

## Week 2

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

#define NUM_CHILD 5

int main()
{
    pid_t pid[NUM_CHILD];
    for(int i=0; i<NUM_CHILD; i++)
    {
        pid[i]=fork();
        if(pid[i]==0)
        {
            printf("child[%d], parent[%d]: \n",getpid(),getppid());
            sleep(10);
            printf("Excecuted\n");
            exit(0);
        }
        else if(pid[i]<0)
        {
            printf("Error Generating the Child Process");
        }
        else
        {

```

```
    printf("parent[%d]: \n",getppid());}

    sleep(1);
}

printf("The process is completed");
while(wait(NULL) != -1 || errno != ECHILD);

return 0;
}
```

## Output

| Output                       |
|------------------------------|
| /tmp/R7becDXY7e.o            |
| parent[14481]:               |
| child[14489], parent[14488]: |
| parent[14481]:               |
| child[14491], parent[14488]: |
| parent[14481]:               |
| child[14492], parent[14488]: |
| parent[14481]:               |
| child[14507], parent[14488]: |
| parent[14481]:               |
| child[14508], parent[14488]: |
| Excecuted                    |
| Excecuted                    |
| Excecuted                    |
| Excecuted                    |
| Excecuted                    |
| The process is completed     |