Laboratory Assignments 4
Subject: Design of Operating Systems
Subject code: CSE 4049

Assignment 4: Familiarization with Process Management in Unix environment.

Objective of this Assignment:

- To trace the different states of a process during its execution.
- To learn the use of different system calls such as (fork(),vfork(),wait(),execl()) for process handling in Unix environment.

1. Write a C program to create a child process using fork() system call. The child process will print the message "Child" with its process identifier and then continue in an indefinite loop.

   The parent process will print the message "Parent" with its process identifier and then continue in an indefinite loop.

   a) Run the program and trace the state of both processes.

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

int main(){
    pid_t pid = fork();
    if(pid < 0){
        perror("Fork failed!");
        exit(EXIT_FAILURE);
    }
    else if(pid == 0){
        printf("Child %d\n", getpid());
        while(1){

        }
    }
        else{
            printf("Parent %d\n", getpid());

            while(1){

            }
        }
    return 0;
}
```

```
student@D001-37:~/2141001059/DOS/Assignment 4$ gedit qn1.c &
[2] 4147
student@D001-37:~/2141001059/DOS/Assignment 4$ gcc qn1.c -o qn1
[2]+  Done                    gedit qn1.c
student@D001-37:~/2141001059/DOS/Assignment 4$ ./qn1
Parent 4159
Child 4160
```

```
student@D001-37:~/2141001059/DOS/Assignment 4$ ps -la
F S   UID     PID    PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
4 S  1001     963     961  1  80   0 - 126587 ep_pol tty2     00:01:11 Xorg
0 S  1001    1147     961  0  80   0 - 49271 do_sys tty2      00:00:00 gnome-se
0 S  1001    3897    3882  1  80   0 - 211100 do_sys pts/0    00:00:11 gedit
0 S  1001    4159    3882  0  80   0 -   624 do_wai pts/0    00:00:00 qn1
1 R  1001    4160    4159 99  80   0 -   624 -      pts/0    00:00:24 qn1
0 R  1001    4200    4012  0  80   0 -  5102 -      pts/1    00:00:00 ps
student@D001-37:~/2141001059/DOS/Assignment 4$
```

b) Terminate the child process. Then trace the state of processes.

```
student@D001-37:~/2141001059/DOS/Assignment 4$ gedit qn1.c &
[2] 4147
student@D001-37:~/2141001059/DOS/Assignment 4$ gcc qn1.c -o qn1
[2]+  Done                    gedit qn1.c
student@D001-37:~/2141001059/DOS/Assignment 4$ ./qn1
Parent 4159
Child 4160
```

```
student@D001-37:~/2141001059/DOS/Assignment 4$ ps -la
F S   UID     PID    PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
4 S  1001     963     961  1  80   0 - 126587 ep_pol tty2     00:01:11 Xorg
0 S  1001    1147     961  0  80   0 - 49271 do_sys tty2      00:00:00 gnome-se
0 S  1001    3897    3882  1  80   0 - 211100 do_sys pts/0    00:00:11 gedit
0 S  1001    4159    3882  0  80   0 -   624 do_wai pts/0    00:00:00 qn1
1 R  1001    4160    4159 99  80   0 -   624 -      pts/0    00:00:24 qn1
0 R  1001    4200    4012  0  80   0 -  5102 -      pts/1    00:00:00 ps
student@D001-37:~/2141001059/DOS/Assignment 4$ kill -9 4160
student@D001-37:~/2141001059/DOS/Assignment 4$ ps -la
F S   UID     PID    PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
4 S  1001     963     961  1  80   0 - 126531 ep_pol tty2     00:01:11 Xorg
0 S  1001    1147     961  0  80   0 - 49271 do_sys tty2      00:00:00 gnome-se
0 S  1001    3897    3882  1  80   0 - 211176 do_sys pts/0    00:00:11 gedit
0 R  1001    4159    3882  7  80   0 -   624 -      pts/0    00:00:03 qn1
0 R  1001    4236    4012  0  80   0 -  5102 -      pts/1    00:00:00 ps
student@D001-37:~/2141001059/DOS/Assignment 4$
```

c) Run the program and trace the state of both processes. Terminate the parent process. Then trace the state of processes.

