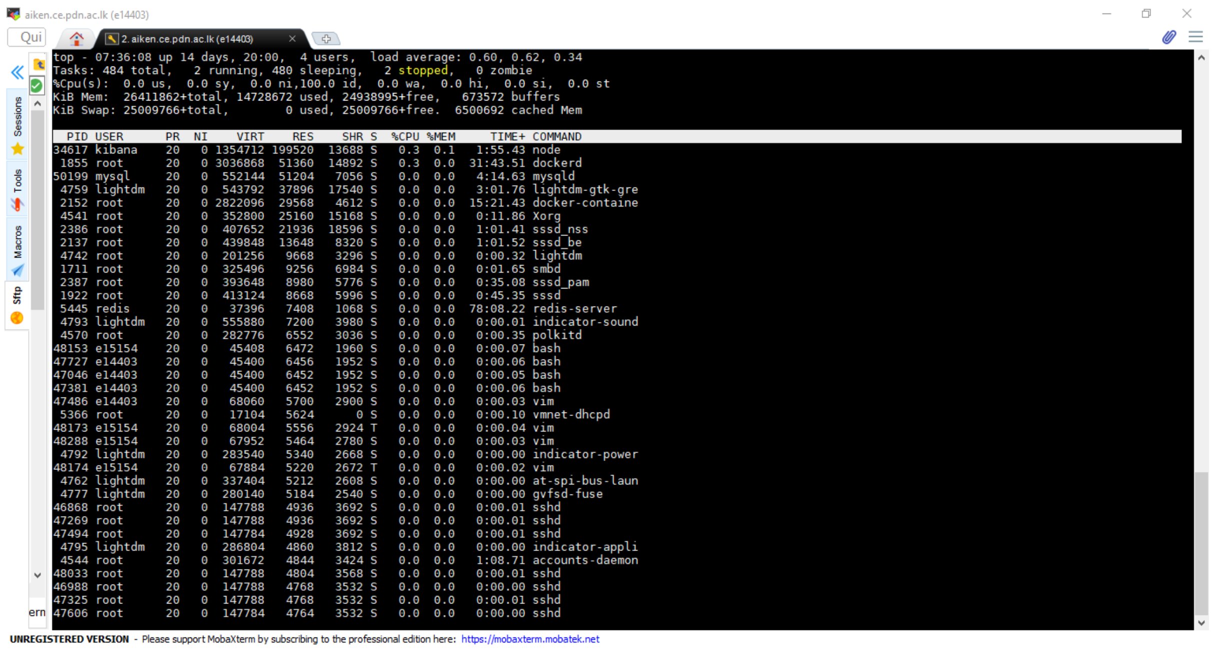
Exercise 1: Use the following commands to view details of the processing running on your system. Note the PIDs.

I. top shows you details of active processes. The processes are sorted by CPU usage by default. Sort them by memory usage.
Command => top -o %MEM

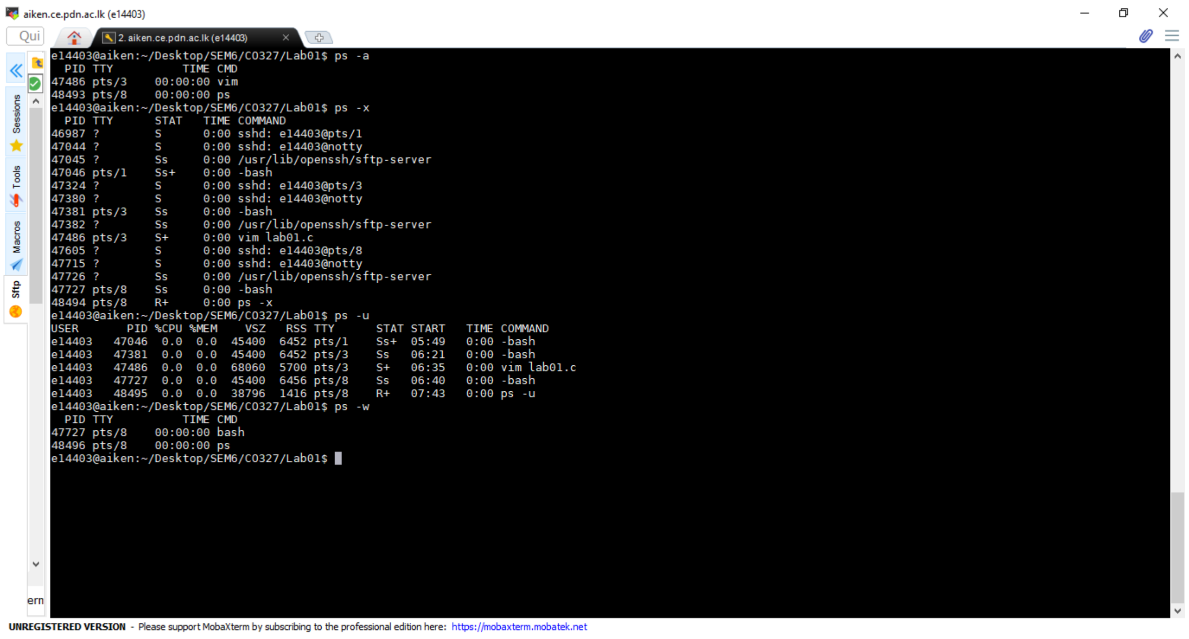


```
top - 07:36:08 up 14 days, 20:00,  4 users,  load average: 0.60, 0.62, 0.34
Tasks: 484 total,   2 running, 480 sleeping,   2 stopped,   0 zombie
%Cpu(s):  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem:  26411862+total, 14728672 used, 24938995+free,   673572 buffers
KiB Swap: 25009766+total,        0 used, 25009766+free.  6500692 cached Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+ COMMAND
34617 kibana    20   0 1354712 199520  13688 S   0.3  0.1   1:55.43 node
 1855 root      20   0 3036868  51360  14892 S   0.3  0.0  31:43.51 dockerd
50199 mysql     20   0  552144  51204   7056 S   0.0  0.0   4:14.63 mysqld
 4759 lightdm   20   0  543792  37896  17540 S   0.0  0.0   3:01.76 lightdm-gtk-gre
 2152 root      20   0 2822096  29568   4612 S   0.0  0.0  15:21.43 docker-containe
 4541 root      20   0  352800  25160  15168 S   0.0  0.0   0:11.86 Xorg
 2386 root      20   0  407652  21936  18596 S   0.0  0.0   1:01.41 sssd_nss
 2137 root      20   0  439848  13648   8320 S   0.0  0.0   1:01.52 sssd_be
 4742 root      20   0  201256   9668   3296 S   0.0  0.0   0:00.32 lightdm
 1711 root      20   0  325496   9256   6984 S   0.0  0.0   0:01.65 smbd
 2387 root      20   0  393648   8980   5776 S   0.0  0.0   0:35.08 sssd_pam
 1922 root      20   0  413124   8668   5996 S   0.0  0.0   0:45.35 sssd
 5445 redis     20   0   37396   7408   1068 S   0.0  0.0  78:08.22 redis-server
 4793 lightdm   20   0  555880   7200   3980 S   0.0  0.0   0:00.01 indicator-sound
 4570 root      20   0  282776   6552   3036 S   0.0  0.0   0:00.35 polkitd
48153 e15154    20   0   45408   6472   1960 S   0.0  0.0   0:00.07 bash
47727 e14403    20   0   45400   6456   1952 S   0.0  0.0   0:00.06 bash
47046 e14403    20   0   45400   6452   1952 S   0.0  0.0   0:00.05 bash
47381 e14403    20   0   45400   6452   1952 S   0.0  0.0   0:00.06 bash
47486 e14403    20   0   68060   5700   2900 S   0.0  0.0   0:00.03 vim
 5366 root      20   0   17104   5624      0 S   0.0  0.0   0:00.10 vmnet-dhcpd
48173 e15154    20   0   68004   5556   2924 T   0.0  0.0   0:00.04 vim
48288 e15154    20   0   67952   5464   2780 S   0.0  0.0   0:00.03 vim
 4792 lightdm   20   0  283540   5340   2668 S   0.0  0.0   0:00.00 indicator-power
48174 e15154    20   0   67884   5220   2672 T   0.0  0.0   0:00.02 vim
 4762 lightdm   20   0  337404   5212   2608 S   0.0  0.0   0:00.00 at-spi-bus-laun
 4777 lightdm   20   0  280140   5184   2540 S   0.0  0.0   0:00.00 gvfsd-fuse
46868 root      20   0  147788   4936   3692 S   0.0  0.0   0:00.01 sshd
47269 root      20   0  147788   4936   3692 S   0.0  0.0   0:00.01 sshd
47494 root      20   0  147784   4928   3692 S   0.0  0.0   0:00.01 sshd
 4795 lightdm   20   0  286804   4860   3812 S   0.0  0.0   0:00.00 indicator-appli
 4544 root      20   0  301672   4844   3424 S   0.0  0.0   1:08.71 accounts-daemon
48033 root      20   0  147788   4804   3568 S   0.0  0.0   0:00.01 sshd
46988 root      20   0  147788   4768   3532 S   0.0  0.0   0:00.00 sshd
47325 root      20   0  147788   4768   3532 S   0.0  0.0   0:00.01 sshd
47606 root      20   0  147784   4764   3532 S   0.0  0.0   0:00.00 sshd
```

