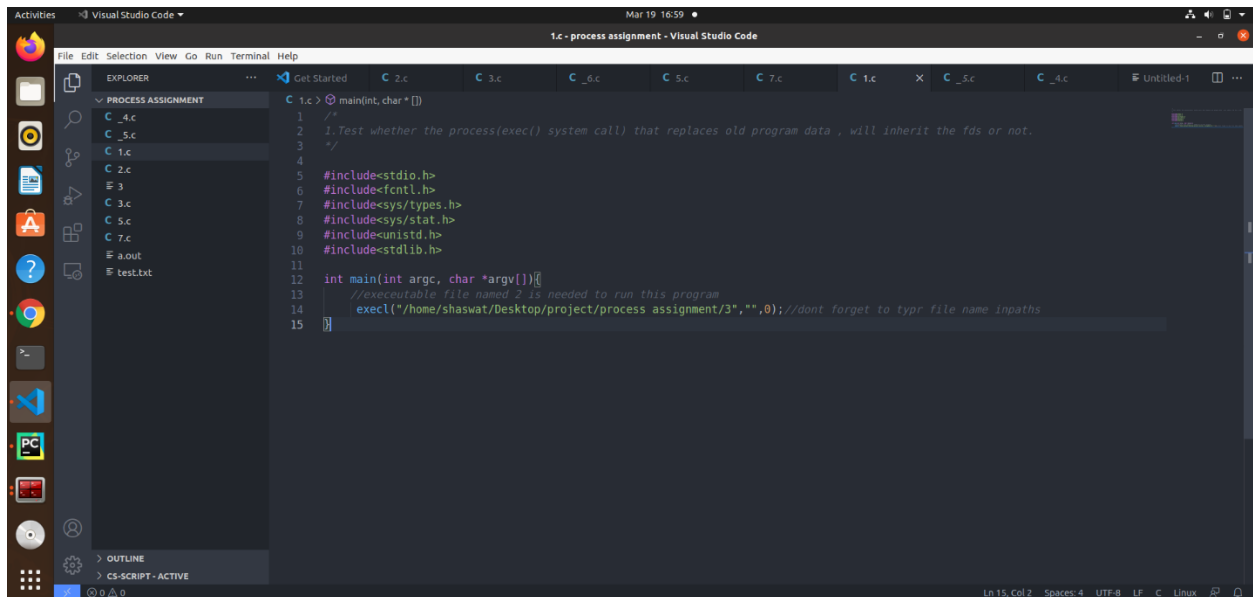
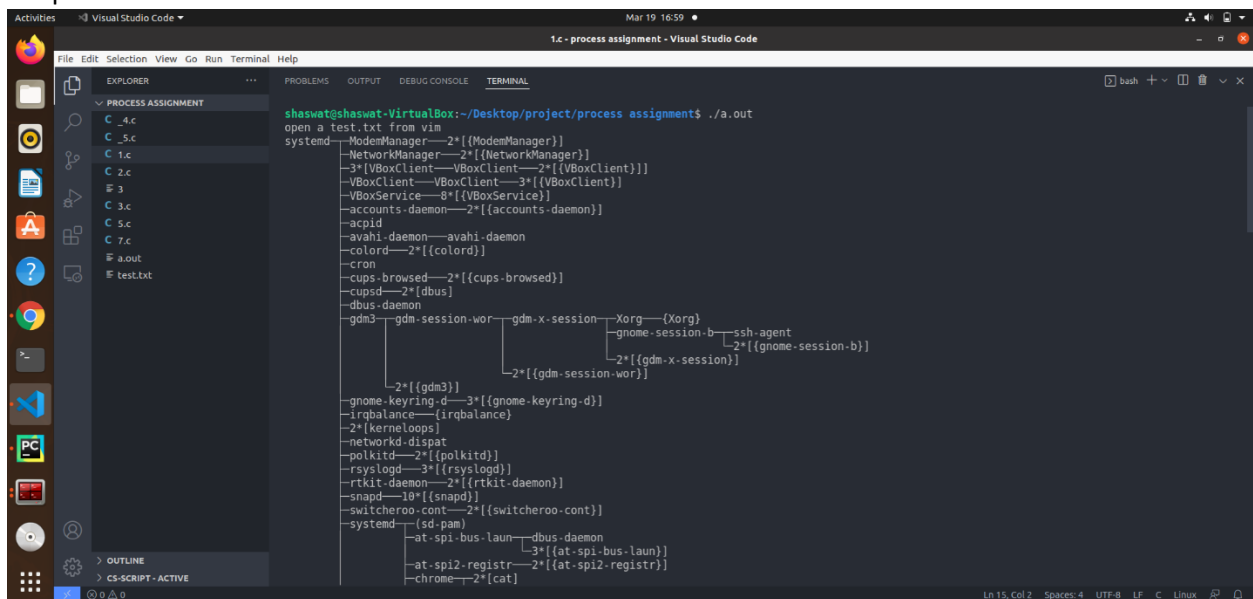


