CS266

ASSIGNMENT 3

NAME:

ARCHIT AGRAWAL

ROLL NO. :

202051213

SECTION:

2

1. ***Write a C program to create the process using fork() and display the parent and child process ID using getpid() and getppid() system calls, respectively.***

<u>Code</u>

```c
#include<stdio.h>
#include<sys/types.h>
#include<unistd.h>

int main(int argc, char const *argv[]){

        printf("\n");
        pid_t b;
        b = fork();

        //If control is returned to the newly created child process
        if(b == 0){
                printf("Child Process ID is : %d\n", getpid());
                printf("Parent Process ID is : %d\n", getpid());
        }
        //else if control is at parent process
        if(b == 0){
                printf("Child Process ID is : %d\n", b);
                printf("Parent Process ID is : %d\n", getpid());
        }

        return 0;
}
```

<u>Terminal</u>

```
archit@archit-VirtualBox:~/Desktop$ gcc q1.c -o q1
archit@archit-VirtualBox:~/Desktop$ ./q1

Child Process ID is : 6918
Parent Process ID is : 6918
Child Process ID is : 0
Parent Process ID is : 6918
archit@archit-VirtualBox:~/Desktop$
```

2. ***Write a C program to demonstrate the execl() system call for***

   (a) display the content of directory (command: ls)

   (b) display the process tree (command: pstree).

Code (a)

```c
#include<unistd.h>

int main(int argc, char const *argv[]){

        char *binaryPath = "/bin/ls";
        char *arg1 = "-a";
        char *arg2 = "-s";

        execl(binaryPath, binaryPath, arg1, arg2, NULL);
        return 0;

}
```

Terminal(a)

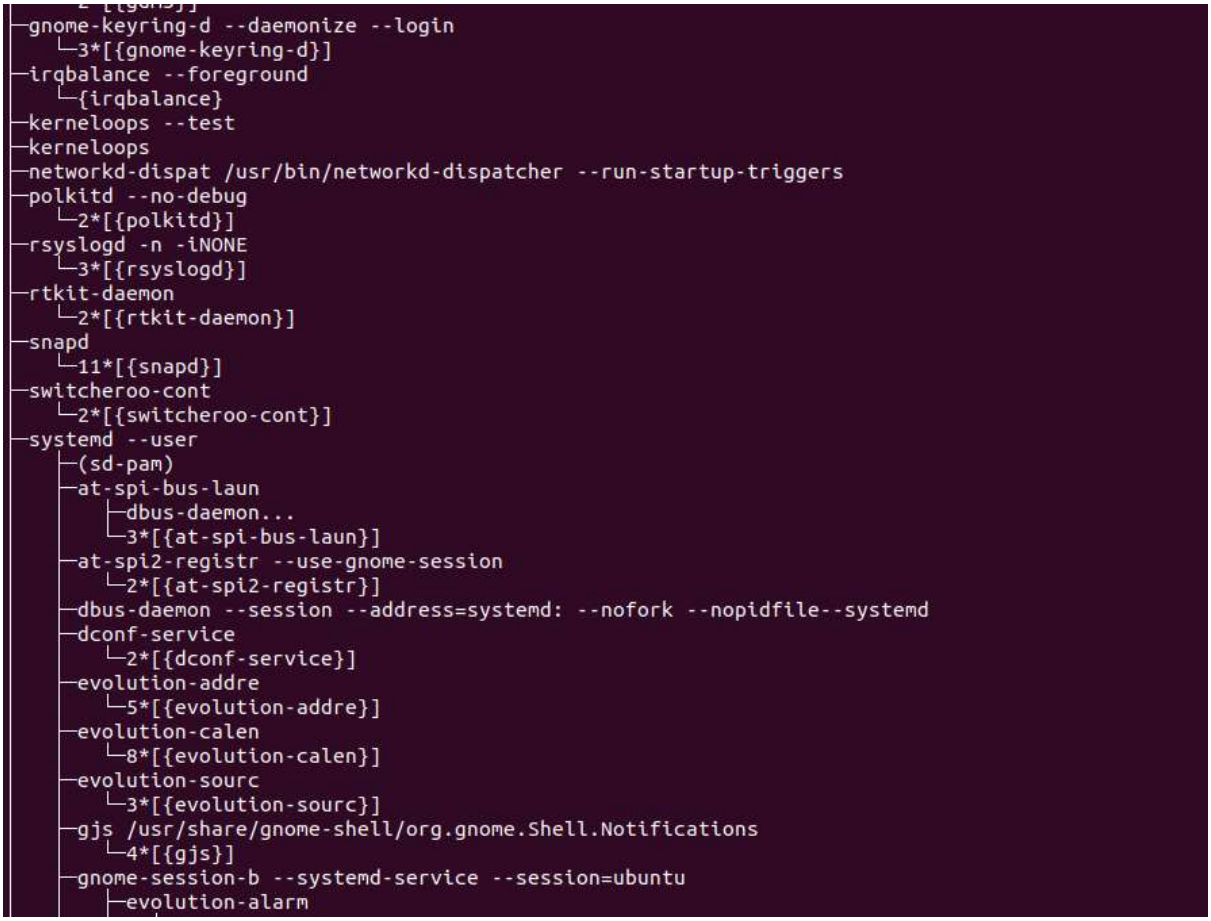
```
archit@archit-VirtualBox:~/Desktop$ gcc q2a.c -o q2a
archit@archit-VirtualBox:~/Desktop$ ./q2a
total 56
 4 .    4 ..   20 q1    4 q1.c   20 q2a    4 q2a.c
archit@archit-VirtualBox:~/Desktop$
```

