1.UNIX SYSTEM CALLS

1. Display parent id & process id

```
#include<stdio.h>
int main()
{
printf("\n Parent Process ID %d",getppid());
printf("\n Child Process ID %d\n",getpid());
}
```

2. Process creation using fork

```
#include<stdio.h>
main()
{
printf("Before FORK \n");
fork();
printf("After FORK \n\n");
}
```

3. Process with fork

```
#include<stdio.h>
main()
{
int pid;
pid=fork();
if(pid>0)
{
printf("From Parent \n");
printf("Parent process id %d\n",getpid());
}
else
{
printf("From Child \n");
printf("Child process id %d\n",getpid());
}
}
```

4. Making child as orphan*

```
#include<stdio.h>
main()
{
int pid,pid1;
pid=fork();
if(pid>0)
{
printf("From parent process\n");
printf("Parent process %d \n",getpid());
}
else
```

```
{
sleep(1);
printf("From child process\n");
printf("child process %d \n",getpid());
}}
```

5. Parent waits till completion of child

```
#include<stdio.h>
main()
{
int pid;
pid=fork();
printf("%d\n",pid);
if(pid==0)
{
printf("From child process \n");
}
else
{
wait(0);
printf("From parent process\n");
}
}
```

6. Program to rename a directory.

```
#include<stdio.h>
main()
{
char s[10],d[10];
printf("Enter source Dir Name:\n");
scanf("%s",s);
printf("Enter New Dir Name:\n");
scanf("%s",d);
if(rename(s,d)==-1)
perror("Can't be changed\n");
else
printf("%s is changed to %s\n\n",s,d);
}
```