1. Test whether the process(exec()) system call) that replaces old program data , will inherit the fd's or not.



```
1.c > main(int, char*[])
1 /*
2 1. Test whether the process(exec()) system call) that replaces old program data , will inherit the fds or not.
3 */
4
5 #include<stdio.h>
6 #include<fcntl.h>
7 #include<sys/types.h>
8 #include<sys/stat.h>
9 #include<unistd.h>
10 #include<stdlib.h>
11
12 int main(int argc, char *argv[])
13 {
14     //executable file named 2 is needed to run this program
15     execl("/home/shaswat/Desktop/project/process assignment/3","",0); //dont forget to typr file name inpaths
}
```

Output:-



```
shaswat@shaswat-VirtualBox:~/Desktop/project/process assignment$ ./a.out
open a test.txt from vim
systemd--2*[{ModemManager}]
--2*[{NetworkManager}]
--3*[{VBoxClient}]--2*[{VBoxClient}]
--VBoxClient--VBoxClient--3*[{VBoxClient}]
--VBoxService--6*[{VBoxService}]
--accounts-daemon--2*[{accounts-daemon}]
--acpid
--avahi-daemon--avahi-daemon
--colord--2*[{colord}]
--cron
--cups-browsed--2*[{cups-browsed}]
--cupsd--2*[{dbus}]
--dbus-daemon
--gdm3--gdm-session-wor--gdm-x-session--Xorg--(Xorg)
--gnome-session-b--ssh-agent
--2*[{gnome-session-b}]
--2*[{gdm-session-wor}]
--2*[{gdm3}]
--gnome-keyring-d--3*[{gnome-keyring-d}]
--irqbalance--(irqbalance)
--2*[{kerneloops}]
--networkd-dispatcher
--polkitd--2*[{polkitd}]
--rsyslogd--3*[{rsyslogd}]
--rtkit-daemon--2*[{rtkit-daemon}]
--snapd--10*[{snapd}]
--switcheroo-control--2*[{switcheroo-control}]
--systemd
--at-spi-bus-launcher--dbus-daemon
--3*[{at-spi-bus-launcher}]
--at-spi2-registrar--2*[{at-spi2-registrar}]
--chrome--2*[{cat}]
```

2. Write a program such that parent process create two child processes, such that each child executes a separate task.

The screenshot shows the Visual Studio Code editor with a C file named `2.c` open. The code is as follows:

```
1 //2. Write a program such that parent process create two child processes, such that each child
2 //executes a separate task.
3
4
5 #include<stdio.h>
6
7
8 int main()
9 {
10     for(int i=0; i<2; i++)
11     {
12         if(fork() == 0)
13         {
14             printf("[son] pid %d from [parent] pid %d\n", getpid(), getppid());
15             exit(0);
16         }
17     }
18     for(int i=0; i<2; i++)
19     {
20         wait(NULL);
21     }
22 }
```

The Explorer sidebar on the left shows a project named "PROCESS ASSIGNMENT" with files `_4.c`, `_5.c`, `1.c`, `2.c`, `3.c`, `5.c`, `7.c`, `a.out`, and `test.txt`. The status bar at the bottom indicates "Ln 22, Col 1", "Tab Size: 4", "UTF-8", "LF", "C", "Linux".

