H M Mythreya
PES2UG20CS130
OS Lab Week-2

1) Creating a parent and child process.

```
┌──(kali㉿kali)-[~/…/OS_LAB/OS/LAB/Week 2]
└─$ gcc example1.c

┌──(kali㉿kali)-[~/…/OS_LAB/OS/LAB/Week 2]
└─$ gcc example1.o

┌──(kali㉿kali)-[~/…/OS_LAB/OS/LAB/Week 2]
└─$ ./a.out
This is parent. Process Id = 1769, y = -1
This is child. Process Id = 1770, Parent Process Id = 1769, y = 1

┌──(kali㉿kali)-[~/…/OS_LAB/OS/LAB/Week 2]
└─$
```

```
kali@kali: ~/Desktop/OS_LAB/OS/LAB/Week 2
File  Edit  View  Search  Terminal  Help

┌──(kali㉿kali)-[~/…/OS_LAB/OS/LAB/Week 2]
└─$ gcc example1a.c

┌──(kali㉿kali)-[~/…/OS_LAB/OS/LAB/Week 2]
└─$ ./a.out
This is parent. Process Id = 1812, y = -1
This is child. Process Id = 1813, Parent Process Id = 1812, y = 1
This is child. Process Id = 1813, Parent Process Id = 1812, y = 2
This is parent. Process Id = 1812, y = -2
This is child. Process Id = 1813, Parent Process Id = 1812, y = 3
This is parent. Process Id = 1812, y = -3
This is child. Process Id = 1813, Parent Process Id = 1812, y = 4
This is parent. Process Id = 1812, y = -4
This is parent. Process Id = 1812, y = -5
This is child. Process Id = 1813, Parent Process Id = 1812, y = 5
```

2) Creating a parent and child process with different ID's.

```
┌──(kali㉿kali)-[~/…/OS_LAB/OS/LAB/Week 2]
└─$ gcc example2.c

┌──(kali㉿kali)-[~/…/OS_LAB/OS/LAB/Week 2]
└─$ ./a.out
This is child. Process Id = 1845, Parent Process Id = 1844, y = 1
This is parent. Process Id = 1844, y = -1

┌──(kali㉿kali)-[~/…/OS_LAB/OS/LAB/Week 2]
└─$
```

3) Creating parent and child, and grandchild processes.

```
┌──(kali㉿kali)-[~/…/OS_LAB/OS/LAB/Week 2]
└─$ gcc example3.c

┌──(kali㉿kali)-[~/…/OS_LAB/OS/LAB/Week 2]
└─$ ./a.out
This is grandchild. Process Id = 1873, Parent Process Id = 1872, y = 5
This is child. Process Id = 1872, Parent Process Id = 1871, y = 1
This is parent. Process Id = 1871, y = -1

┌──(kali㉿kali)-[~/…/OS_LAB/OS/LAB/Week 2]
└─$
```

```
┌──(kali㉿kali)-[~/…/OS_LAB/OS/LAB/Week 2]
└─$ gcc example4.c

┌──(kali㉿kali)-[~/…/OS_LAB/OS/LAB/Week 2]
└─$ ./a.out
This is grandchild. Process Id = 1924, Parent Process Id = 1923, y = 5
zombie.c  welcome.c  orphan.c  example4.c  example3.c  example2.c  example1.o  example1.c  example1a.c  a.out
This is child. Process Id = 1923, Parent Process Id = 1922, y = 1
This is parent. Process Id = 1922, y = -1
a.out  example1a.c  example1.c  example1.o  example2.c  example3.c  example4.c  orphan.c  welcome.c  zombie.c

┌──(kali㉿kali)-[~/…/OS_LAB/OS/LAB/Week 2]
└─$
```

4) Demonstrating orphan process

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

int main()
{
    int pid = fork();

    if (pid > 0) // parent process
    {
        printf("In parent process process id = %d\n", getpid());
        exit(0);
    }
    else if (pid == 0) //child process
    {
        sleep(30);
        printf("In child process. process id = %d, parent process id = %d\n", getpid(), getppid());
    }

    return 0;
}
```

```
┌──(kali㉿kali)-[~/…/OS_LAB/OS/LAB/Week 2]
└─$ gcc orphan.c

┌──(kali㉿kali)-[~/…/OS_LAB/OS/LAB/Week 2]
└─$ ./a.out
In parent process process id = 2033
```

5) Program to demonstrate zombie process

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

int main()
{
    pid_t pid = fork();

    if (pid > 0) //Parent process
    {
       sleep(50);
       printf("In parent\n");
    }
    else if (pid == 0) //Child process
    {
       printf("In child\n");
       exit(0);
    }

    return 0;
}
```

```
┌──(kali㉿kali)-[~/…/OS_LAB/OS/LAB/Week 2]
└─$ gcc zombie.c

┌──(kali㉿kali)-[~/…/OS_LAB/OS/LAB/Week 2]
└─$ ./a.out
In child
```