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net

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

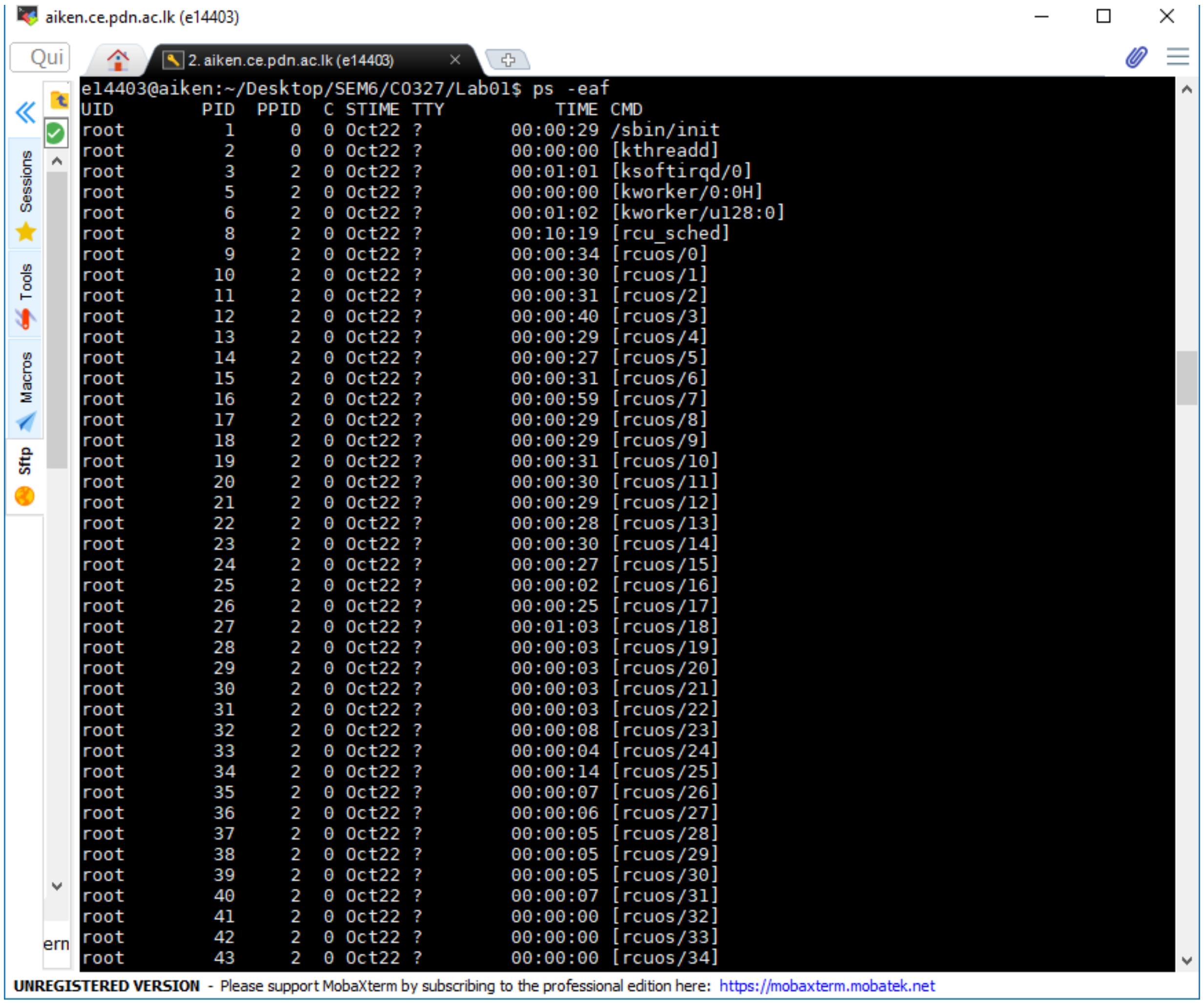
- ps -a => select all the processes except both session leaders and processes not associated with the terminal
- ps -x => view all processes owned by the current user



```
e14403@aiken:~/Desktop/SEM6/CO327/Lab01$ ps -a
  PID TTY          TIME CMD
47486 pts/3    00:00:00 vim
48493 pts/8    00:00:00 ps
e14403@aiken:~/Desktop/SEM6/CO327/Lab01$ ps -x
  PID TTY      STAT   TIME COMMAND
46987 ?        S      0:00 sshd: e14403@pts/1
47044 ?        S      0:00 sshd: e14403@notty
47045 ?        Ss     0:00 /usr/lib/openssh/sftp-server
47046 pts/1    Ss+    0:00 -bash
47324 ?        S      0:00 sshd: e14403@pts/3
47380 ?        S      0:00 sshd: e14403@notty
47381 pts/3    Ss     0:00 -bash
47382 ?        Ss     0:00 /usr/lib/openssh/sftp-server
47486 pts/3    S+     0:00 vim lab01.c
47605 ?        S      0:00 sshd: e14403@pts/8
47715 ?        S      0:00 sshd: e14403@notty
47726 ?        Ss     0:00 /usr/lib/openssh/sftp-server
47727 pts/8    Ss     0:00 -bash
48494 pts/8    R+     0:00 ps -x
e14403@aiken:~/Desktop/SEM6/CO327/Lab01$ ps -u
USER       PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
e14403   47046  0.0  0.0  45400  6452 pts/1    Ss+  05:49   0:00 -bash
e14403   47381  0.0  0.0  45400  6452 pts/3    Ss   06:21   0:00 -bash
e14403   47486  0.0  0.0  68060  5700 pts/3    S+   06:35   0:00 vim lab01.c
e14403   47727  0.0  0.0  45400  6456 pts/8    Ss   06:40   0:00 -bash
e14403   48495  0.0  0.0  38796  1416 pts/8    R+   07:43   0:00 ps -u
e14403@aiken:~/Desktop/SEM6/CO327/Lab01$ ps -w
  PID TTY          TIME CMD
47727 pts/8    00:00:00 bash
48496 pts/8    00:00:00 ps
e14403@aiken:~/Desktop/SEM6/CO327/Lab01$ █
```

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net

- ps -u => Select by effective user ID (EUID) or name. This selects the processes whose effective user name or ID is in user list.
- ps -w => adjusting the window size
- ps –eaf => gives all the processes in the order of the PID


aiken.ce.pdn.ac.lk (e14403)
2. aiken.ce.pdn.ac.lk (e14403)
Sessions
Tools
Macros
Sftp
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net

```
e14403@aiken:~/Desktop/SEM6/CO327/Lab01$ ps -eaf
UID        PID  PPID  C STIME TTY          TIME CMD
root         1     0  0 Oct22 ?        00:00:29 /sbin/init
root         2     0  0 Oct22 ?        00:00:00 [kthreadd]
root         3     2  0 Oct22 ?        00:01:01 [ksoftirqd/0]
root         5     2  0 Oct22 ?        00:00:00 [kworker/0:0H]
root         6     2  0 Oct22 ?        00:01:02 [kworker/u128:0]
root         8     2  0 Oct22 ?        00:10:19 [rcu_sched]
root         9     2  0 Oct22 ?        00:00:34 [rcuos/0]
root        10     2  0 Oct22 ?        00:00:30 [rcuos/1]
root        11     2  0 Oct22 ?        00:00:31 [rcuos/2]
root        12     2  0 Oct22 ?        00:00:40 [rcuos/3]
root        13     2  0 Oct22 ?        00:00:29 [rcuos/4]
root        14     2  0 Oct22 ?        00:00:27 [rcuos/5]
root        15     2  0 Oct22 ?        00:00:31 [rcuos/6]
root        16     2  0 Oct22 ?        00:00:59 [rcuos/7]
root        17     2  0 Oct22 ?        00:00:29 [rcuos/8]
root        18     2  0 Oct22 ?        00:00:29 [rcuos/9]
root        19     2  0 Oct22 ?        00:00:31 [rcuos/10]
root        20     2  0 Oct22 ?        00:00:30 [rcuos/11]
root        21     2  0 Oct22 ?        00:00:29 [rcuos/12]
root        22     2  0 Oct22 ?        00:00:28 [rcuos/13]
root        23     2  0 Oct22 ?        00:00:30 [rcuos/14]
root        24     2  0 Oct22 ?        00:00:27 [rcuos/15]
root        25     2  0 Oct22 ?        00:00:02 [rcuos/16]
root        26     2  0 Oct22 ?        00:00:25 [rcuos/17]
root        27     2  0 Oct22 ?        00:01:03 [rcuos/18]
root        28     2  0 Oct22 ?        00:00:03 [rcuos/19]
root        29     2  0 Oct22 ?        00:00:03 [rcuos/20]
root        30     2  0 Oct22 ?        00:00:03 [rcuos/21]
root        31     2  0 Oct22 ?        00:00:03 [rcuos/22]
root        32     2  0 Oct22 ?        00:00:08 [rcuos/23]
root        33     2  0 Oct22 ?        00:00:04 [rcuos/24]
root        34     2  0 Oct22 ?        00:00:14 [rcuos/25]
root        35     2  0 Oct22 ?        00:00:07 [rcuos/26]
root        36     2  0 Oct22 ?        00:00:06 [rcuos/27]
root        37     2  0 Oct22 ?        00:00:05 [rcuos/28]
root        38     2  0 Oct22 ?        00:00:05 [rcuos/29]
root        39     2  0 Oct22 ?        00:00:05 [rcuos/30]
root        40     2  0 Oct22 ?        00:00:07 [rcuos/31]
root        41     2  0 Oct22 ?        00:00:00 [rcuos/32]
root        42     2  0 Oct22 ?        00:00:00 [rcuos/33]
root        43     2  0 Oct22 ?        00:00:00 [rcuos/34]
```