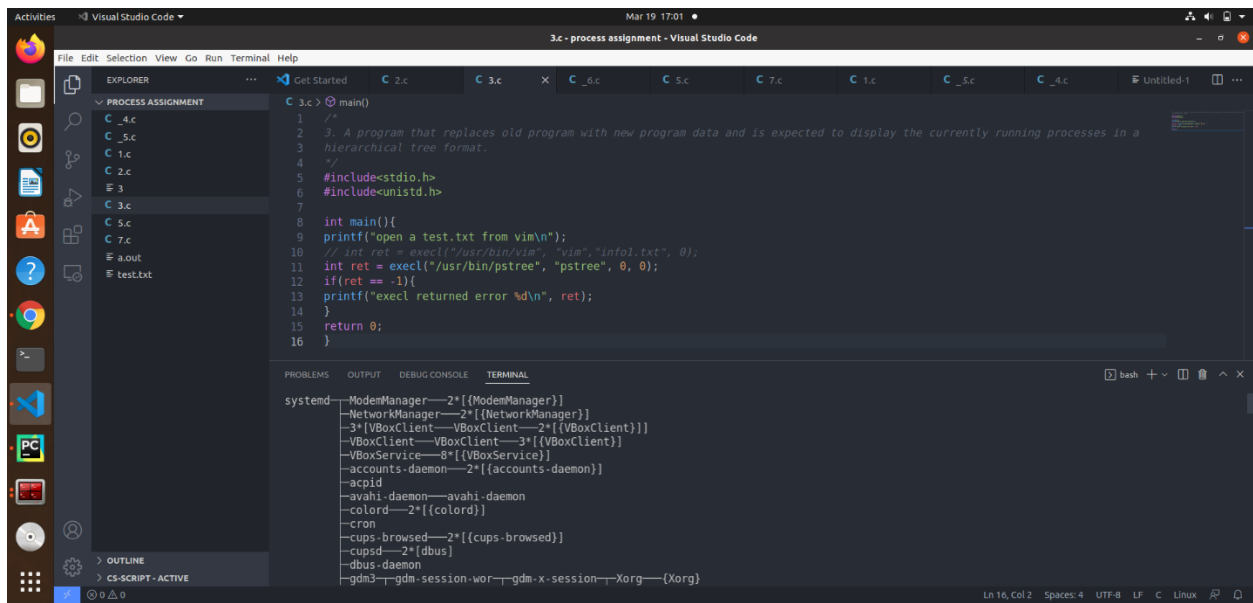
Output:-

The screenshot shows the Visual Studio Code editor with the "TERMINAL" view active. The terminal output is as follows:

```
shaswat@shaswat-VirtualBox:~/Desktop/project/process assignment$ ./a.out
[son] pid 15342 from [parent] pid 15340
[son] pid 15341 from [parent] pid 15340
shaswat@shaswat-VirtualBox:~/Desktop/project/process assignment$
```

The Explorer sidebar on the left is the same as in the first screenshot. The status bar at the bottom indicates "Ln 22, Col 1", "Tab Size: 4", "UTF-8", "LF", "C", "Linux".

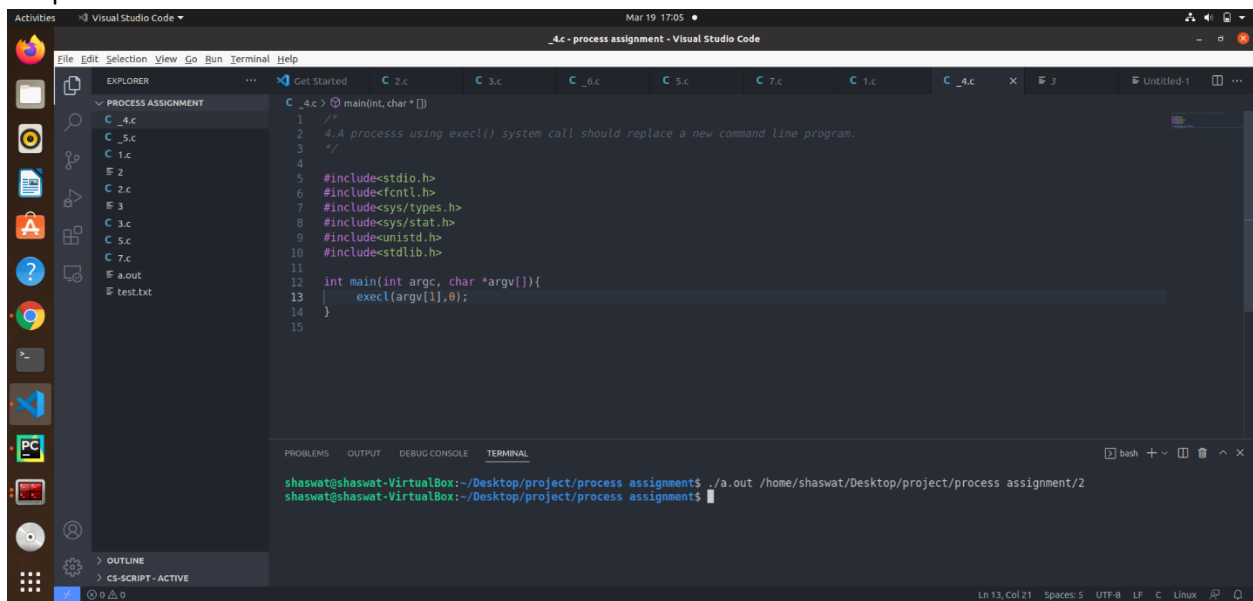
3. 3. A program that replaces old program with new program data and is expected to display the currently running processes in a hierarchical tree format.



```
1 /*
2 3. A program that replaces old program with new program data and is expected to display the currently running processes in a
3 hierarchical tree format.
4 */
5 #include<stdio.h>
6 #include<unistd.h>
7
8 int main(){
9     printf("open a test.txt from vim\n");
10    // int ret = execl("/usr/bin/vim", "vim", "info1.txt", 0);
11    int ret = execl("/usr/bin/pstree", "pstree", 0, 0);
12    if(ret == -1){
13        printf("execl returned error %d\n", ret);
14    }
15    return 0;
16 }
```

```
systemd--ModemManager--2*[{ModemManager}]
--NetworkManager--2*[{NetworkManager}]
--3*[VBoxClient--VBoxClient--2*[{VBoxClient}]]
--VBoxClient--VBoxClient--3*[{VBoxClient}]
--VBoxService--8*[{VBoxService}]
--accounts-daemon--2*[{accounts-daemon}]
--acpid
--avahi-daemon--avahi-daemon
--colord--2*[{colord}]
--cron
--cups-browsed--2*[{cups-browsed}]
--cupsd--2*[{dbus}]
--dbus-daemon
--gdm3--gdm-session-wor--gdm-x-session--Xorg--(Xorg)
```