```
student@D001-37:~/2141001059/DOS/Assignment 4$ gedit qn1.c &
[2] 4147
student@D001-37:~/2141001059/DOS/Assignment 4$ gcc qn1.c -o qn1
[2]+  Done                    gedit qn1.c
student@D001-37:~/2141001059/DOS/Assignment 4$ ./qn1
Parent 4159
Child 4160
Killed
student@D001-37:~/2141001059/DOS/Assignment 4$
```

```
student@D001-37:~/2141001059/DOS/Assignment 4$ ps -la
F S   UID     PID    PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
4 S  1001     963     961  1  80   0 - 126587 ep_pol tty2     00:01:11 Xorg
0 S  1001    1147     961  0  80   0 - 49271 do_sys tty2      00:00:00 gnome-se
0 S  1001    3897    3882  1  80   0 - 211100 do_sys pts/0    00:00:11 gedit
0 S  1001    4159    3882  0  80   0 -   624 do_wai pts/0    00:00:00 qn1
1 R  1001    4160    4159 99  80   0 -   624 -      pts/0    00:00:24 qn1
0 R  1001    4200    4012  0  80   0 -  5102 -      pts/1    00:00:00 ps
student@D001-37:~/2141001059/DOS/Assignment 4$ kill -9 4160
student@D001-37:~/2141001059/DOS/Assignment 4$ ps -la
F S   UID     PID    PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
4 S  1001     963     961  1  80   0 - 126531 ep_pol tty2     00:01:11 Xorg
0 S  1001    1147     961  0  80   0 - 49271 do_sys tty2      00:00:00 gnome-se
0 S  1001    3897    3882  1  80   0 - 211176 do_sys pts/0    00:00:11 gedit
0 R  1001    4159    3882  7  80   0 -   624 -      pts/0    00:00:03 qn1
0 R  1001    4236    4012  0  80   0 -  5102 -      pts/1    00:00:00 ps
student@D001-37:~/2141001059/DOS/Assignment 4$ kill -9 4159
student@D001-37:~/2141001059/DOS/Assignment 4$ ps -la
F S   UID     PID    PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
4 S  1001     963     961  1  80   0 - 126547 ep_pol tty2     00:01:11 Xorg
0 S  1001    1147     961  0  80   0 - 49271 do_sys tty2      00:00:00 gnome-se
0 S  1001    3897    3882  1  80   0 - 211176 do_sys pts/0    00:00:11 gedit
0 R  1001    4269    4012  0  80   0 -  5102 -      pts/1    00:00:00 ps
student@D001-37:~/2141001059/DOS/Assignment 4$
```

d) Modify the program so that the parent process after displaying the message will wait for child process to complete its task. Again run the program and trace the state of both processes.

```c
#include <stdio.h>
#include<unistd.h>
#include<sys/types.h>
#include<sys/wait.h>
int main(){
    if(fork()== 0){
        printf("Child : %d\n",getpid());
        while(1);
    }
    else{
        printf("Parent: %d\n",getpid());
        wait(NULL);
        while(1);
    }
    return(0);
}
```

```
student@D001-37:~/2141001059/DOS/Assignment 4$ gedit qn1.c &
[2] 4147
student@D001-37:~/2141001059/DOS/Assignment 4$ gcc qn1.c -o qn1
[2]+  Done                    gedit qn1.c
student@D001-37:~/2141001059/DOS/Assignment 4$ ./qn1
Parent 4159
Child 4160
Killed
student@D001-37:~/2141001059/DOS/Assignment 4$ gcc qn1.c -o qn1
student@D001-37:~/2141001059/DOS/Assignment 4$ ./qn1
Parent 4307
Child 4308
```

```
student@D001-37:~/2141001059/DOS/Assignment 4$ ps -la
F S   UID     PID    PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
4 S  1001     963     961  1  80   0 - 126587 ep_pol tty2     00:01:11 Xorg
0 S  1001    1147     961  0  80   0 - 49271 do_sys tty2     00:00:00 gnome-se
0 S  1001    3897    3882  1  80   0 - 211100 do_sys pts/0    00:00:11 gedit
0 S  1001    4159    3882  0  80   0 -   624 do_wai pts/0    00:00:00 qn1
1 R  1001    4160    4159 99  80   0 -   624 -      pts/0    00:00:24 qn1
0 R  1001    4200    4012  0  80   0 -  5102 -      pts/1    00:00:00 ps
student@D001-37:~/2141001059/DOS/Assignment 4$ kill -9 4160
student@D001-37:~/2141001059/DOS/Assignment 4$ ps -la
F S   UID     PID    PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
4 S  1001     963     961  1  80   0 - 126531 ep_pol tty2     00:01:11 Xorg
0 S  1001    1147     961  0  80   0 - 49271 do_sys tty2     00:00:00 gnome-se
0 S  1001    3897    3882  1  80   0 - 211176 do_sys pts/0    00:00:11 gedit
0 R  1001    4159    3882  7  80   0 -   624 -      pts/0    00:00:03 qn1
0 R  1001    4236    4012  0  80   0 -  5102 -      pts/1    00:00:00 ps
student@D001-37:~/2141001059/DOS/Assignment 4$ kill -9 4159
student@D001-37:~/2141001059/DOS/Assignment 4$ ps -la
F S   UID     PID    PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
4 S  1001     963     961  1  80   0 - 126547 ep_pol tty2     00:01:11 Xorg
0 S  1001    1147     961  0  80   0 - 49271 do_sys tty2     00:00:00 gnome-se
0 S  1001    3897    3882  1  80   0 - 211176 do_sys pts/0    00:00:11 gedit
0 R  1001    4269    4012  0  80   0 -  5102 -      pts/1    00:00:00 ps
student@D001-37:~/2141001059/DOS/Assignment 4$ ps -la
F S   UID     PID    PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
4 S  1001     963     961  1  80   0 - 126559 ep_pol tty2     00:01:12 Xorg
0 S  1001    1147     961  0  80   0 - 49271 do_sys tty2     00:00:00 gnome-se
0 S  1001    3897    3882  1  80   0 - 211418 do_sys pts/0    00:00:11 gedit
0 S  1001    4307    3882  0  80   0 -   624 do_wai pts/0    00:00:00 qn1
1 R  1001    4308    4307 99  80   0 -   624 -      pts/0    00:00:19 qn1
0 R  1001    4339    4012  0  80   0 -  5102 -      pts/1    00:00:00 ps
student@D001-37:~/2141001059/DOS/Assignment 4$
```

e) Terminate the child process. Then trace the state of processes.