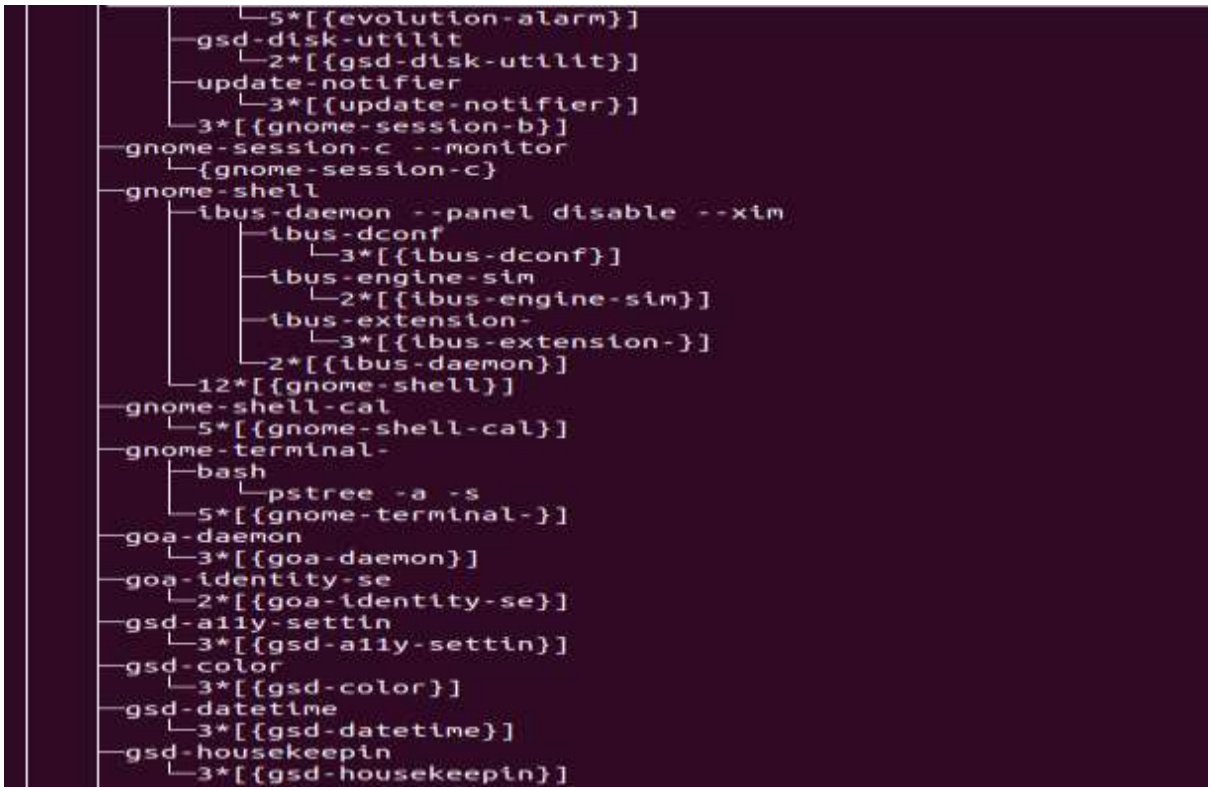
Code (b)

