

Code/Output:-

2. Write a program for process creation using C

- Orphan Process

Code:-

```
GNU nano 7.2
#include <stdio.h>
#include <unistd.h>

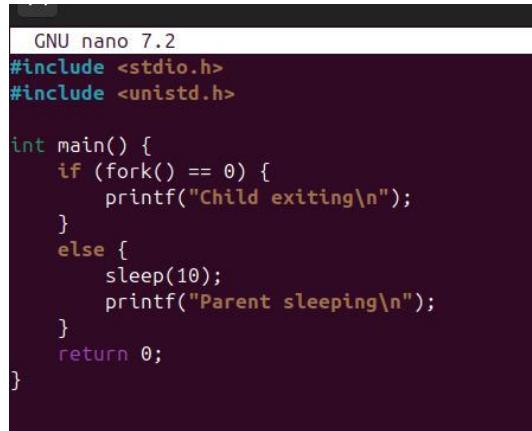
int main() {
    if (fork() == 0) {
        sleep(5);
        printf("Child Process\n");
        printf("PID: %d\n", getpid());
        printf("PPID: %d\n", getppid());
    }
    else {
        printf("Parent exiting\n");
    }
    return 0;
}
```

Output:-

```
Parent PID: 11954
student@M-309-07:~$ nano fork.c
student@M-309-07:~$ nano orphan.c
student@M-309-07:~$ ./orphan
bash: ./orphan: No such file or directory
student@M-309-07:~$ gcc orphan.c -o orphan
student@M-309-07:~$ ./orphan
Parent process exiting...
student@M-309-07:~$ Child process
New Parent PID: 1406
```

- **Zombie Process**

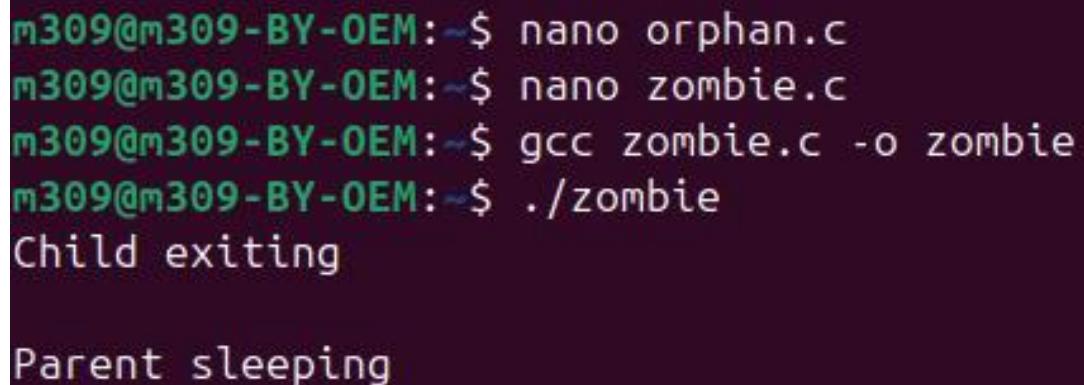
Code:-



```
GNU nano 7.2
#include <stdio.h>
#include <unistd.h>

int main() {
    if (fork() == 0) {
        printf("Child exiting\n");
    }
    else {
        sleep(10);
        printf("Parent sleeping\n");
    }
    return 0;
}
```

Output:-



```
m309@m309-BY-OEM:~$ nano orphan.c
m309@m309-BY-OEM:~$ nano zombie.c
m309@m309-BY-OEM:~$ gcc zombie.c -o zombie
m309@m309-BY-OEM:~$ ./zombie
Child exiting

Parent sleeping
```

4. Create the process using fork 0 system call

- Child Process creation
- Parent Process creation

Code:-

The screenshot shows a terminal window titled "fork.c" with the following code:

```
Feb 2 5:06 PM
student@M-309-07
GNU nano 7.2
#include <stdio.h>
#include <unistd.h>

int main() {
    pid_t pid;

    pid = fork();

    if (pid == 0) {
        printf("Child Process\n");
        printf("Child PID: %d\n", getpid());
        printf("Parent PID: %d\n", getppid());
    }
    else {
        printf("Parent Process\n");
        printf("Parent PID: %d\n", getpid());
    }

    return 0;
}
```

Output:-

The screenshot shows a terminal window with the following session:

```
student@M-309-07:~$ nano fork.c
student@M-309-07:~$ gcc fork.c
student@M-309-07:~$ ./fork
bash: ./fork: No such file or directory
student@M-309-07:~$ gcc fork -o fork
gcc: fatal error: input file ‘fork’ is the same as output file
compilation terminated.
student@M-309-07:~$ gcc fork.c -o fork
student@M-309-07:~$ ./fork
Parent Process
Parent PID: 11334
Child Process
Child PID: 11335
Parent PID: 11334
student@M-309-07:~$
```