```
student@D001-37:~/2141001059/DOS/Assignment 4$ gedit qn1.c &
[2] 4147
student@D001-37:~/2141001059/DOS/Assignment 4$ gcc qn1.c -o qn1
[2]+  Done                    gedit qn1.c
student@D001-37:~/2141001059/DOS/Assignment 4$ ./qn1
Parent 4159
Child 4160
Killed
student@D001-37:~/2141001059/DOS/Assignment 4$ gcc qn1.c -o qn1
student@D001-37:~/2141001059/DOS/Assignment 4$ ./qn1
Parent 4307
Child 4308
```

```
student@D001-37:~/2141001059/DOS/Assignment 4$ ps -la
F S   UID     PID    PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
4 S  1001     963     961  1  80   0 - 126587 ep_pol tty2     00:01:11 Xorg
0 S  1001    1147     961  0  80   0 - 49271 do_sys tty2     00:00:00 gnome-se
0 S  1001    3897    3882  1  80   0 - 211100 do_sys pts/0    00:00:11 gedit
0 S  1001    4159    3882  0  80   0 -   624 do_wai pts/0    00:00:00 qn1
1 R  1001    4160    4159 99  80   0 -   624 -      pts/0    00:00:24 qn1
0 R  1001    4200    4012  0  80   0 -  5102 -      pts/1    00:00:00 ps
student@D001-37:~/2141001059/DOS/Assignment 4$ kill -9 4160
student@D001-37:~/2141001059/DOS/Assignment 4$ ps -la
F S   UID     PID    PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
4 S  1001     963     961  1  80   0 - 126531 ep_pol tty2     00:01:11 Xorg
0 S  1001    1147     961  0  80   0 - 49271 do_sys tty2     00:00:00 gnome-se
0 S  1001    3897    3882  1  80   0 - 211176 do_sys pts/0    00:00:11 gedit
0 R  1001    4159    3882  7  80   0 -   624 -      pts/0    00:00:03 qn1
0 R  1001    4236    4012  0  80   0 -  5102 -      pts/1    00:00:00 ps
student@D001-37:~/2141001059/DOS/Assignment 4$ kill -9 4159
student@D001-37:~/2141001059/DOS/Assignment 4$ ps -la
F S   UID     PID    PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
4 S  1001     963     961  1  80   0 - 126547 ep_pol tty2     00:01:11 Xorg
0 S  1001    1147     961  0  80   0 - 49271 do_sys tty2     00:00:00 gnome-se
0 S  1001    3897    3882  1  80   0 - 211176 do_sys pts/0    00:00:11 gedit
0 R  1001    4269    4012  0  80   0 -  5102 -      pts/1    00:00:00 ps
student@D001-37:~/2141001059/DOS/Assignment 4$ ps -la
F S   UID     PID    PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
4 S  1001     963     961  1  80   0 - 126559 ep_pol tty2     00:01:12 Xorg
0 S  1001    1147     961  0  80   0 - 49271 do_sys tty2     00:00:00 gnome-se
0 S  1001    3897    3882  1  80   0 - 211418 do_sys pts/0    00:00:11 gedit
0 S  1001    4307    3882  0  80   0 -   624 do_wai pts/0    00:00:00 qn1
1 R  1001    4308    4307 99  80   0 -   624 -      pts/0    00:00:19 qn1
0 R  1001    4339    4012  0  80   0 -  5102 -      pts/1    00:00:00 ps
student@D001-37:~/2141001059/DOS/Assignment 4$ kill -9 4308
student@D001-37:~/2141001059/DOS/Assignment 4$ ps -la
F S   UID     PID    PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
4 S  1001     963     961  1  80   0 - 126559 ep_pol tty2     00:01:13 Xorg
0 S  1001    1147     961  0  80   0 - 49271 do_sys tty2     00:00:00 gnome-se
0 S  1001    3897    3882  1  80   0 - 211418 do_sys pts/0    00:00:11 gedit
0 R  1001    4307    3882  8  80   0 -   624 -      pts/0    00:00:03 qn1
0 R  1001    4366    4012  0  80   0 -  5102 -      pts/1    00:00:00 ps
student@D001-37:~/2141001059/DOS/Assignment 4$
```

2. Trace the output of the following codes:

a)
```
int main( )
{
        if(fork()==0)
printf("1");        else
           printf("2");
printf("3");
return 0; }
```

b)
```
int main( )
      {
         if(vfork()==0)
         {
                printf("1");
_exit(0);