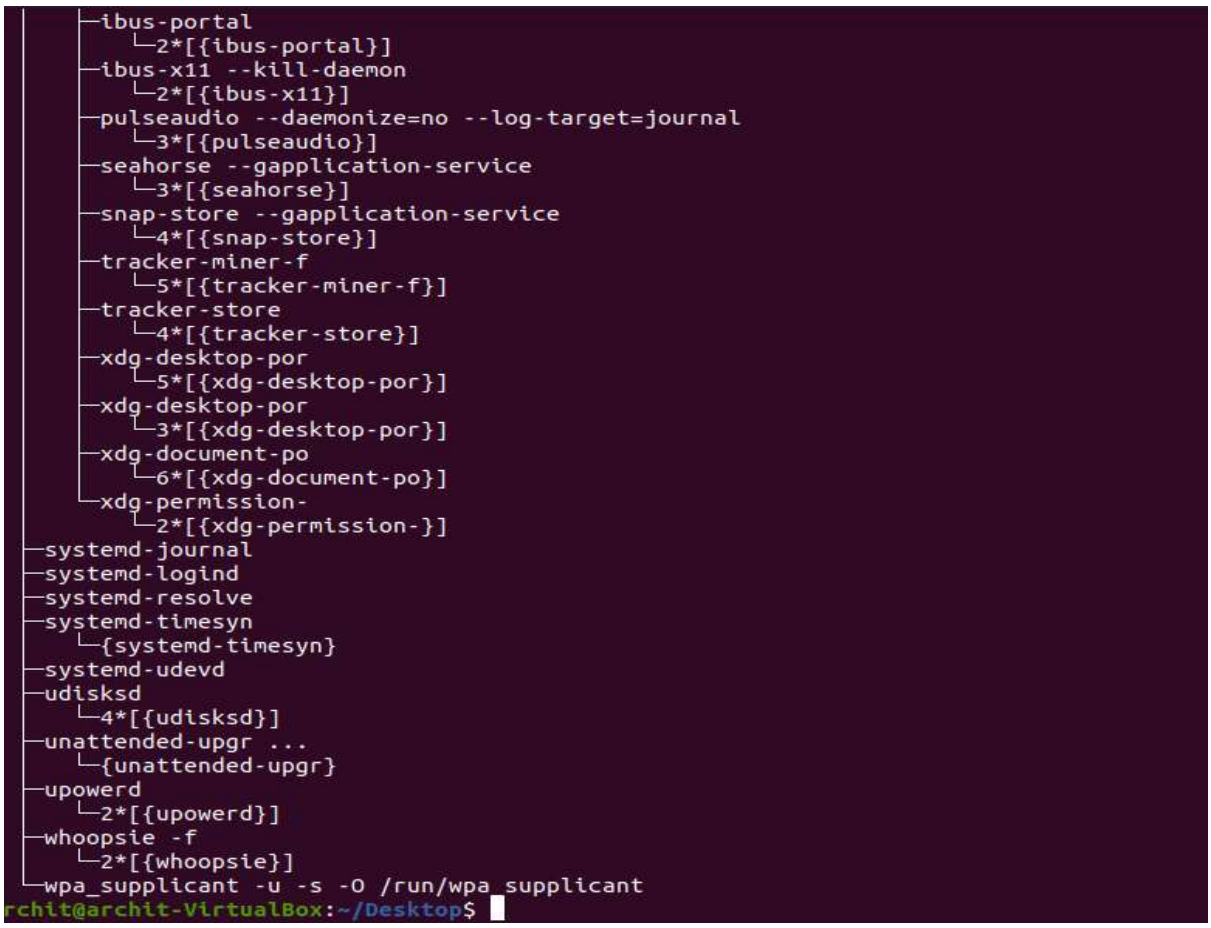
```c
#include<stdio.h>
#include<sys/types.h>
#include<unistd.h>

int main(int argc, char const *argv[]){

        char *binaryPath = "/bin/pstree";
        char *arg1 = "-a";
        char *arg2 = "-s";

        execl(binaryPath, binaryPath, arg1, arg2, NULL);
        return 0;

}