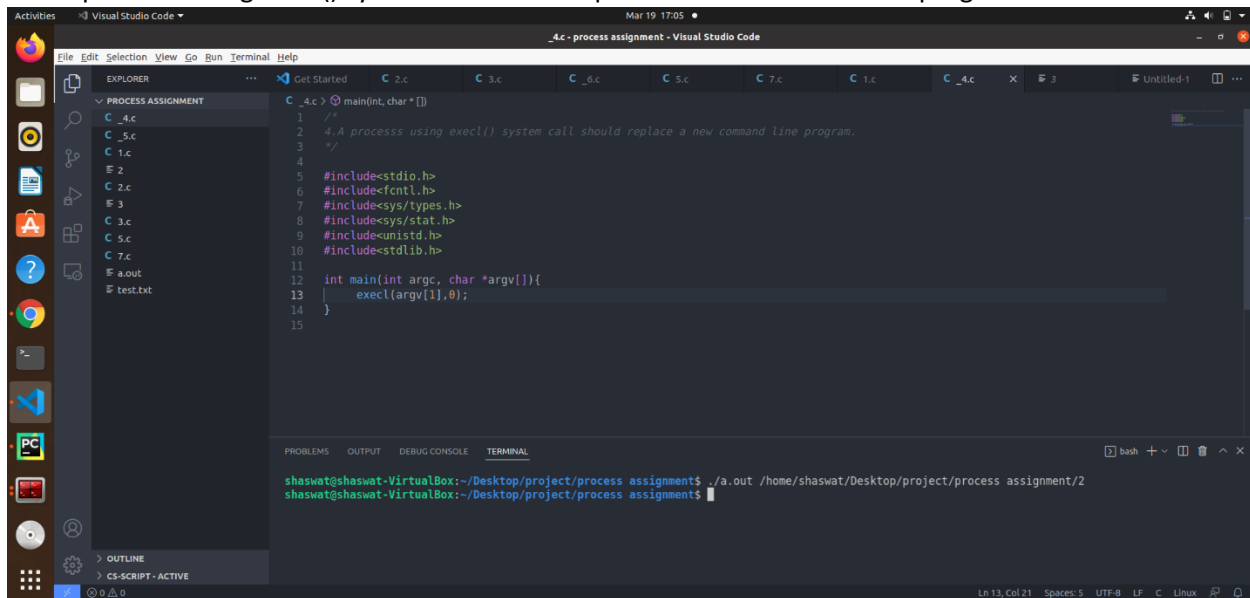
Output:-



```
1 /*
2 4.A processs using execl() system call should replace a new command line program.
3 */
4
5 #include<stdio.h>
6 #include<fcntl.h>
7 #include<sys/types.h>
8 #include<sys/stat.h>
9 #include<unistd.h>
10 #include<stdlib.h>
11
12 int main(int argc, char *argv[]){
13     execl(argv[1],0);
14 }
15
```

```
shaswat@shaswat-VirtualBox:~/Desktop/project/process assignment$ ./a.out /home/shaswat/Desktop/project/process assignment/2
shaswat@shaswat-VirtualBox:~/Desktop/project/process assignment$
```