         }
else
printf("2");
printf("3");
        }
```


c)
```
int main( )
{
 pid_t pid;
int i=5;
pid=fork();
i=i+1;
 if(pid= =0)
 {
  printf("Child: %d",i);
 }
else
 {
  wait(NULL);
  printf("Parent: %d",i);
 }
 return 0;
}
```

d)
```
int main( )
{
 pid_t pid;
int i=5;
pid=vfork();
i=i+1;
 if(pid==0)
 {
  printf("Child: %d",i);
  _exit(0);
 }
else
 {
  printf("Parent: %d",i);
 }
 return 0;
}
```

e)
```
int main( )
{
 pid_t pid;
int i=5;
pid=fork();
 if(pid= =0)
 {
i=i+1;
  printf("Child: %d",i);
 }
else
 {
  wait(NULL);
  printf("Parent: %d",i);
 }
 return 0;
}
```

f)
```
int main( )
{
pid_t pid;  int
i=5;
pid=vfork();
 if(pid==0)
 {
i=i+1;
  printf("Child: %d",i);
  _exit(0);
 }
else
 {
  printf("Parent: %d",i);
 }
 return 0;
}
```

g)
```
int main( )
{
int i=5;
 if(fork( )==0)
 {
     printf("Child: %d",i);
 }
else
 {
    printf("Parent: %d",i);
 }
 return 0;
}
```

h)
```
int main( )
{
int i=5;
 if(vfork( )==0)
 {
    printf("Child: %d",i);
    _exit(0);
 }
else
 {
    printf("Parent: %d",i);
 }
  return 0;
}
```

i)
```
int main( )
{
 if(fork( )==0)
 {
     printf("1");
 }
else
 {
      wait(NULL);
printf("2");
printf("3");
   }
   return 0;
 }
```

j)
```
int main( )
{
 if(vfork( )==0)
 {
printf("1");
     _exit(0);
 }      else
{
printf("2");
printf("3");
 }
 return 0;
}
```

k)
```
int main( )
{
pid_t c1;
int n=10;
c1=fork( );
 if(c1==0)
 {
     printf(" Child\n");
n=20;
     printf("n=%d \n",n);
 }
else
 {
      wait(NULL);
printf("Parent\n");
     printf("n=%d \n",n);
 }
  return 0;
}
```

l)
```
int main( )
{
  pid_t c1;
int n=10;
c1=vfork( );
  if(c1==0)
  {
       printf(" Child\n");
n=20;
       printf("n=%d \n",n);
       _exit(0);
  }
else
{
       printf("Parent\n");
       printf("n=%d \n",n);
  }
   return 0;
}
```

m)
```
int main( )
{
int i=5;
fork();
i=i+1;
fork();
fprintf ( stderr,"% d",i);
return 0;
}
```

n)
```
int main( )
{
pid_t pid;
int i=5;
pid=vfork();
if(pid==0)
{
printf("Child: %d",i);
_exit(0);
}
else
{
i=i+1;
printf("Parent: %d",i);
}
return 0;
}
```

o)
```
int main( )
{ int
i=5;
if(fork()==0)
i=i+1;
else i=i-1;
fprintf(stderr,"%d",i);
return 0;
}
```

p)
```
int main( )
{
int i=5;
if(vfork()==0)
{
i=i+1;
_exit(0);
}
else
i=i-1;
fprintf(stderr,"%d",i);
return 0;
}
```

q)
```
int main( )
{
int j,i=5;
for(j=1;j<3;j++)
{
if(fork()==0)
{
i=i+1;
break;
}
else
wait(NULL);
}
fprintf(stderr,"%d",i);
return 0;
}
```

r)
```
int main( )
{
int j,i=5;
for(j=1;j<3;j++)
{
if(fork()!=0)
{
i=i-1;
break;
}
}
fprintf(stderr,"%d",i);
return 0;
}
```

s) int main( )
{
if(fork() == 0)
if(fork())
printf("1\n");
return 0;
}
	t) void fun1(){
	fork();
	fork();
	printf("1\n");
	}
	int
	main() {
	fun1();
	printf("1\n");
	return 0;
	}

A