```

Terminal(b)


architt@archit-VirtualBox:~/Desktop$ gcc q2b.c -o q2b
architt@archit-VirtualBox:~/Desktop$ ./q2b
systemd splash
├─ModemManager
│ └─2*[{ModemManager}]
├─NetworkManager --no-daemon
│ └─2*[{NetworkManager}]
├─accounts-daemon
│ └─2*[{accounts-daemon}]
├─acpid
├─avahi-daemon
│ └─avahi-daemon
├─boltd
│ └─2*[{boltd}]
├─colord
│ └─2*[{colord}]
├─cron -f
├─cups-browsed
│ └─2*[{cups-browsed}]
├─cupsd -l
│ ├─dbus dbus://
│ └─dbus dbus://
├─dbus-daemon --system --address=systemd: --nofork --nopidfile--systemd-
├─fwupd
│ └─4*[{fwupd}]
├─gdm3
│ ├─gdm-session-wor
│ │ ├─gdm-x-session --run-scriptenv GNOME_SHELL_SESSION_MODE=ubuntu /us
│ │ │ ├─Xorg vt2 -displayfd 3 -auth /run/user/1000/gdm/Xauthority...
│ │ │ │ └─{Xorg}
│ │ │ ├─gnome-session-b --systemd --systemd --session=ubuntu
│ │ │ │ ├─ssh-agent /usr/bin/im-launch env ...
│ │ │ │ └─2*[{gnome-session-b}]
│ │ │ └─2*[{gdm-x-session}]
│ │ └─2*[{gdm-session-wor}]
│ └─2*[{gdm3}]
├─gnome-keyring-d --daemonize --login


├─gnome-keyring-d --daemonize --login
│ └─3*[{gnome-keyring-d}]
├─irqbalance --foreground
│ └─{irqbalance}
├─kerneloops --test
├─kerneloops
├─networkd-dispat /usr/bin/networkd-dispatcher --run-startup-triggers
├─polkitd --no-debug
│ └─2*[{polkitd}]
├─rsyslogd -n -iNONE
│ └─3*[{rsyslogd}]
├─rtkit-daemon
│ └─2*[{rtkit-daemon}]
├─snapd
│ └─11*[{snapd}]
├─switcheroo-cont
│ └─2*[{switcheroo-cont}]
├─systemd --user
│ ├─(sd-pam)
│ ├─at-spi-bus-laun
│ │ ├─dbus-daemon...
│ │ └─3*[{at-spi-bus-laun}]
│ ├─at-spi2-registr --use-gnome-session
│ │ └─2*[{at-spi2-registr}]
│ ├─dbus-daemon --session --address=systemd: --nofork --nopidfile--systemd
│ ├─dconf-service
│ │ └─2*[{dconf-service}]
│ ├─evolution-addre
│ │ └─5*[{evolution-addre}]
│ ├─evolution-calen
│ │ └─8*[{evolution-calen}]
│ ├─evolution-sourc
│ │ └─3*[{evolution-sourc}]
│ ├─gjs /usr/share/gnome-shell/org.gnome.Shell.Notifications
│ │ └─4*[{gjs}]
│ ├─gnome-session-b --systemd-service --session=ubuntu
│ │ ├─evolution-alarm