The name of the process with PID = 1 is “init” (/sbin/init is the path)

Exercise 2:

```
int main(void) {

      int pid;

      pid = fork();

      if (pid < 0) {

      perror("fork");

      exit(1); }

      if (pid == 0)

      puts("This is the child process");

      else

      puts("This is the parent process");

return 0; }
```

I. In what order are the messages from parent and child printed? Is the order always the same?
“This is the parent process
This is the child process”
Order is same always.

II. How many children will the following program spawn? Draw a diagram illustrating the parent-child relationships between processes.

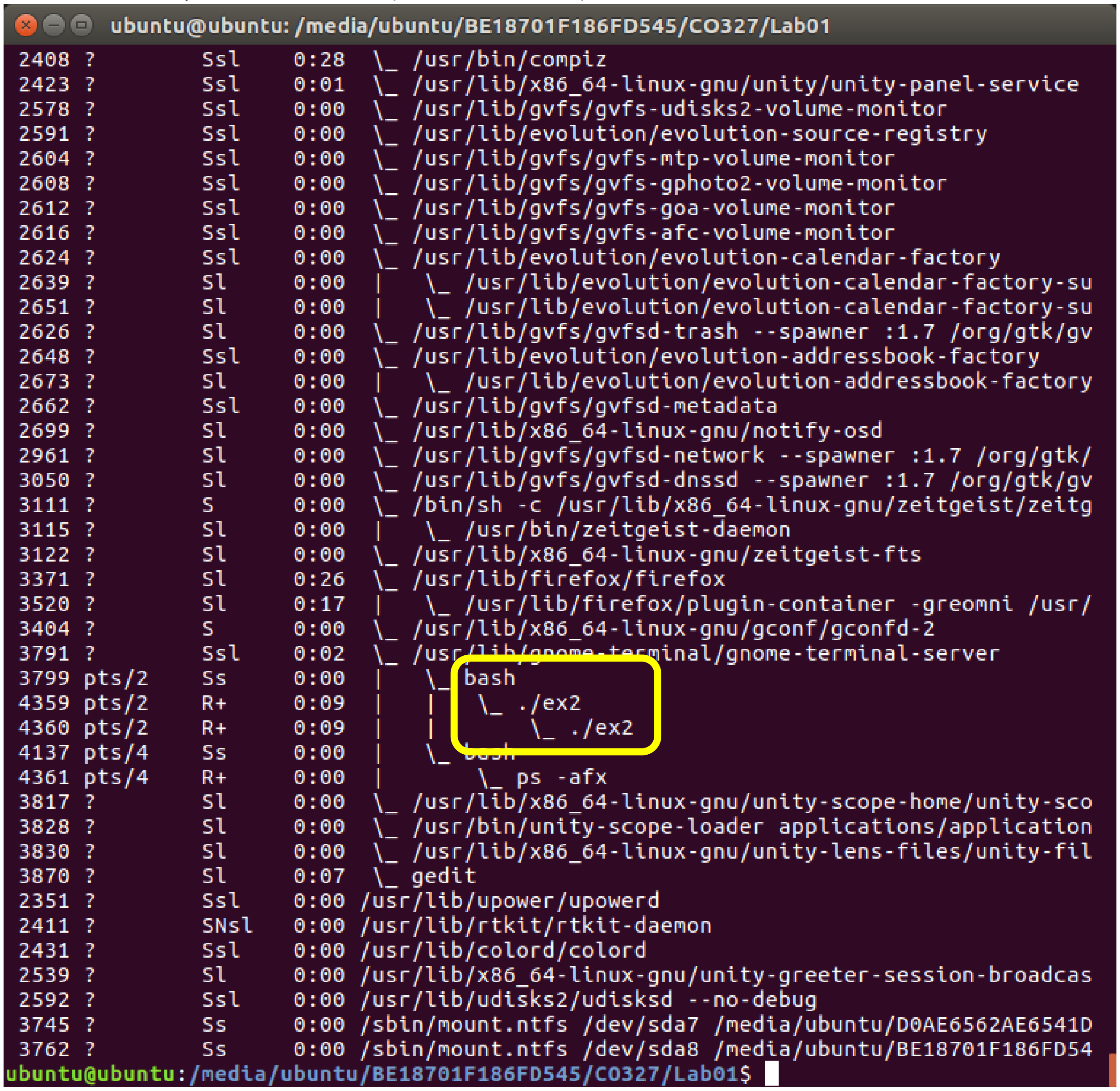
```
int main(void) {
      for (int i=0; i<3 ; i++)
         fork();
}
```

When we run the given code, it runs and dies within milliseconds. Therefore any information about the process is hard to gain. To avoid the particular situation code must be modified as follows.

```
#include<stdio.h>
int main(void)
{
   int i;
   for(i=0;i<3;i++)
   fork();
   while (1);

   return 0;
}
```

After running this code enter command ps -afx to visualize the processes.
Here is the results.