```
student@D001-37:~/2141001059/DOS/Assignment 4$ gedit qn2a.c &
[2] 4463
student@D001-37:~/2141001059/DOS/Assignment 4$ gcc qn2a.c -o qn2a
[2]+  Done                    gedit qn2a.c
student@D001-37:~/2141001059/DOS/Assignment 4$ ./qn2a
2313student@D001-37:~/2141001059/DOS/Assignment 4$
```

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

int main(){
    if(fork()==0)
        printf("1");
    else
        printf("2");
    printf("3");
    return 0;
}
```

B

```
student@D001-37:~/2141001059/DOS/Assignment 4$ gedit qn2b.c &
[2] 4529
student@D001-37:~/2141001059/DOS/Assignment 4$ gcc qn2b.c -o qn2b
[2]+  Done                    gedit qn2b.c
student@D001-37:~/2141001059/DOS/Assignment 4$ ./qn2b
123student@D001-37:~/2141001059/DOS/Assignment 4$
```

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

int main(){
    if(vfork()==0){
        printf("1");
        _exit(0);
    }
    else
        printf("2");
        printf("3");
}
```

C

```
student@D001-37:~/2141001059/DOS/Assignment 4$ gedit qn2c.c &
[2] 4614
student@D001-37:~/2141001059/DOS/Assignment 4$ gcc qn2c.c  -o qn2c
qn2c.c: In function ‘main’:
qn2c.c:13:3: warning: implicit declaration of function ‘wait’ [-Wimplicit-func
tion-declaration]
   13 |   wait(NULL);
      |   ^~~~
[2]+  Done                    gedit qn2c.c
student@D001-37:~/2141001059/DOS/Assignment 4$ gcc qn2c.c -w qn2c
student@D001-37:~/2141001059/DOS/Assignment 4$ ./qn2c
Child: 6Parent: 6student@D001-37:~/2141001059/DOS/Assignment 4$
```

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

int main(){
    pid_t pid;
    int i=5;
    pid = fork();
    i = i+1;
    if(pid==0){
        printf("Child: %d", i);
    }
    else{
        wait(NULL);
        printf("Parent: %d", i);
    }
    return 0;
}
```

D

```
student@D001-37:~/2141001059/DOS/Assignment 4$ gedit qn2d.c &
[2] 4685
student@D001-37:~/2141001059/DOS/Assignment 4$ gcc qn2d.c -o qn2d
[2]+  Done                    gedit qn2d.c
student@D001-37:~/2141001059/DOS/Assignment 4$ ./qn2d
Child: 6Parent: 7student@D001-37:~/2141001059/DOS/Assignment 4$
```

qn2c.c

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

int main(){
    pid_t pid;
    int i=5;
    pid=vfork();
    i=i+1;
    if(pid==0){
        printf("Child: %d", i);
        _exit(0);
    }
    else{
        printf("Parent: %d", i);
    }
    return 0;
}
```

E

```
student@D001-37:~/2141001059/DOS/Assignment 4$ gedit qn2e.c &
[2] 4747
student@D001-37:~/2141001059/DOS/Assignment 4$ gcc qn2e.c -o qn2e
qn2e.c: In function ‘main’:
qn2e.c:13:3: warning: implicit declaration of function ‘wait’ [-Wimplicit-func
tion-declaration]
   13 |   wait(NULL);
      |   ^~~~
[2]+  Done                    gedit qn2e.c
student@D001-37:~/2141001059/DOS/Assignment 4$ gcc qn2e.c -w qn2e
student@D001-37:~/2141001059/DOS/Assignment 4$ ./qn2e
Child: 6Parent: 5student@D001-37:~/2141001059/DOS/Assignment 4$
```

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

int main(){
    pid_t pid;
    int i=5;
    pid=fork();
    if(pid==0){
        i=i+1;
        printf("Child: %d", i);
    }
    else{
        wait(NULL);
        printf("Parent: %d", i);
    }
    return 0;
}
```

F

```
student@D001-37:~/2141001059/DOS/Assignment 4$ gedit qn2f.c &
[2] 4815
student@D001-37:~/2141001059/DOS/Assignment 4$ gcc qn2f.c -o qn2f
[2]+  Done                    gedit qn2f.c
student@D001-37:~/2141001059/DOS/Assignment 4$ ./qn2f
Child: 6Parent: 6student@D001-37:~/2141001059/DOS/Assignment 4$
```

qn2e.c

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

int main(){
    pid_t pid;
    int i=5;
    pid=vfork();
    if(pid==0){
        i=i+1;
        printf("Child: %d", i);
        _exit(0);
    }
    else{
        printf("Parent: %d", i);
    }
    return 0;
}
```

G