└─5*[{evolution-alarm}]
├─gsd-disk-utilit
└─2*[{gsd-disk-utilit}]
├─update-notifier
└─3*[{update-notifier}]
└─3*[{gnome-session-b}]
├─gnome-session-c --monitor
└─{gnome-session-c}
├─gnome-shell
├─ibus-daemon --panel disable --xim
├─ibus-dconf
└─3*[{ibus-dconf}]
├─ibus-engine-sim
└─2*[{ibus-engine-sim}]
├─ibus-extension-
└─3*[{ibus-extension-}]
└─2*[{ibus-daemon}]
└─12*[{gnome-shell}]
├─gnome-shell-cal
└─5*[{gnome-shell-cal}]
├─gnome-terminal-
├─bash
└─pstree -a -s
└─5*[{gnome-terminal-}]
├─goa-daemon
└─3*[{goa-daemon}]
├─goa-identity-se
└─2*[{goa-identity-se}]
├─gsd-a11y-settin
└─3*[{gsd-a11y-settin}]
├─gsd-color
└─3*[{gsd-color}]
├─gsd-datetime
└─3*[{gsd-datetime}]
├─gsd-housekeepin
└─3*[{gsd-housekeepin}]


├─ibus-portal
└─2*[{ibus-portal}]
├─ibus-x11 --kill-daemon
└─2*[{ibus-x11}]
├─pulseaudio --daemonize=no --log-target=journal
└─3*[{pulseaudio}]
├─seahorse --gapplication-service
└─3*[{seahorse}]
├─snap-store --gapplication-service
└─4*[{snap-store}]
├─tracker-miner-f
└─5*[{tracker-miner-f}]
├─tracker-store
└─4*[{tracker-store}]
├─xdg-desktop-por
└─5*[{xdg-desktop-por}]
├─xdg-desktop-por
└─3*[{xdg-desktop-por}]
├─xdg-document-po
└─6*[{xdg-document-po}]
└─xdg-permission-
└─2*[{xdg-permission-}]
├─systemd-journal
├─systemd-logind
├─systemd-resolve
├─systemd-timesyn
└─{systemd-timesyn}
├─systemd-udevd
├─udisksd
└─4*[{udisksd}]
├─unattended-upgr ...
└─{unattended-upgr}
├─upowerd
└─2*[{upowerd}]
├─whoopsie -f
└─2*[{whoopsie}]
└─wpa_supplicant -u -s -O /run/wpa_supplicant
rchit@archit-VirtualBox:~/Desktop$

3. ***Write a C program to demonstrate wait and sleep system calls return the pid of the child that terminated.***

Code

```c
#include<stdio.h>
#include<sys/types.h>
#include<sys/wait.h>
#include<unistd.h>

int main(int argc, char const *argv[]){
        printf("\n");
        if(fork() == 0){
                printf("Child process is sleeping for 10 seconds\n");
                sleep(10);
                printf("Child process is awake now\n");
        } else {
                printf("This is parent process with ID : %d\n", getpid());
                printf("Parent process is waiting for child process to terminate\n");
                pid_t childPid = wait(NULL);
                printf("Child process with PID %d has been terminated\n", childPid);
        }

        return 0;
}
```

Terminal

```
archit@archit-VirtualBox:~/Desktop$ gcc q3.c -o q3
archit@archit-VirtualBox:~/Desktop$ ./q3

This is parent process with ID : 7710
Parent process is waiting for child process to terminate
Child process is sleeping for 10 seconds
Child process is awake now
Child process with PID 7711 has been terminated
archit@archit-VirtualBox:~/Desktop$
```