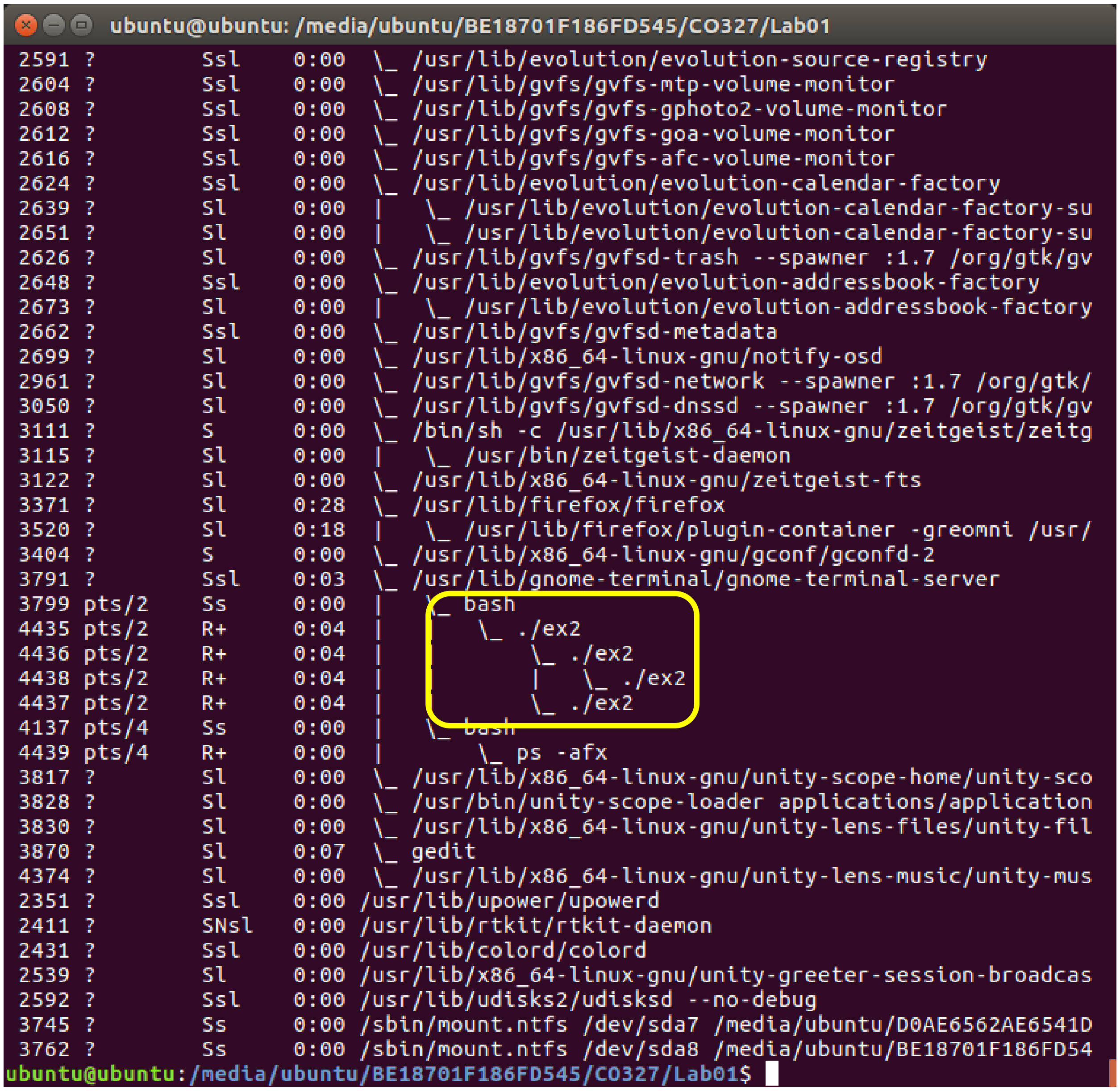
When loop reduced to once. (fork is done once)



```
ubuntu@ubuntu: /media/ubuntu/BE18701F186FD545/CO327/Lab01
 2408 ?        Ssl    0:28  \_ /usr/bin/compiz
 2423 ?        Ssl    0:01  \_ /usr/lib/x86_64-linux-gnu/unity/unity-panel-service
 2578 ?        Ssl    0:00  \_ /usr/lib/gvfs/gvfs-udisks2-volume-monitor
 2591 ?        Ssl    0:00  \_ /usr/lib/evolution/evolution-source-registry
 2604 ?        Ssl    0:00  \_ /usr/lib/gvfs/gvfs-mtp-volume-monitor
 2608 ?        Ssl    0:00  \_ /usr/lib/gvfs/gvfs-gphoto2-volume-monitor
 2612 ?        Ssl    0:00  \_ /usr/lib/gvfs/gvfs-goa-volume-monitor
 2616 ?        Ssl    0:00  \_ /usr/lib/gvfs/gvfs-afc-volume-monitor
 2624 ?        Ssl    0:00  \_ /usr/lib/evolution/evolution-calendar-factory
 2639 ?        Sl     0:00  |   \_ /usr/lib/evolution/evolution-calendar-factory-su
 2651 ?        Sl     0:00  |   \_ /usr/lib/evolution/evolution-calendar-factory-su
 2626 ?        Sl     0:00  \_ /usr/lib/gvfs/gvfsd-trash --spawner :1.7 /org/gtk/gv
 2648 ?        Ssl    0:00  \_ /usr/lib/evolution/evolution-addressbook-factory
 2673 ?        Sl     0:00  |   \_ /usr/lib/evolution/evolution-addressbook-factory
 2662 ?        Ssl    0:00  \_ /usr/lib/gvfs/gvfsd-metadata
 2699 ?        Sl     0:00  \_ /usr/lib/x86_64-linux-gnu/notify-osd
 2961 ?        Sl     0:00  \_ /usr/lib/gvfs/gvfsd-network --spawner :1.7 /org/gtk/
 3050 ?        Sl     0:00  \_ /usr/lib/gvfs/gvfsd-dnssd --spawner :1.7 /org/gtk/gv
 3111 ?        S      0:00  \_ /bin/sh -c /usr/lib/x86_64-linux-gnu/zeitgeist/zeitg
 3115 ?        Sl     0:00  |   \_ /usr/bin/zeitgeist-daemon
 3122 ?        Sl     0:00  \_ /usr/lib/x86_64-linux-gnu/zeitgeist-fts
 3371 ?        Sl     0:26  \_ /usr/lib/firefox/firefox
 3520 ?        Sl     0:17  |   \_ /usr/lib/firefox/plugin-container -greomni /usr/
 3404 ?        S      0:00  \_ /usr/lib/x86_64-linux-gnu/gconf/gconfd-2
 3791 ?        Ssl    0:02  \_ /usr/lib/gnome-terminal/gnome-terminal-server
 3799 pts/2    Ss     0:00  |   \_ bash
 4359 pts/2    R+     0:09  |   |   \_ ./ex2
 4360 pts/2    R+     0:09  |   |       \_ ./ex2
 4137 pts/4    Ss     0:00  |   \_ bash
 4361 pts/4    R+     0:00  |       \_ ps -afx
 3817 ?        Sl     0:00  \_ /usr/lib/x86_64-linux-gnu/unity-scope-home/unity-sco
 3828 ?        Sl     0:00  \_ /usr/bin/unity-scope-loader applications/application
 3830 ?        Sl     0:00  \_ /usr/lib/x86_64-linux-gnu/unity-lens-files/unity-fil
 3870 ?        Sl     0:07  \_ gedit
 2351 ?        Ssl    0:00 /usr/lib/upower/upowerd
 2411 ?        SNsl   0:00 /usr/lib/rtkit/rtkit-daemon
 2431 ?        Ssl    0:00 /usr/lib/colord/colord
 2539 ?        Sl     0:00 /usr/lib/x86_64-linux-gnu/unity-greeter-session-broadcas
 2592 ?        Ssl    0:00 /usr/lib/udisks2/udisksd --no-debug
 3745 ?        Ss     0:00 /sbin/mount.ntfs /dev/sda7 /media/ubuntu/D0AE6562AE6541D
 3762 ?        Ss     0:00 /sbin/mount.ntfs /dev/sda8 /media/ubuntu/BE18701F186FD54
ubuntu@ubuntu:/media/ubuntu/BE18701F186FD545/CO327/Lab01$
```

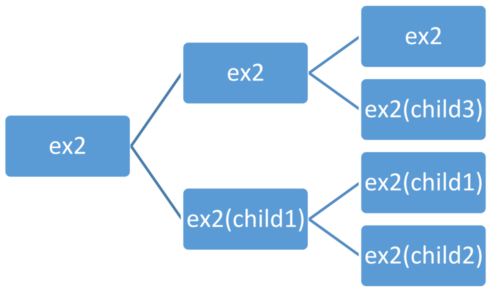
Two processes.

When looping happen twice. (Fork is called twice)