#### 4. A process using `execl()` system call should replace a new command line program.

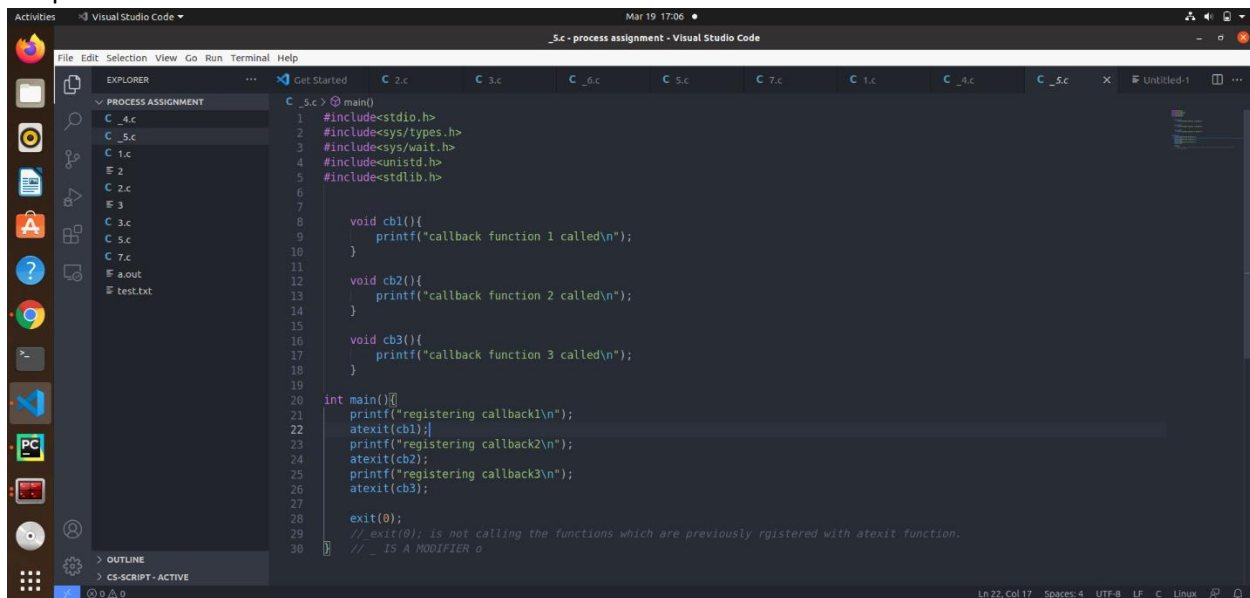


The screenshot shows the Visual Studio Code editor with a C program named `_4.c`. The program uses the `execl()` system call to replace the current process with a new command line program. The code is as follows:

```
1  /*
2  4.A processs using execl() system call should replace a new command line program.
3  */
4
5  #include<stdio.h>
6  #include<fcntl.h>
7  #include<sys/types.h>
8  #include<sys/stat.h>
9  #include<unistd.h>
10 #include<stdlib.h>
11
12 int main(int argc, char *argv[]){
13     execl(argv[1],0);
14 }
15
```

The terminal output shows the command `./a.out /home/shaswat/Desktop/project/process assignment/2` being executed, resulting in the prompt `shaswat@shaswat-VirtualBox:~/Desktop/project/process assignment$`.

#### Output:-



The screenshot shows the Visual Studio Code editor with a C program named `_5.c`. The program uses the `atexit()` system call to register callback functions. The code is as follows:

```
1  #include<stdio.h>
2  #include<sys/types.h>
3  #include<sys/wait.h>
4  #include<unistd.h>
5  #include<stdlib.h>
6
7
8  void cb1(){
9      printf("callback function 1 called\n");
10 }
11
12 void cb2(){
13     printf("callback function 2 called\n");
14 }
15
16 void cb3(){
17     printf("callback function 3 called\n");
18 }
19
20 int main(){
21     printf("registering callback1\n");
22     atexit(cb1);
23     printf("registering callback2\n");
24     atexit(cb2);
25     printf("registering callback3\n");
26     atexit(cb3);
27
28     exit(0);
29     // exit(0); is not calling the functions which are previously registered with atexit function.
30     // _ IS A MODIFIER o

```

The terminal output shows the command `./a.out` being executed, resulting in the prompt `shaswat@shaswat-VirtualBox:~/Desktop/project/process assignment$`.

#### 5. Write a program parent process wait until ,while child process open a file and read file data into empty buffer.

The screenshot shows the Visual Studio Code editor with a C program titled "Sc - process assignment". The code is as follows:

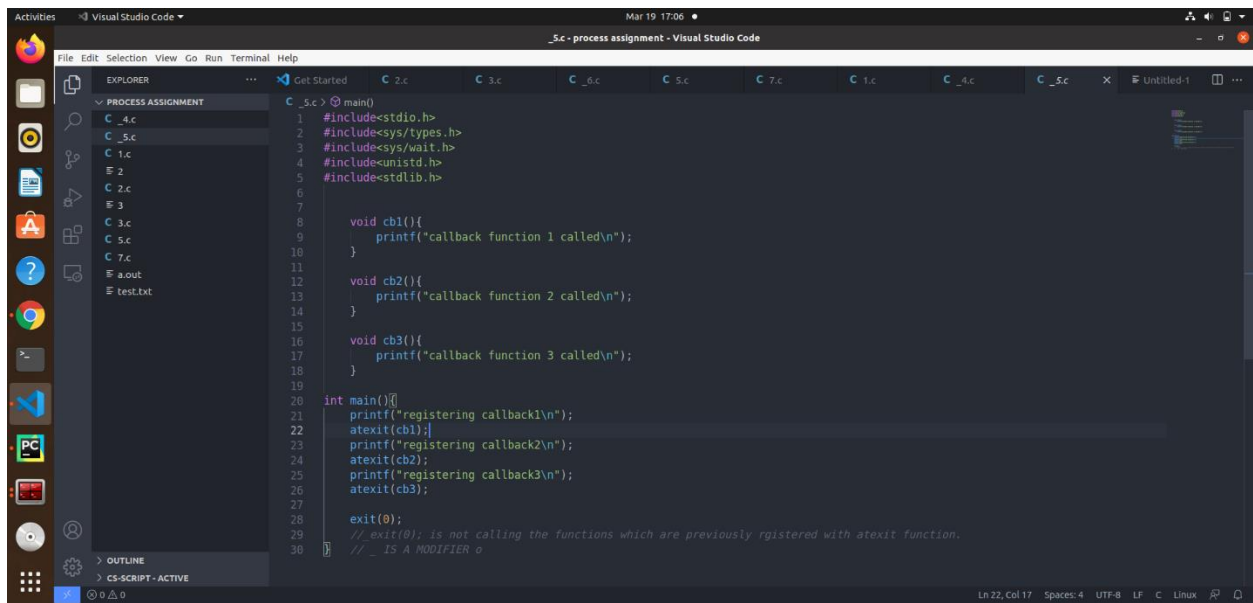
```
1  /*5. Write a program parent process wait untill ,while child process open a file and read file
2  data into empy buffer.*/
3  #include<stdio.h>
4  #include<fcntl.h>
5  #include<sys/types.h>
6  #include<sys/stat.h>
7  #include<unistd.h>
8  #include<stdlib.h>
9  #include<string.h>
10
11 int main(){
12     int pid,fd;
13     int i;
14     pid = fork();
15     char wbuf[30]="a file that will be written.";
16     if (pid == 0){
17         fd=open("test.txt", O_CREAT | O_RDWR, 0777);
18
19         if((i=write(fd,wbuf,30))<0){
20             printf("file written.\n");
21         }
22
23         write(1,wbuf,30);
24         printf("read data is:%s\n",wbuf);
25     }
26     else{
27         wait(0);
28         printf("parent running");
29     }
30 }
```

Output:-

The screenshot shows the Visual Studio Code editor with the same C program. The terminal output is as follows:

```
shaswat@shaswat-VirtualBox:~/Desktop/project/process assignment$ ./a.out
a file that will be written.read data is:a file that will be written.
shaswat@shaswat-VirtualBox:~/Desktop/project/process assignment$
```

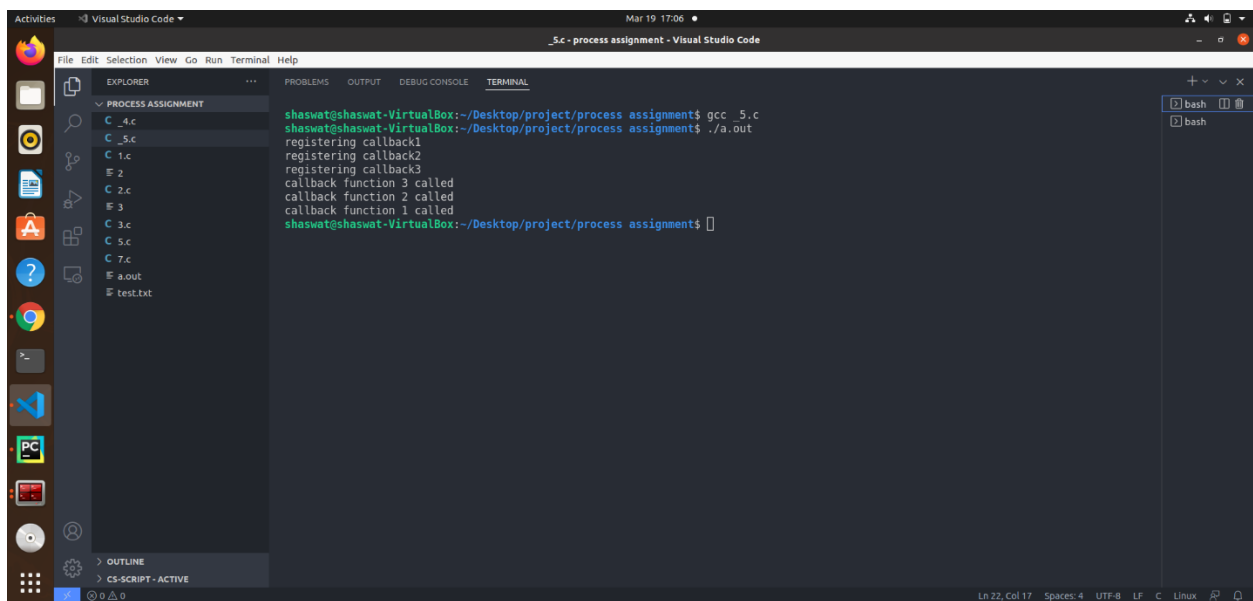
6. Write a program, where functions of the program are called in the reverse order of their function calls from main().