```
student@D001-37:~/2141001059/DOS/Assignment 4$ gedit qn2g.c &
[2] 4893
student@D001-37:~/2141001059/DOS/Assignment 4$ gcc qn2g.c -o qn2g
[2]+  Done                    gedit qn2g.c
student@D001-37:~/2141001059/DOS/Assignment 4$ ./qn2g
Parent: 5Child: 5student@D001-37:~/2141001059/DOS/Assignment 4$
```

qn2e.c

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

int main(){
    int i=5;
    if(fork()==0){
        printf("Child: %d", i);
    }
    else{
        printf("Parent: %d", i);
    }
    return 0;
}
```

H

```
student@D001-37:~/2141001059/DOS/Assignment 4$ gedit qn2h.c &
[2] 3685
student@D001-37:~/2141001059/DOS/Assignment 4$ gcc qn2h.c -o qn2h
[2]+  Done                    gedit qn2h.c
student@D001-37:~/2141001059/DOS/Assignment 4$ ./qn2h
Child: 5 Parent 5student@D001-37:~/2141001059/DOS/Assignment 4$
```

```c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>

int main(){
    int i=5;
    if(vfork()==0){
        printf("Child: %d ", i);
        _exit(0);
    }
    else{
        printf("Parent %d", i);
    }
    return 0;
}
```

I

```
student@D001-37:~/2141001059/DOS/Assignment 4$ gedit qn2i.c &
[2] 3732
student@D001-37:~/2141001059/DOS/Assignment 4$ gcc qn2i.c -o qn2i
qn2i.c: In function ‘main’:
qn2i.c:10:3: warning: implicit declaration of function ‘wait’ [-Wimplicit-func
tion-declaration]
   10 |   wait(NULL);
      |   ^~~~
[2]+  Done                    gedit qn2i.c
student@D001-37:~/2141001059/DOS/Assignment 4$ gcc qn2i.c -w qn2i
student@D001-37:~/2141001059/DOS/Assignment 4$ ./qn2i
123student@D001-37:~/2141001059/DOS/Assignment 4$
```

```c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>

int main(){
    if(fork()==0){
        printf("1");
    }
    else{
        wait(NULL);
        printf("2");
        printf("3");
    }
    return 0;
}
```

J

```
student@D001-37:~/2141001059/DOS/Assignment 4$ gedit qn2j.c &
[2] 3780
student@D001-37:~/2141001059/DOS/Assignment 4$ gcc qn2j.c -o qn2j
[2]+  Done                    gedit qn2j.c
student@D001-37:~/2141001059/DOS/Assignment 4$ ./qn2j
123student@D001-37:~/2141001059/DOS/Assignment 4$
```

```c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>

int main(){
    if(vfork()==0){
        printf("1");
        _exit(0);
    }
    else{
        printf("2");
        printf("3");
    }
    return 0;
}
```

K

```
student@D001-37:~/2141001059/DOS/Assignment 4$ gedit qn2k.c &
[2] 3846
student@D001-37:~/2141001059/DOS/Assignment 4$ gcc qn2k.c -o qn2k
qn2k.c: In function ‘main’:
qn2k.c:15:3: warning: implicit declaration of function ‘wait’ [-Wimplicit-func
tion-declaration]
   15 |   wait(NULL);
      |   ^~~~
[2]+  Done                    gedit qn2k.c
student@D001-37:~/2141001059/DOS/Assignment 4$ gcc qn2k.c -w qn2k
student@D001-37:~/2141001059/DOS/Assignment 4$ ./qn2k
Child
n = 20
Parent
n = 10
student@D001-37:~/2141001059/DOS/Assignment 4$
```

```c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>

int main(){
    pid_t c1;
    int n=10;
    c1 = fork();
    if(c1==0){
        printf("Child\n");
        n = 20;
        printf("n = %d \n", n);
    }
    else{
        wait(NULL);
        printf("Parent\n");
        printf("n = %d \n", n);
    }
    return 0;
}
```

L

```
student@D001-37:~/2141001059/DOS/Assignment 4$ gedit qn2l.c &
[2] 3918
student@D001-37:~/2141001059/DOS/Assignment 4$ gcc qn2l.c -o qn2l
[2]+  Done                    gedit qn2l.c
student@D001-37:~/2141001059/DOS/Assignment 4$ ./qn2l
Child
n = 20
Parent
n = 20
student@D001-37:~/2141001059/DOS/Assignment 4$
```

qn2j.c

```c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>

int main(){
    pid_t c1;
    int n=10;
    c1 = vfork();
    if(c1==0){
        printf("Child\n");
        n = 20;
        printf("n = %d \n", n);
        _exit(0);
    }
    else{
        printf("Parent\n");
        printf("n = %d \n", n);
    }
    return 0;
}
```