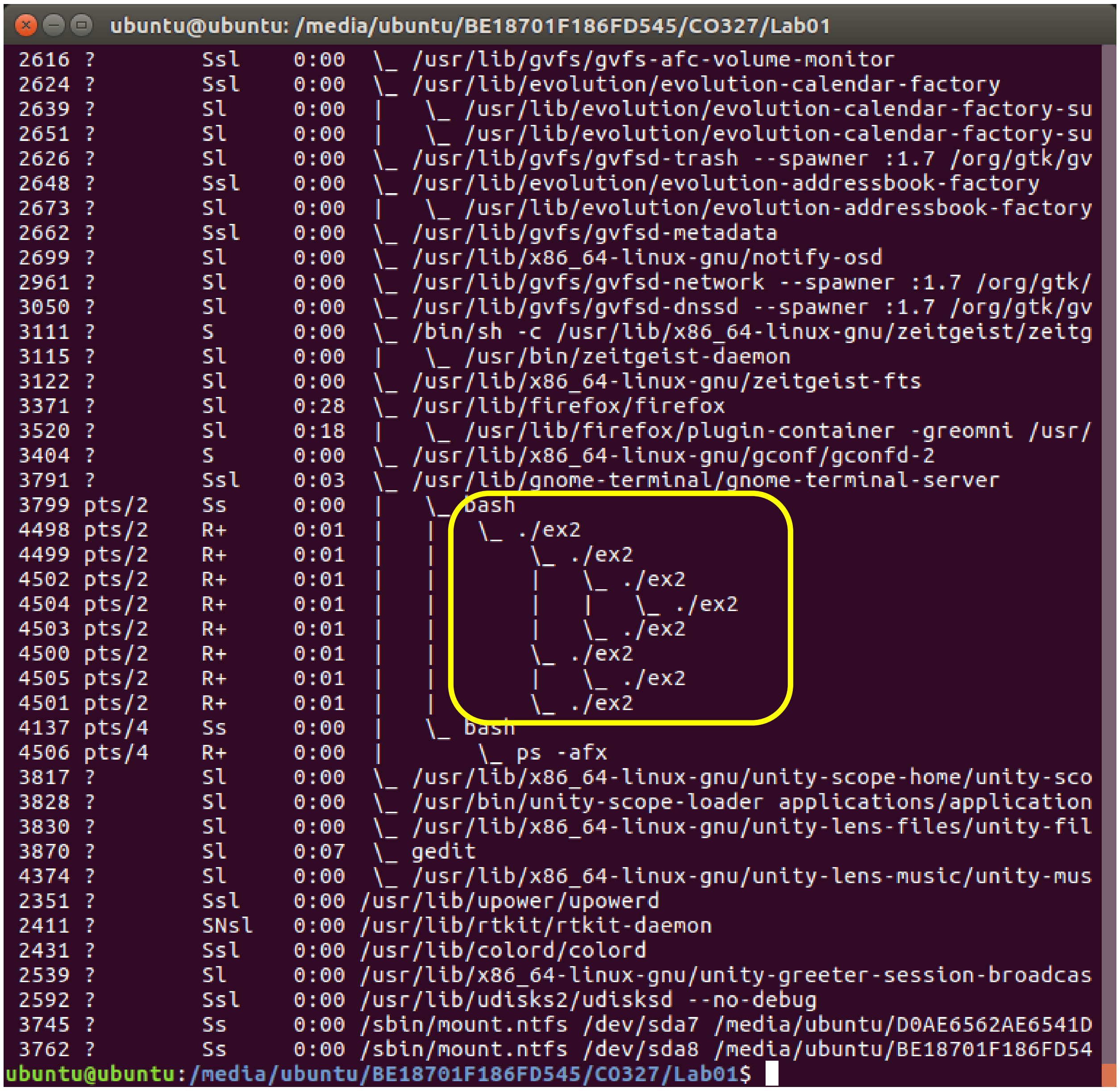


```
ubuntu@ubuntu: /media/ubuntu/BE18701F186FD545/CO327/Lab01
 2591 ?        Ssl    0:00  \_ /usr/lib/evolution/evolution-source-registry
 2604 ?        Ssl    0:00  \_ /usr/lib/gvfs/gvfs-mtp-volume-monitor
 2608 ?        Ssl    0:00  \_ /usr/lib/gvfs/gvfs-gphoto2-volume-monitor
 2612 ?        Ssl    0:00  \_ /usr/lib/gvfs/gvfs-goa-volume-monitor
 2616 ?        Ssl    0:00  \_ /usr/lib/gvfs/gvfs-afc-volume-monitor
 2624 ?        Ssl    0:00  \_ /usr/lib/evolution/evolution-calendar-factory
 2639 ?        Sl     0:00  |   \_ /usr/lib/evolution/evolution-calendar-factory-su
 2651 ?        Sl     0:00  |   \_ /usr/lib/evolution/evolution-calendar-factory-su
 2626 ?        Sl     0:00  \_ /usr/lib/gvfs/gvfsd-trash --spawner :1.7 /org/gtk/gv
 2648 ?        Ssl    0:00  \_ /usr/lib/evolution/evolution-addressbook-factory
 2673 ?        Sl     0:00  |   \_ /usr/lib/evolution/evolution-addressbook-factory
 2662 ?        Ssl    0:00  \_ /usr/lib/gvfs/gvfsd-metadata
 2699 ?        Sl     0:00  \_ /usr/lib/x86_64-linux-gnu/notify-osd
 2961 ?        Sl     0:00  \_ /usr/lib/gvfs/gvfsd-network --spawner :1.7 /org/gtk/
 3050 ?        Sl     0:00  \_ /usr/lib/gvfs/gvfsd-dnssd --spawner :1.7 /org/gtk/gv
 3111 ?        S      0:00  \_ /bin/sh -c /usr/lib/x86_64-linux-gnu/zeitgeist/zeitg
 3115 ?        Sl     0:00  |   \_ /usr/bin/zeitgeist-daemon
 3122 ?        Sl     0:00  \_ /usr/lib/x86_64-linux-gnu/zeitgeist-fts
 3371 ?        Sl     0:28  \_ /usr/lib/firefox/firefox
 3520 ?        Sl     0:18  |   \_ /usr/lib/firefox/plugin-container -greomni /usr/
 3404 ?        S      0:00  \_ /usr/lib/x86_64-linux-gnu/gconf/gconfd-2
 3791 ?        Ssl    0:03  \_ /usr/lib/gnome-terminal/gnome-terminal-server
 3799 pts/2    Ss     0:00  |   \_ bash
 4435 pts/2    R+     0:04  |   |   \_ ./ex2
 4436 pts/2    R+     0:04  |   |       \_ ./ex2
 4438 pts/2    R+     0:04  |   |       |   \_ ./ex2
 4437 pts/2    R+     0:04  |   |       \_ ./ex2
 4137 pts/4    Ss     0:00  |   \_ bash
 4439 pts/4    R+     0:00  |       \_ ps -afx
 3817 ?        Sl     0:00  \_ /usr/lib/x86_64-linux-gnu/unity-scope-home/unity-sco
 3828 ?        Sl     0:00  \_ /usr/bin/unity-scope-loader applications/application
 3830 ?        Sl     0:00  \_ /usr/lib/x86_64-linux-gnu/unity-lens-files/unity-fil
 3870 ?        Sl     0:07  \_ gedit
 4374 ?        Sl     0:00  \_ /usr/lib/x86_64-linux-gnu/unity-lens-music/unity-mus
 2351 ?        Ssl    0:00 /usr/lib/upower/upowerd
 2411 ?        SNsl   0:00 /usr/lib/rtkit/rtkit-daemon
 2431 ?        Ssl    0:00 /usr/lib/colord/colord
 2539 ?        Sl     0:00 /usr/lib/x86_64-linux-gnu/unity-greeter-session-broadcas
 2592 ?        Ssl    0:00 /usr/lib/udisks2/udisksd --no-debug
 3745 ?        Ss     0:00 /sbin/mount.ntfs /dev/sda7 /media/ubuntu/D0AE6562AE6541D
 3762 ?        Ss     0:00 /sbin/mount.ntfs /dev/sda8 /media/ubuntu/BE18701F186FD54
ubuntu@ubuntu:/media/ubuntu/BE18701F186FD545/CO327/Lab01$
```

Four ends. Which means 3 different child processes + original process.

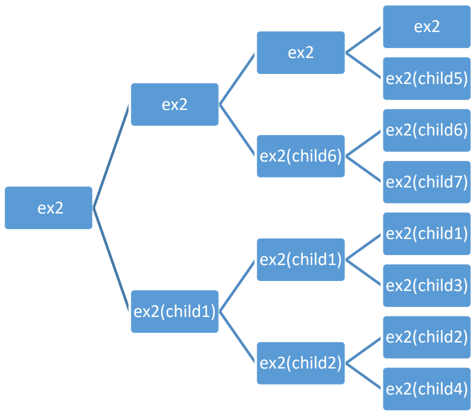


When looping happens 3 times. (Fork is called 3 times)



```
ubuntu@ubuntu: /media/ubuntu/BE18701F186FD545/CO327/Lab01
 2616 ?        Ssl    0:00  \_ /usr/lib/gvfs/gvfs-afc-volume-monitor
 2624 ?        Ssl    0:00  \_ /usr/lib/evolution/evolution-calendar-factory
 2639 ?        Sl     0:00  |   \_ /usr/lib/evolution/evolution-calendar-factory-su
 2651 ?        Sl     0:00  |   \_ /usr/lib/evolution/evolution-calendar-factory-su
 2626 ?        Sl     0:00  \_ /usr/lib/gvfs/gvfsd-trash --spawner :1.7 /org/gtk/gv
 2648 ?        Ssl    0:00  \_ /usr/lib/evolution/evolution-addressbook-factory
 2673 ?        Sl     0:00  |   \_ /usr/lib/evolution/evolution-addressbook-factory
 2662 ?        Ssl    0:00  \_ /usr/lib/gvfs/gvfsd-metadata
 2699 ?        Sl     0:00  \_ /usr/lib/x86_64-linux-gnu/notify-osd
 2961 ?        Sl     0:00  \_ /usr/lib/gvfs/gvfsd-network --spawner :1.7 /org/gtk/
 3050 ?        Sl     0:00  \_ /usr/lib/gvfs/gvfsd-dnssd --spawner :1.7 /org/gtk/gv
 3111 ?        S      0:00  \_ /bin/sh -c /usr/lib/x86_64-linux-gnu/zeitgeist/zeitg
 3115 ?        Sl     0:00  |   \_ /usr/bin/zeitgeist-daemon
 3122 ?        Sl     0:00  \_ /usr/lib/x86_64-linux-gnu/zeitgeist-fts
 3371 ?        Sl     0:28  \_ /usr/lib/firefox/firefox
 3520 ?        Sl     0:18  |   \_ /usr/lib/firefox/plugin-container -greomni /usr/
 3404 ?        S      0:00  \_ /usr/lib/x86_64-linux-gnu/gconf/gconfd-2
 3791 ?        Ssl    0:03  \_ /usr/lib/gnome-terminal/gnome-terminal-server
 3799 pts/2    Ss     0:00  |   \_ bash
 4498 pts/2    R+     0:01  |   |   \_ ./ex2
 4499 pts/2    R+     0:01  |   |       \_ ./ex2
 4502 pts/2    R+     0:01  |   |       |   \_ ./ex2
 4504 pts/2    R+     0:01  |   |       |   |   \_ ./ex2
 4503 pts/2    R+     0:01  |   |       |   \_ ./ex2
 4500 pts/2    R+     0:01  |   |       \_ ./ex2
 4505 pts/2    R+     0:01  |   |       |   \_ ./ex2
 4501 pts/2    R+     0:01  |   |       \_ ./ex2
 4137 pts/4    Ss     0:00  |   \_ bash
 4506 pts/4    R+     0:00  |       \_ ps -afx
 3817 ?        Sl     0:00  \_ /usr/lib/x86_64-linux-gnu/unity-scope-home/unity-sco
 3828 ?        Sl     0:00  \_ /usr/bin/unity-scope-loader applications/application
 3830 ?        Sl     0:00  \_ /usr/lib/x86_64-linux-gnu/unity-lens-files/unity-fil
 3870 ?        Sl     0:07  \_ gedit
 4374 ?        Sl     0:00  \_ /usr/lib/x86_64-linux-gnu/unity-lens-music/unity-mus
 2351 ?        Ssl    0:00 /usr/lib/upower/upowerd
 2411 ?        SNsl   0:00 /usr/lib/rtkit/rtkit-daemon
 2431 ?        Ssl    0:00 /usr/lib/colord/colord
 2539 ?        Sl     0:00 /usr/lib/x86_64-linux-gnu/unity-greeter-session-broadcas
 2592 ?        Ssl    0:00 /usr/lib/udisks2/udisksd --no-debug
 3745 ?        Ss     0:00 /sbin/mount.ntfs /dev/sda7 /media/ubuntu/D0AE6562AE6541D
 3762 ?        Ss     0:00 /sbin/mount.ntfs /dev/sda8 /media/ubuntu/BE18701F186FD54
ubuntu@ubuntu:/media/ubuntu/BE18701F186FD545/CO327/Lab01$
```

8 ends. Which means 7 child processes + original process.