The screenshot shows the Visual Studio Code editor with a C program titled "\_5.c - process assignment". The code defines three callback functions (cb1, cb2, cb3) and a main function that registers them with atexit and then calls exit(0). A comment explains that exit(0) does not call the registered functions.

```
1 #include<stdio.h>
2 #include<sys/types.h>
3 #include<sys/wait.h>
4 #include<unistd.h>
5 #include<stdlib.h>
6
7
8 void cb1(){
9     printf("callback function 1 called\n");
10 }
11
12 void cb2(){
13     printf("callback function 2 called\n");
14 }
15
16 void cb3(){
17     printf("callback function 3 called\n");
18 }
19
20 int main(){
21     printf("registering callback1\n");
22     atexit(cb1);
23     printf("registering callback2\n");
24     atexit(cb2);
25     printf("registering callback3\n");
26     atexit(cb3);
27
28     exit(0);
29     // exit(0); is not calling the functions which are previously registered with atexit function.
30     // _ IS A MODIFIER o
```

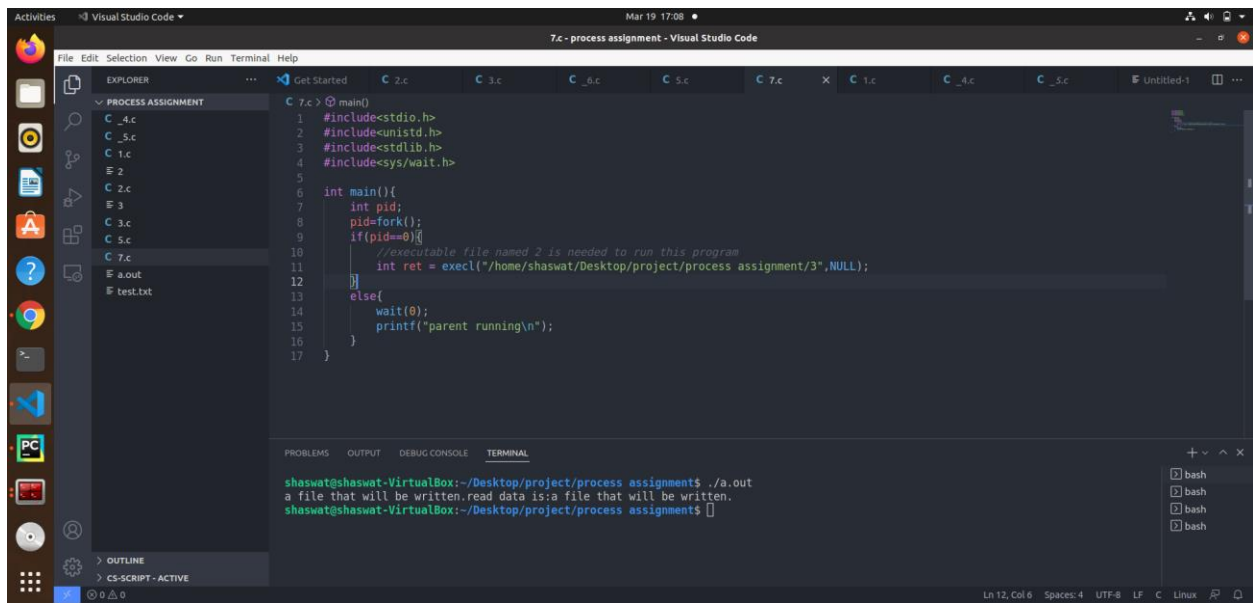
Output:-



The screenshot shows the Visual Studio Code terminal window with the output of the C program. The user has compiled the program with gcc and executed it with ./a.out. The output shows the registration of the three callback functions and then the execution of each callback function in order (3, 2, 1).

```
shaswat@shaswat-VirtualBox:~/Desktop/project/process assignment$ gcc _5.c
shaswat@shaswat-VirtualBox:~/Desktop/project/process assignment$ ./a.out
registering callback1
registering callback2
registering callback3
callback function 3 called
callback function 2 called
callback function 1 called
shaswat@shaswat-VirtualBox:~/Desktop/project/process assignment$
```

7. 6. Write a program child executes(exec()) a new program , while parent waits for child task to get complete.



