# NETWORK PROGRAMMING LAB

## EXPERIMENT NO 2

Day 1:**23/03/21**

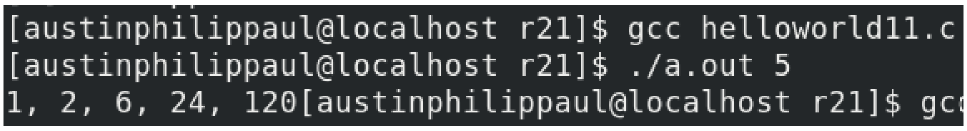
**Aim:**

To familiarize and implement programs related to process.

**1)Write a Unix C Program using fork() system call that generates the factorial and gives a sequence of series like 1, 2, 6, 24, 120... in the child process. The number of sequences is given in command line.**

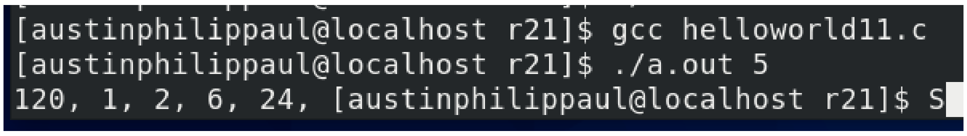
```
#include<stdio.h>

#include<unistd.h>

#include<sys/wait.h>

#include<stdlib.h>


int main(int argc,char *argv[])

{

int fact=1,i=2,n;

n=*argv[1];

n-=48;

while((i-2)<n)

{



if(fork()==0)

{	printf("%d, ",fact);
```

```
        exit(0);

}
else
 {      wait(NULL);
        fact*=i++;
}
}
printf("\b\b");
return 0;
}
```



```
[austinphilippaul@localhost r21]$ gcc helloworld11.c
[austinphilippaul@localhost r21]$ ./a.out 5
1, 2, 6, 24, 120[austinphilippaul@localhost r21]$ gc
```

Without Wait()

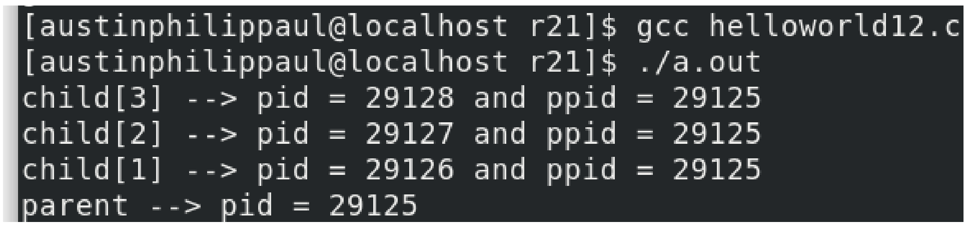


```
[austinphilippaul@localhost r21]$ gcc helloworld11.c
[austinphilippaul@localhost r21]$ ./a.out 5
120, 1, 2, 6, 24, [austinphilippaul@localhost r21]$ S
```

**2)Program to create four processes (1 parent and 3 children) where they terminates in a sequence as follows :**
**(a) Parent process terminates at last**
**(b) First child terminates before parent and after second child.**
**(c) Second child terminates after last and before first child.**
**(d) Third child terminates first.**

```
#include<stdio.h>
```

```
#include<unistd.h>

#include<stdlib.h>



int main()

{

int pid, pid1, pid2;

        pid = fork();

        if (pid == 0)

         {

                sleep(5);

                printf("child[1] --> pid = %d and ppid = %d\n",getpid(), getppid());

         }


else {

pid1 = fork();

if (pid1 == 0) {

sleep(2);

printf("child[2] --> pid = %d and ppid = %d\n",getpid(),getppid());

}

else {

pid2 = fork();

if (pid2 == 0) {


printf("child[3] --> pid = %d and ppid = %d\n",getpid(), getppid());

}
else {

sleep(8);//Cannot use wait as parent will continue when any one child
terminates

printf("parent --> pid = %d\n", getpid());

}

}

}

return 0;

}
```



```
[austinphilippaul@localhost r21]$ gcc helloworld12.c
[austinphilippaul@localhost r21]$ ./a.out
child[3] --> pid = 29128 and ppid = 29125
child[2] --> pid = 29127 and ppid = 29125
child[1] --> pid = 29126 and ppid = 29125
parent --> pid = 29125
```