Exercise 3: Modify the program in section 1.1 so that the parent always prints its message after the child. Refer to man 2 wait for details.

```
#include<stdio.h>

#include<stdlib.h>

#include<sys/types.h>

#include<sys/wait.h>

int main(void){

      int pid;

      pid  =  fork();

      wait(NULL);               //wait is put here

      if(pid  <  0){

            perror("fork");

            exit(1);

      }

      if(pid  ==  0)

            puts("This  is the  child  process");

      else

            puts("This  is the  parent  process");

      return 0;

}
```

Exercise 4:

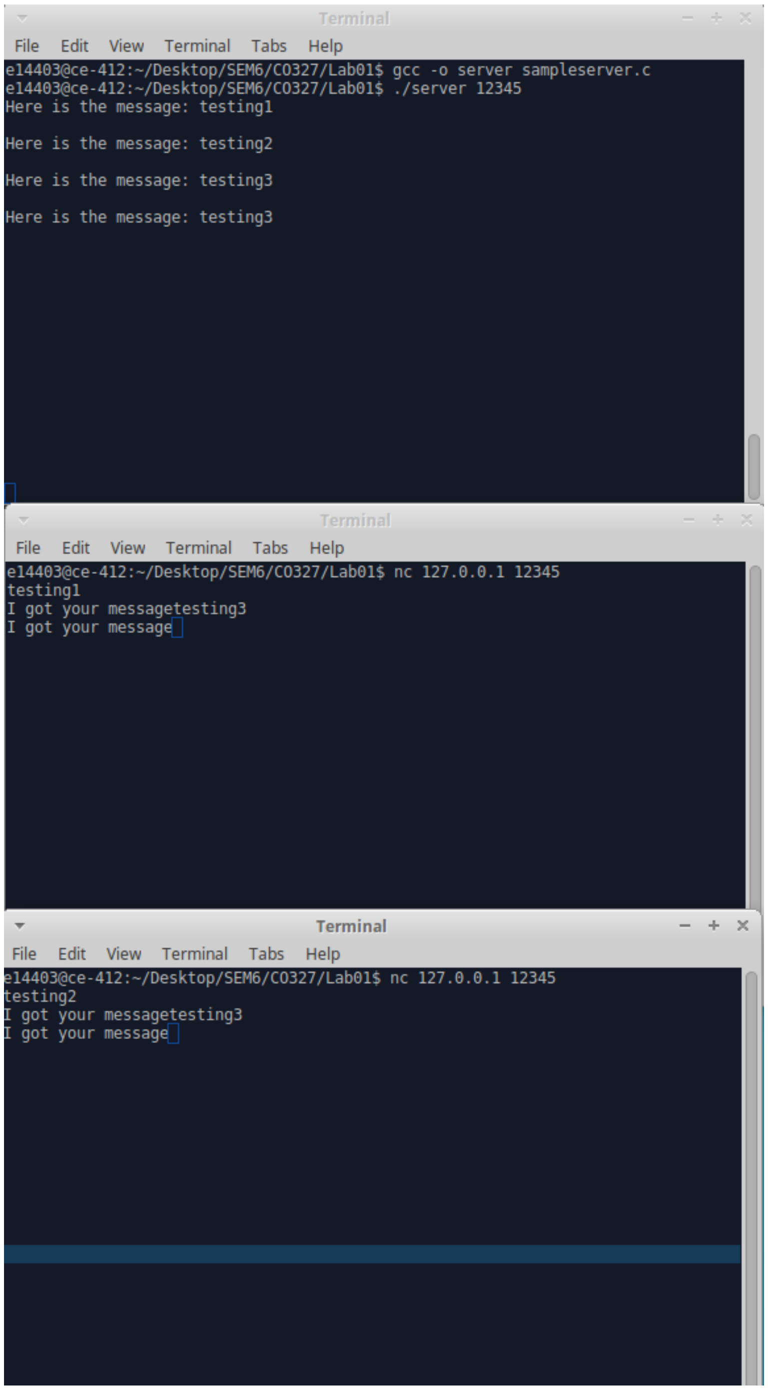
i. Compile and run the above code giving it a path as an argument. How many times is the message “Program ls has terminated” printed?
Message “Program ls has terminated” didn't appear at any moment.

ii. 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.

```
#include<stdio.h>
#include<stdlib.h>
#include<sys/types.h>
#include<sys/wait.h>
int main(char argc,char **argv){
       char str[50];
       int pid;
       while(1){
              pid  =  fork();
              wait(NULL);
              if(pid  <  0){
                     perror("fork");
                     exit(1);
              }
              if(pid  ==  0){
                     printf("Enter the command :");
                     scanf("%[^\n]", str);

                     execlp("/bin/sh","/bin/sh", "-c", str,(char *)NULL);
                     puts("Program has terminated");
              }
       }
       return 0;
}
```

Exercise 5:

i. 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.



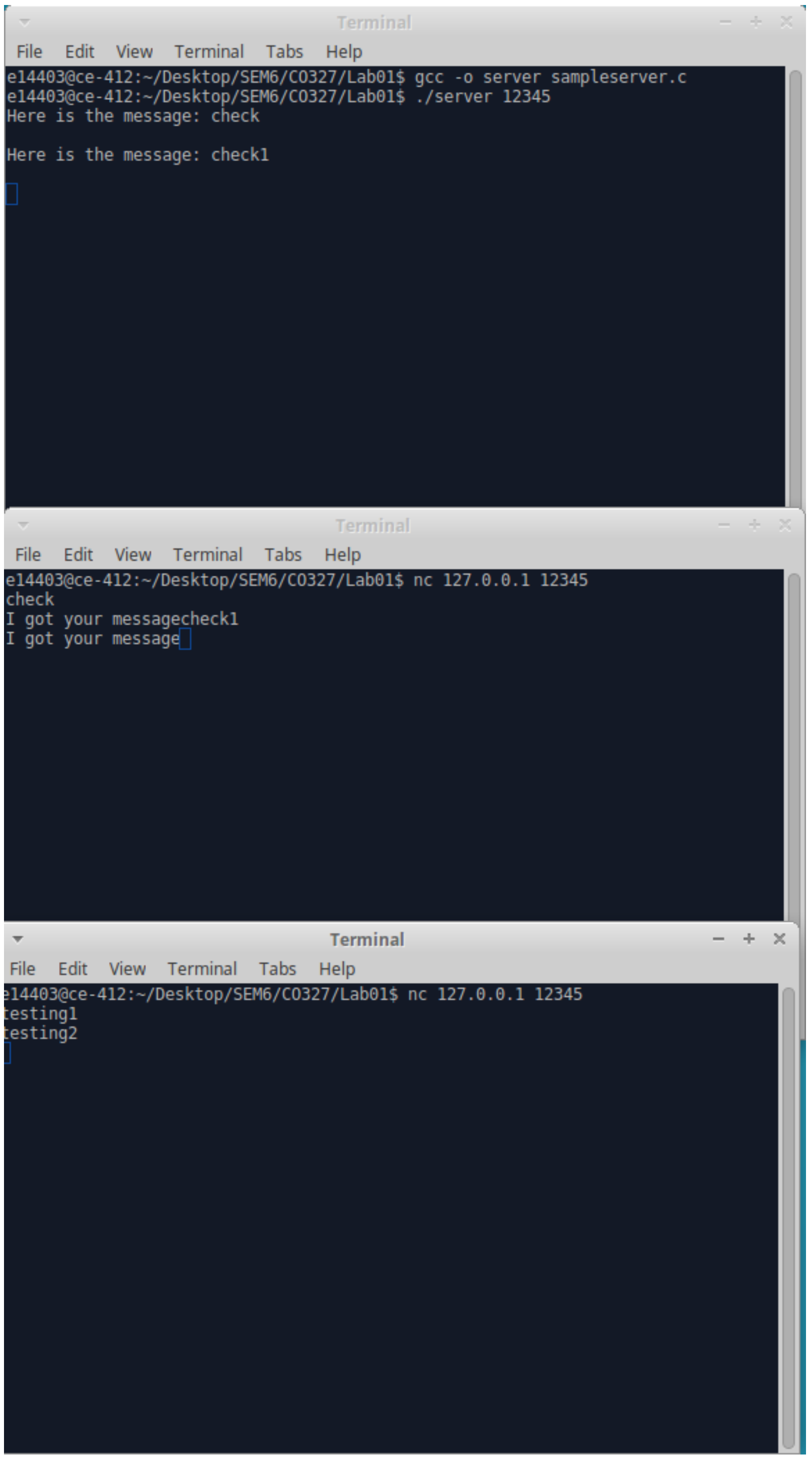
```
e14403@ce-412:~/Desktop/SEM6/CO327/Lab01$ gcc -o server sampleserver.c
e14403@ce-412:~/Desktop/SEM6/CO327/Lab01$ ./server 12345
Here is the message: testing1

Here is the message: testing2

Here is the message: testing3

Here is the message: testing3
```

```
e14403@ce-412:~/Desktop/SEM6/CO327/Lab01$ nc 127.0.0.1 12345
testing1
I got your messagetesting3
I got your message
```

```
e14403@ce-412:~/Desktop/SEM6/CO327/Lab01$ nc 127.0.0.1 12345
testing2
I got your messagetesting3
I got your message
```