4. ***Write a C program to create the multiple processes using fork() and display the process IDs and their parent process IDs in hierarchy.***

Code

```c
#include<stdio.h>
#include<unistd.h>

int main(){
        //Creating first child
        printf("\n");
        int n1 = fork();
        int n2 = fork();

        if(n1 > 0 && n2 > 0){
                printf("parent\n");
                printf("%d %d\n", n1, n2);
                printf("My ID is %d\n", getpid());
                printf("My parent ID is %d \n", getppid());
        } else if(n1 == 0 && n2 > 0){
                printf("First Child\n");
                printf("%d %d\n", n1, n2);
                printf("My ID is %d\n", getpid());
                printf("My parent ID is %d \n", getppid());
        } else if(n1 > 0 && n2 == 0){
                printf("Second Child\n");
                printf("%d %d\n", n1, n2);
                printf("My ID is %d\n", getpid());
                printf("My parent ID is %d \n", getppid());
        } else{
                printf("Third Child\n");
                printf("%d %d\n", n1, n2);
                printf("My ID is %d\n", getpid());
                printf("My parent ID is %d \n", getppid());
        }

        return 0;
}
```

Terminal

```
archit@archit-VirtualBox:~/Desktop$ ./q4

parent
8206 8207
My ID is 8205
My parent ID is 8056
First Child
0 8208
My ID is 8206
My parent ID is 8205
Second Child
8206 0
My ID is 8207
My parent ID is 737
archit@archit-VirtualBox:~/Desktop$ Third Child
0 0
My ID is 8208
My parent ID is 737
```

5. ***Write C program to display the directory content using readdir system call.***

Code

```c
#include<stdio.h>
#include<sys/types.h>
#include<unistd.h>
#include<dirent.h>

int main(int argc, char const *argv[]){
printf("\n");
        struct dirent *de;

        DIR *dir = opendir(".");

        if(dir == NULL){
                printf("Unable to open current directory");
                return 0;
        }

        while((de = readdir(dir)) != NULL){
                printf("%s\n", de -> d_name);
        }

        closedir(dir);
        return 0;
}
```

Terminal

```
architt@architt-VirtualBox:~/Desktop$ gcc q5.c -o q5
architt@architt-VirtualBox:~/Desktop$ ./q5

q5.c
q2a.c
q4.c
q5
q2b
q1.c
q2a
q3
..
q1
q4
q2b.c
.
q3.c
architt@architt-VirtualBox:~/Desktop$
```

6. ***Write a c program to interrupt and terminate the current process using signal handlers. (use SIGINT i.e., Ctrl-C to interrupt and Ctrl-\to kill the process)***

Code

```c
#include<stdio.h>
#include<signal.h>
#include<unistd.h>

void sig_handler(int signo){
        if(signo == SIGINT){
                printf("Current Process Interrupted\n");
        } else if(signo == SIGQUIT){
                printf("Process Terminated\n");
                kill(getpid(), SIGSEGV);
        }
}

int main(void){
        printf("\n");
        signal(SIGINT, sig_handler);
        signal(SIGQUIT, sig_handler);

        while(1){
                printf("Processing...\n");
                sleep(3);
        }

        return 0;
}
```

Terminal

```
architt@architt-VirtualBox:~$ cd Desktop/
architt@architt-VirtualBox:~/Desktop$ ./q6

Processing...
Processing...
Processing...
Processing...
Processing...
Processing...
Processing...
Processing...
Processing...
Processing...
Processing...
Processing...
Processing...
^CCurrent Process Interrupted
Processing...
Processing...
Processing...
Processing...
Processing...
^CCurrent Process Interrupted
Processing...
^TProcessing...
Processing...
Processing...
^\Process Terminated
Segmentation fault (core dumped)
architt@architt-VirtualBox:~/Desktop$
```