M

```
student@D001-37:~/2141001059/DOS/Assignment 4$ gedit qn2m.c &
[1] 4008
student@D001-37:~/2141001059/DOS/Assignment 4$ gcc qn2m.c -o qn2m
student@D001-37:~/2141001059/DOS/Assignment 4$ ./qn2m
 6 6 6 6student@D001-37:~/2141001059/DOS/Assignment 4$
```

```c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>

int main(){
    int i=5;
    fork();
    i = i+1;
    fork();
    fprintf( stderr, " %d", i);
    return 0;
}
```

N

```
student@D001-37:~/2141001059/DOS/Assignment 4$ gedit qn2n.c &
[2] 4064
student@D001-37:~/2141001059/DOS/Assignment 4$ gcc qn2n.c -o qn2n
[2]+  Done                    gedit qn2n.c
student@D001-37:~/2141001059/DOS/Assignment 4$ ./qn2n
Child: 5Parent: 10student@D001-37:~/2141001059/DOS/Assignment 4$
```

qn2m.c

```c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>

int main(){
    pid_t pid;
    int i=5;
    pid=vfork();
    if(pid==0){
        printf("Child: %d", i);
        _exit(0);
    }
    else{
        i=i+i;
        printf("Parent: %d", i);
    }
    return 0;
}
```

O

```
student@D001-37:~/2141001059/DOS/Assignment 4$ gedit qn2o.c &
[2] 4102
student@D001-37:~/2141001059/DOS/Assignment 4$ gcc qn2o.c -o qn2o
[2]+  Done                    gedit qn2o.c
student@D001-37:~/2141001059/DOS/Assignment 4$ ./qn2o
46student@D001-37:~/2141001059/DOS/Assignment 4$
```

qn2m.c

```c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>

int main(){
    int i=5;
    if(fork()==0)
    i=i+1;
    else
    i=i-1;
    fprintf(stderr, "%d",i);
    return 0;
}
```

P

```
student@D001-37:~/2141001059/DOS/Assignment 4$ gedit qn2p.c &
[2] 4219
student@D001-37:~/2141001059/DOS/Assignment 4$ gcc qn2p.c -o qn2p
[2]+  Done                    gedit qn2p.c
student@D001-37:~/2141001059/DOS/Assignment 4$ ./qn2p
5student@D001-37:~/2141001059/DOS/Assignment 4$
```

```c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>

int main(){
    int i=5;
    if(vfork()==0){
        i=i+1;
        _exit(0);
    }
    else
        i=i-1;
    fprintf(stderr, "%d", i);
    return 0;
}
```

Q

```
student@D001-37:~/2141001059/DOS/Assignment 4$ gedit qn2q.c &
[2] 4265
student@D001-37:~/2141001059/DOS/Assignment 4$ gcc qn2q.c -o qn2q
qn2q.c: In function ‘main’:
qn2q.c:13:4: warning: implicit declaration of function ‘wait’ [-Wimplicit-func
tion-declaration]
   13 |    wait(NULL);
      |    ^~~~
[2]+  Done                    gedit qn2q.c
student@D001-37:~/2141001059/DOS/Assignment 4$ gcc qn2q.c -w qn2q
/usr/bin/ld: qn2q: stderr: invalid version 2 (max 0)
/usr/bin/ld: qn2q: error adding symbols: bad value
collect2: error: ld returned 1 exit status
student@D001-37:~/2141001059/DOS/Assignment 4$ ./qn2q
665student@D001-37:~/2141001059/DOS/Assignment 4$
```

qn2p.c

```c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>

int main(){
    int j, i=5;
    for(j=1;j<3;j++){
        if(fork()==0){
            i=i+1;
            break;
        }
        else
            wait(NULL);
    }
    fprintf(stderr, "%d", i);
    return 0;
}
```

R

```
student@D001-37:~/2141001059/DOS/Assignment 4$ gedit qn2r.c &
[2] 4319
student@D001-37:~/2141001059/DOS/Assignment 4$ gcc qn2r.c -o qn2r
[2]+  Done                    gedit qn2r.c
student@D001-37:~/2141001059/DOS/Assignment 4$ ./qn2r
445student@D001-37:~/2141001059/DOS/Assignment 4$
```

qn2p.c

```c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>

int main(){
    int j, i=5;
    for(j=1;j<3;j++){
        if(fork()!=0){
            i=i-1;
            break;
        }
    }
    fprintf(stderr, "%d", i);
    return 0;
}
```

S

```
student@D001-37:~/2141001059/DOS/Assignment 4$ gedit qn2s.c &
[2] 4366
student@D001-37:~/2141001059/DOS/Assignment 4$ gcc qn2s.c -o qn2s
[2]+  Done                    gedit qn2s.c
student@D001-37:~/2141001059/DOS/Assignment 4$ ./qn2s
1
student@D001-37:~/2141001059/DOS/Assignment 4$
```

qn2p.c

```c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>

int main(){
    if(fork()==0)
    if(fork())
    printf("1\n");

    return 0;
}
```

