NAME : MOITRISH MAITY   DEPT : CSE   ROLL NO: 48   YEAR : 3RD

ASSIGNMENT ON PROCESS CREATION

1. Write a C program for calling a program from another program.

CODE :

```
#include <unistd.h>
#include <stdio.h>
#include <sys/types.h>
#include <stdlib.h>
int main (void)
{
   int pid;
   pid = fork();
   if (pid > 0)
   {
           sleep(1);
           printf("Parent process running..\n");
   }
   else if (pid == 0)
   {

           printf("Child process running..\n");
   }
   else
   {
           printf("Fork error\n");
           exit(1);
   }
   printf("Process terminated...\n");
   exit(0);
}
```

2. Write a C program to print messages when the child process dies.

CODE :

```
#include <unistd.h>
#include <stdio.h>
#include <sys/types.h>
#include <stdlib.h>
int main (void)
{
   int pid;
   pid = fork();
   if (pid > 0)
```

```
    {
            sleep(1);
            printf("Parent process running..\n");
    }
    else if (pid == 0)
    {

            printf("Child process running..\n");
    }
    else
    {
            printf("Fork error\n");
            exit(1);
    }
    if(pid==0)
    {
            printf("Child process terminated..\n");
    }
    exit(0);
}
```

3. **Create 8 processes with a minimum number of fork calling. And also create the zombie process and orphan process.**

CODE :

1ST PART :

```
#include <unistd.h>
#include <stdio.h>
#include <sys/types.h>
#include <stdlib.h>
int main (void)
{
    int pid;
    pid = fork();
    pid = fork();
    pid = fork();
    if (pid > 0)
    {
            sleep(1);
            printf("Process running having Process ID: %d\n",getpid()); }
    else if (pid == 0)
    {
            printf("Process running having Process ID: %d\n",getpid()); }
    else
    {
            printf("Fork error\n");
            exit(1);
    }
```

```
   exit(0);
}
```

ORPHAN PART :

```
#include <unistd.h>
#include <stdio.h>
#include <sys/types.h>
#include <stdlib.h>
int main (void)
{
   int pid;
   pid = fork();
   if (pid > 0)
   {
           printf("Parent process running having Process ID:  %d\n",getpid());
   }
   else if (pid == 0)
   {
           sleep(2);
           printf("Orphan process running having Process ID:  %d\n",getpid());
   }
   else
   {
           printf("Fork error\n");
           exit(1);
   }
   if(pid>0)
   {
           printf("Parent process terminated having Process ID:  %d\n",getpid());
   }
   exit(0);
}
```

ZOMBIE PART :

```
#include <unistd.h>
#include <stdio.h>
#include <sys/types.h>
#include <stdlib.h>
int main (void)
{
   int* p = (int*) malloc(2);
   *p=0;
   printf("Zombie porcess having Process ID: %d created...\n",getpid()); exit(0);
}
```