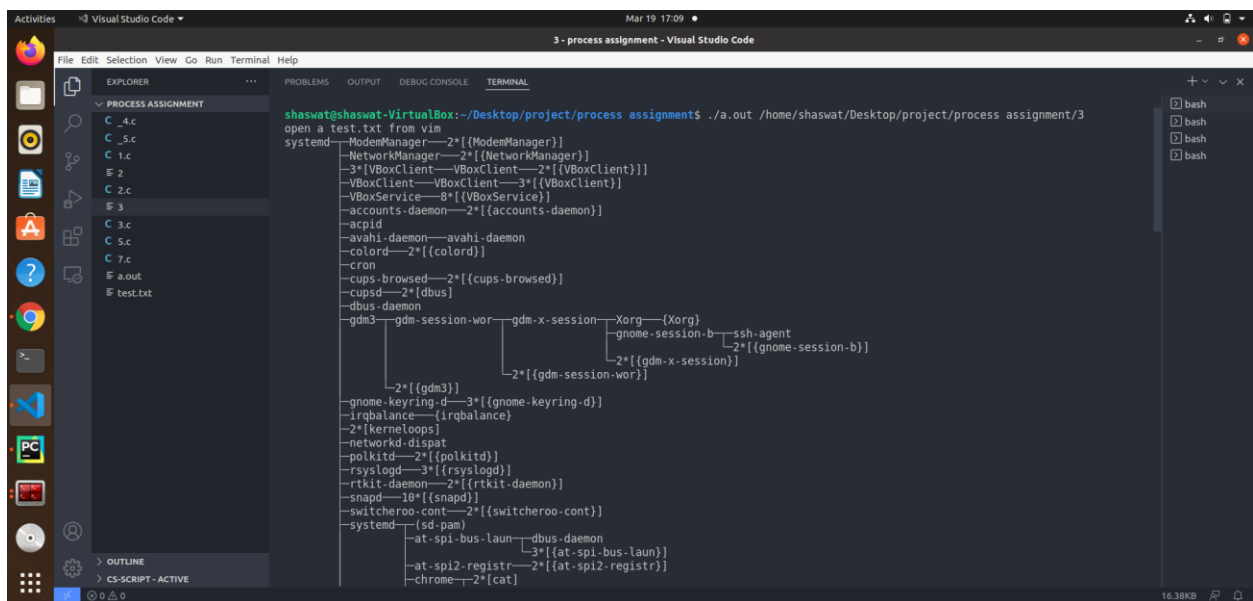
Visual Studio Code interface showing a C program for process assignment 7c. The program is named `7c.c` and is located in the `~/Desktop/project/process assignment/` directory. The code includes `stdio.h`, `unistd.h`, `stdlib.h`, and `sys/wait.h`. It defines a `main` function that forks a child process. The child process executes `exec("/home/shaswat/Desktop/project/process assignment/3", NULL);`. The parent process waits for the child to finish and prints "parent running\n".

```
1 #include<stdio.h>
2 #include<unistd.h>
3 #include<stdlib.h>
4 #include<sys/wait.h>
5
6 int main(){
7     int pid;
8     pid=fork();
9     if(pid==0){
10         //executable file named 2 is needed to run this program
11         int ret = exec("/home/shaswat/Desktop/project/process assignment/3",NULL);
12     }
13     else{
14         wait(0);
15         printf("parent running\n");
16     }
17 }
```

The terminal output shows the execution of the program:

```
shaswat@shaswat-VirtualBox:~/Desktop/project/process assignment$ ./a.out
a file that will be written.read data is:a file that will be written.
shaswat@shaswat-VirtualBox:~/Desktop/project/process assignment$
```

Output:-



Visual Studio Code interface showing the system process tree for process assignment 3. The terminal output displays the results of the `ps aux` command, showing the process hierarchy.

```
shaswat@shaswat-VirtualBox:~/Desktop/project/process assignment$ ./a.out /home/shaswat/Desktop/project/process assignment/3
open a test.txt from vim
systemd--ModemManager--2*[{ModemManager}]
--NetworkManager--2*[{NetworkManager}]
--3*[VBoxClient--VBoxClient--2*[{VBoxClient}]]
--VBoxClient--VBoxClient--3*[{VBoxClient}]
--VBoxService--8*[{VBoxService}]
--accounts-daemon--2*[{accounts-daemon}]
--acpid
--avahi-daemon--avahi-daemon
--colord--2*[{colord}]
--cron
--cups-browsed--2*[{cups-browsed}]
--cupsd--2*[{dbus}]
--dbus-daemon
--gdm3--gdm-session-wor--gdm-x-session--Xorg--(Xorg)
--gnome-session-b--ssh-agent
--2*[{gnome-session-b}]
--2*[{gdm-x-session}]
--2*[{gdm-session-wor}]
--2*[{gdm3}]
--gnome-keyring-d--3*[{gnome-keyring-d}]
--irqbalance--(irqbalance)
--2*[kerneloops]
--networkd-dispat
--polkitd--2*[{polkitd}]
--rsyslogd--3*[{rsyslogd}]
--rtkit-daemon--2*[{rtkit-daemon}]
--snapd--18*[{snapd}]
--switcheroo-cont--2*[{switcheroo-cont}]
--systemd--(sd-pam)
--at-spi-bus-laun--dbus-daemon
--3*[{at-spi-bus-laun}]
--at-spi2-registr--2*[{at-spi2-registr}]
--chrome--2*[cat]
```