T

```
student@D001-37:~/2141001059/DOS/Assignment 4$ gedit qn2t.c &
[1] 4526
student@D001-37:~/2141001059/DOS/Assignment 4$ gcc qn2t.c -o qn2t
[1]+  Done                    gedit qn2t.c
student@D001-37:~/2141001059/DOS/Assignment 4$ ./qn2t
1
1
1
1
1
1
1
1
student@D001-37:~/2141001059/DOS/Assignment 4$
```

```c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>

void fun1(){
    fork();
    fork();
    printf("1\n");
}

int main(){
    fun1();
    printf("1\n");
    return 0;
}
```

3. Write a C program that will create three child process to perform the following operations respectively:
    - First child will copy the content of file1 to file2
    - Second child will display the content of file2
    - Third child will display the sorted content of file2 in reverse order.
    - Each child process being created will display its id and its parent process id with appropriate message.
    - The parent process will be delayed for 1 second after creation of each child process. It will display appropriate message with its id after completion of all the child processes.

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

void copyFile() {
    printf("\n Copy process with id %d and parent id %d: Copying file1 to file2\n", getpid(), getppid());
    execlp("cp", "cp", "file1.txt", "file2.txt", NULL);
    exit(0);
}

void displayFile() {
    printf("\n Display process with id %d and parent id %d: Displaying content of file2\n", getpid(), getppid());
    execlp("cat", "cat", "file2.txt", NULL);
    exit(0);
}

void reverseSortFile() {
    printf("\n Reverse Display process with id %d and parent id %d: Displaying sorted content of file2 in reverse order\n", getpid(), getppid());
    execlp("sort", "sort", "file2.txt", "-r", NULL);
    exit(0);
}

int main() {
    pid_t child1, child2, child3;

    child1 = fork();
    if (child1 == 0) {
        copyFile();
    }
    sleep(1);

    child2 = fork();
    if (child2 == 0) {
        displayFile();
    }
    sleep(1);
```

```
    child3 = fork();
    if (child3 == 0) {
        reverseSortFile();
    }
    sleep(1);

    waitpid(child1, NULL, 0);
    waitpid(child2, NULL, 0);
    waitpid(child3, NULL, 0);

    printf(" \n Parent process with id %d: All child processes completed\n", getpid());

    return 0;
}
```

```
student@D001-37:~/2141001059/DOS/Assignment 4$ cat >file1.txt
9
5
3
0
7
[1]+  Done                    gedit qn3.c
student@D001-37:~/2141001059/DOS/Assignment 4$ cat file1.txt
9
5
3
0
7
student@D001-37:~/2141001059/DOS/Assignment 4$ gcc qn3.c -o qn3
student@D001-37:~/2141001059/DOS/Assignment 4$ ./qn3

 Copy process with id 6307 and parent id 6306: Copying file1 to file2

 Display process with id 6308 and parent id 6306: Displaying content of file2
9
5
3
0
7

 Reverse Display process with id 6309 and parent id 6306: Displaying sorted co
ntent of file2 in reverse order
9
7
5
3
0

 Parent process with id 6306: All child processes completed
student@D001-37:~/2141001059/DOS/Assignment 4$
```

4. Write a C program that will create a child process to generate a Fibonacci series of specified length and store it in an array. The parent process will wait for the child to complete its task and then display the Fibonacci series and then display the prime Fibonacci number in the series along with its position with appropriate message.

```c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/wait.h>
void f(int n, int *fib) {
    int a = 0, b = 1;
    for (int i = 0; i < n; i++) {
        fib[i] = a; int temp = a + b; a = b; b = temp;
    }
}
int isPrime(int num) {
    if (num < 2) return 0;
    for (int i = 2; i * i <= num; i++) {
        if (num % i == 0) return 0;
    }
    return 1;
}
int main() {
    int n = 10, *fib = (int *)malloc(n * sizeof(int)); pid_t c; int status;
    c = vfork();
    if (c == 0) {
        f(n, fib);
        exit(1);
    } else {
        wait(NULL);
        for (int i = 0; i < n; i++) {
            printf("%d ", fib[i]);
        }
        printf("\n");
        for (int i = 0; i < n; i++) {
            if (isPrime(fib[i])) {
                printf("fib which is prime is %d at position %d\n", fib[i], i + 1);
                break;
            }
       }
    }
    return 0;
}
```

```
student@D001-37:~/Desktop/Aparna2141001059$ gcc lab4q4.c
student@D001-37:~/Desktop/Aparna2141001059$ ./a.out
Enter the length of the Fibonacci series: 2
Fibonacci Series: 0 0
Prime Fibonacci Number: student@D001-37:~/Desktop/Aparna2141001059$
```