The client sends the path to a file whose contents the server will send back to the client (if the file exists.) Verify that your new server can handle multiple concurrent connections by using nc(). Can two concurrent clients request the same file?