When text in entered in the client terminals, server terminal shows them. Client terminal then displays "I got your message" which was sent by the server

ii. 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?


Terminal
File Edit View Terminal Tabs Help

```
e14403@ce-412:~/Desktop/SEM6/CO327/Lab01$ gcc -o server sampleserver.c
e14403@ce-412:~/Desktop/SEM6/CO327/Lab01$ ./server 12345
Here is the message: check

Here is the message: check1
```

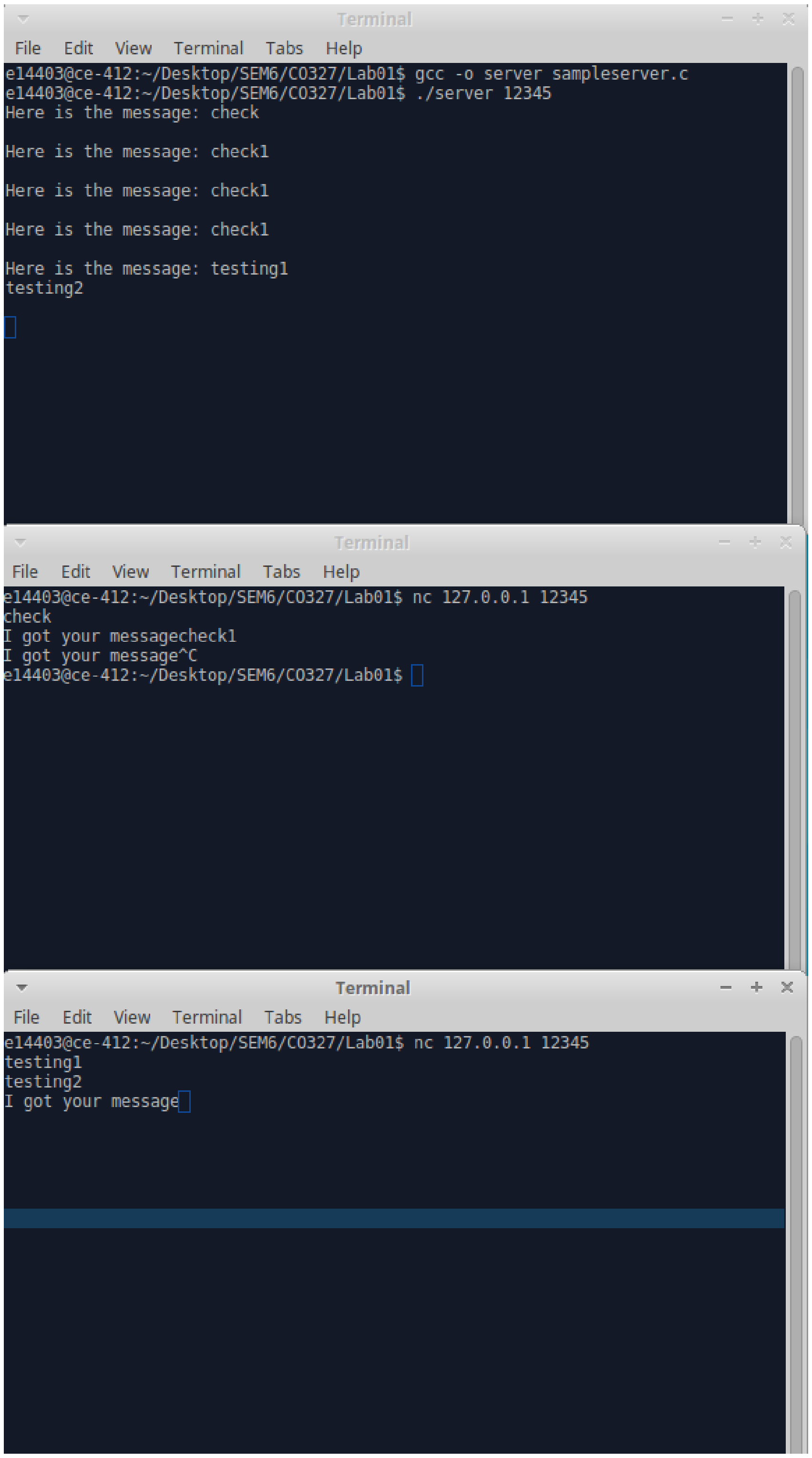
Terminal
File Edit View Terminal Tabs Help

```
e14403@ce-412:~/Desktop/SEM6/CO327/Lab01$ nc 127.0.0.1 12345
check
I got your messagecheck1
I got your message
```

Terminal
File Edit View Terminal Tabs Help

```
14403@ce-412:~/Desktop/SEM6/CO327/Lab01$ nc 127.0.0.1 12345
esting1
esting2
```

Server response only to the client terminal which started 1st.



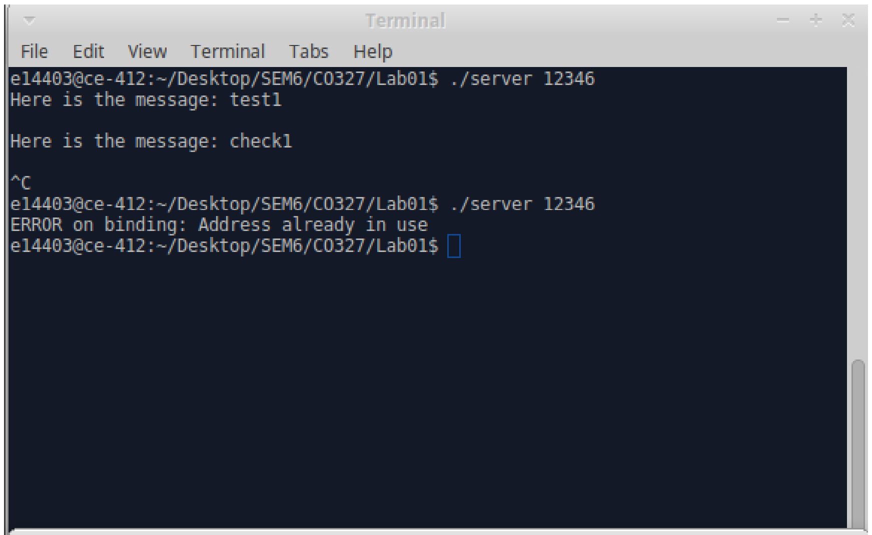
```
e14403@ce-412:~/Desktop/SEM6/CO327/Lab01$ gcc -o server sampleserver.c
e14403@ce-412:~/Desktop/SEM6/CO327/Lab01$ ./server 12345
Here is the message: check

Here is the message: check1

Here is the message: check1

Here is the message: check1

Here is the message: testing1
testing2
```

```
e14403@ce-412:~/Desktop/SEM6/CO327/Lab01$ nc 127.0.0.1 12345
check
I got your messagecheck1
I got your message^C
e14403@ce-412:~/Desktop/SEM6/CO327/Lab01$
```

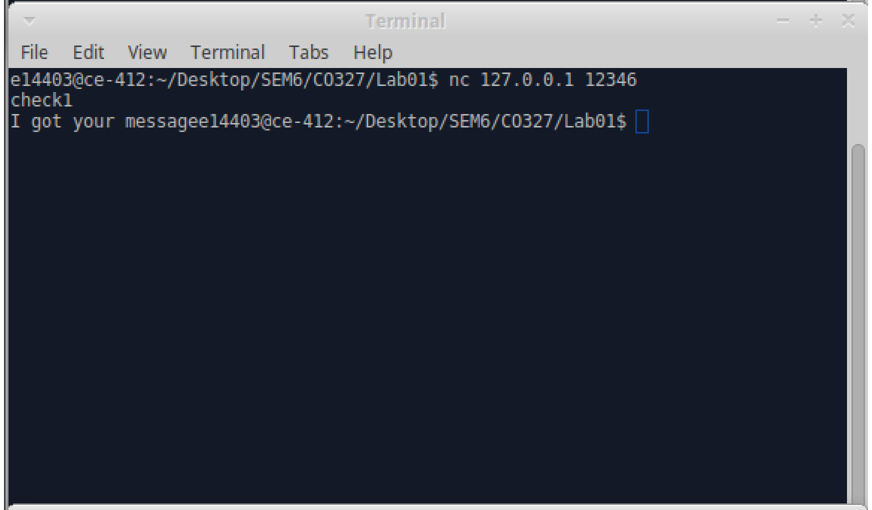
```
e14403@ce-412:~/Desktop/SEM6/CO327/Lab01$ nc 127.0.0.1 12345
testing1
testing2
I got your message
```

After terminating the 1st client's process then server starts to receive the messages sent by the client2.

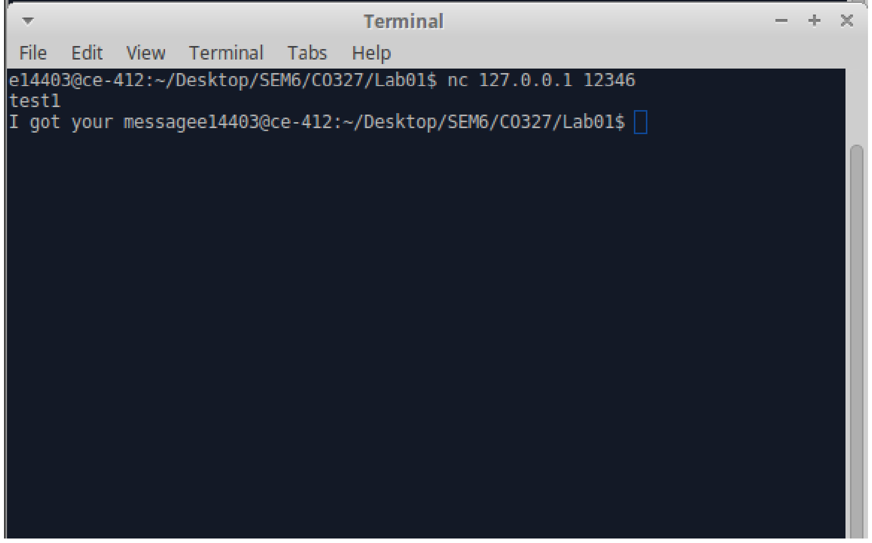
iii. What happens if you terminate the server while a client is connected, and then try to restart it? (Resolving this issue requires a signal handler.)



```
e14403@ce-412:~/Desktop/SEM6/CO327/Lab01$ ./server 12346
Here is the message: test1

Here is the message: check1

^C
e14403@ce-412:~/Desktop/SEM6/CO327/Lab01$ ./server 12346
ERROR on binding: Address already in use
e14403@ce-412:~/Desktop/SEM6/CO327/Lab01$
```



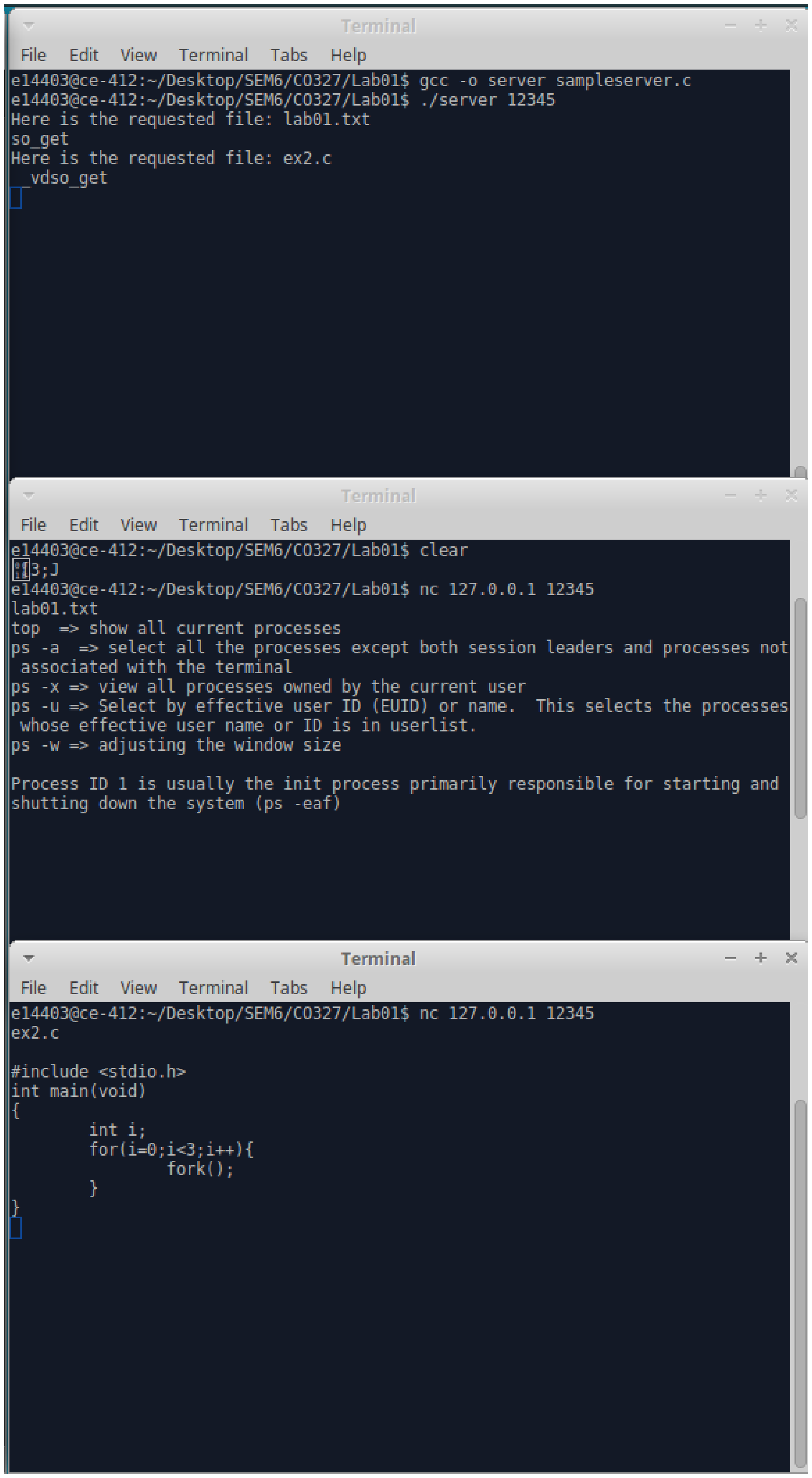
```
e14403@ce-412:~/Desktop/SEM6/CO327/Lab01$ nc 127.0.0.1 12346
check1
I got your messagee14403@ce-412:~/Desktop/SEM6/CO327/Lab01$
```



```
e14403@ce-412:~/Desktop/SEM6/CO327/Lab01$ nc 127.0.0.1 12346
test1
I got your messagee14403@ce-412:~/Desktop/SEM6/CO327/Lab01$
```

When you terminate the server process, communication between clients and server terminates. But the client side processes are bounded with the particular port. As the client side processes are still alive this error appears ("Address already in use").

iv. 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?


Terminal
File Edit View Terminal Tabs Help

```
e14403@ce-412:~/Desktop/SEM6/CO327/Lab01$ gcc -o server sampleserver.c
e14403@ce-412:~/Desktop/SEM6/CO327/Lab01$ ./server 12345
Here is the requested file: lab01.txt
so_get
Here is the requested file: ex2.c
__vdso_get
```

Terminal
File Edit View Terminal Tabs Help

```
e14403@ce-412:~/Desktop/SEM6/CO327/Lab01$ clear
[3;J
e14403@ce-412:~/Desktop/SEM6/CO327/Lab01$ nc 127.0.0.1 12345
lab01.txt
top  => show all current processes
ps -a  => select all the processes except both session leaders and processes not
 associated with the terminal
ps -x => view all processes owned by the current user
ps -u => Select by effective user ID (EUID) or name.  This selects the processes
 whose effective user name or ID is in userlist.
ps -w => adjusting the window size

Process ID 1 is usually the init process primarily responsible for starting and
shutting down the system (ps -eaf)
```

```
e14403@ce-412:~/Desktop/SEM6/CO327/Lab01$ nc 127.0.0.1 12345
ex2.c

#include <stdio.h>
int main(void)
{
        int i;
        for(i=0;i<3;i++){
                fork();
        }
}
```