

Laboratory Assignment 4
On
Design of Operating System (CSE 4049)

Submitted by

Name : Anurag Jain
Reg. No. : 2141010002
Semester : 5th
Branch : CSE
Section : 15 ('O')
Session : 2023-2024
Admission Batch : 2021



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING FACULTY
OF ENGINEERING & TECHNOLOGY (ITER)**

SIKSHA 'O' ANUSANDHAN DEEMED TO BE UNIVERSITY

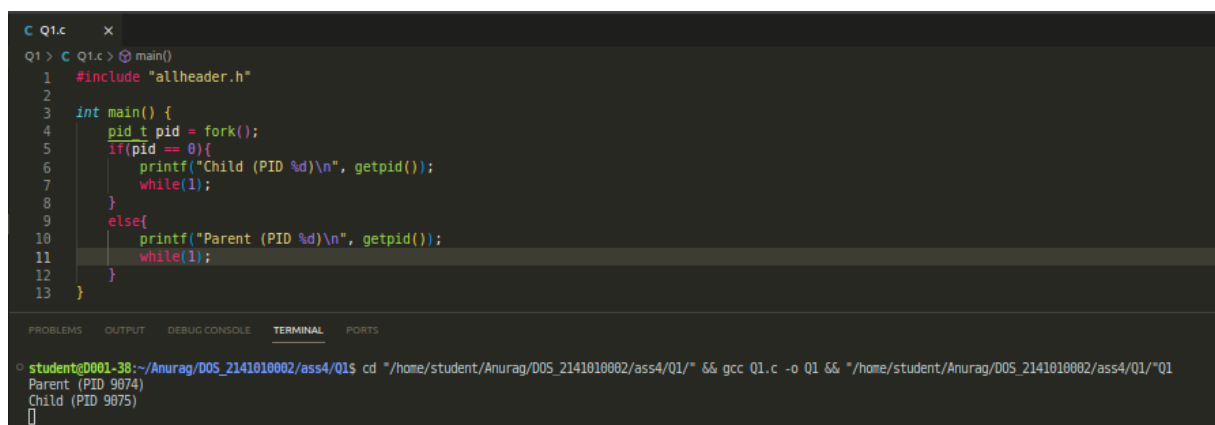
BHUBANESWAR, ODISHA – 751030

**Laboratory Assignment 4 Subject: Design Of
Operating System**
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.
 - b) Terminate the child process. Then trace the state of processes.
 - c) Run the program and trace the state of both processes. Terminate the parent process. Then trace the state of processes.
 - 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.
 - e) Terminate the child process. Then trace the state of processes.



```
C Q1.c x
Q1 > C Q1.c > main()
1  #include "allheader.h"
2
3  int main() {
4      pid_t pid = fork();
5      if(pid == 0){
6          printf("Child (PID %d)\n", getpid());
7          while(1);
8      }
9      else{
10         printf("Parent (PID %d)\n", getpid());
11         while(1);
12     }
13 }
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS

```
student@001-38:~/Anurag/DOS_2141010002/ass4/Q1$ cd ~/home/student/Anurag/DOS_2141010002/ass4/Q1/ && gcc Q1.c -o Q1 && ./Q1
Parent (PID 9074)
Child (PID 9075)
```

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
• student@D001-38:~/Anurag/DOS_2141010002/ass4$ ps -al
F S UID PID PPID C PRI NI ADDR SZ WCHAN TTY TIME CMD
4 S 1001 994 992 2 80 0 - 128376 ep_pol tty2 00:02:32 Xorg
0 S 1001 1153 992 0 80 0 - 49303 do_sys tty2 00:00:00 gnome-session-b
0 R 1001 9074 3663 99 80 0 - 625 - pts/2 00:00:20 Q1
1 R 1001 9075 9074 99 80 0 - 625 - pts/2 00:00:20 Q1
4 R 1001 9140 2198 0 80 0 - 2188 - pts/0 00:00:00 ps
• student@D001-38:~/Anurag/DOS_2141010002/ass4$ kill -9 9075
• student@D001-38:~/Anurag/DOS_2141010002/ass4$ ps -al
F S UID PID PPID C PRI NI ADDR SZ WCHAN TTY TIME CMD
4 S 1001 994 992 2 80 0 - 128376 ep_pol tty2 00:02:32 Xorg
0 S 1001 1153 992 0 80 0 - 49303 do_sys tty2 00:00:00 gnome-session-b
0 R 1001 9074 3663 99 80 0 - 625 - pts/2 00:00:54 Q1
1 Z 1001 9075 9074 97 80 0 - 0 - pts/2 00:00:52 Q1 <defunct>
4 R 1001 9181 2198 0 80 0 - 2188 - pts/0 00:00:00 ps

```

```

• student@D001-38:~/Anurag/DOS_2141010002/ass4$ ps -al
F S UID PID PPID C PRI NI ADDR SZ WCHAN TTY TIME CMD
4 S 1001 994 992 2 80 0 - 128376 ep_pol tty2 00:02:34 Xorg
0 S 1001 1153 992 0 80 0 - 49303 do_sys tty2 00:00:00 gnome-session-b
0 R 1001 9326 3663 99 80 0 - 625 - pts/2 00:00:11 Q1
1 R 1001 9327 9326 99 80 0 - 625 - pts/2 00:00:11 Q1
4 R 1001 9341 2198 0 80 0 - 2188 - pts/0 00:00:00 ps
• student@D001-38:~/Anurag/DOS_2141010002/ass4$ kill -9 9326
• student@D001-38:~/Anurag/DOS_2141010002/ass4$ ps -al
F S UID PID PPID C PRI NI ADDR SZ WCHAN TTY TIME CMD
4 S 1001 994 992 2 80 0 - 128376 ep_pol tty2 00:02:34 Xorg
0 S 1001 1153 992 0 80 0 - 49303 do_sys tty2 00:00:00 gnome-session-b
1 R 1001 9327 950 99 80 0 - 625 - pts/2 00:00:28 Q1
4 R 1001 9431 2198 0 80 0 - 2188 - pts/0 00:00:00 ps

```

```

• student@D001-38:~/Anurag/DOS_2141010002/ass4/Q1$ cd ~/home/student/An
Parent (PID 9326)
Child (PID 9327)
Killed
• student@D001-38:~/Anurag/DOS_2141010002/ass4/Q1$ █

```

```

• student@D001-38:~/Anurag/DOS_2141010002/ass4/Q1$ cd ~/home/student/An
Parent (PID 9729)
waiting...
Child (PID 9730)
█

```

```

• student@D001-38:~/Anurag/DOS_2141010002/ass4$ ps -al
F S UID PID PPID C PRI NI ADDR SZ WCHAN TTY TIME CMD
4 S 1001 994 992 2 80 0 - 128376 ep_pol tty2 00:02:38 Xorg
0 S 1001 1153 992 0 80 0 - 49303 do_sys tty2 00:00:00 gnome-session-b
1 R 1001 9327 950 99 80 0 - 625 - pts/2 00:03:34 Q1
0 S 1001 9729 3663 0 80 0 - 625 do_wai pts/2 00:00:00 Q1
1 R 1001 9730 9729 99 80 0 - 625 - pts/2 00:00:08 Q1
4 R 1001 9754 2198 0 80 0 - 2188 - pts/0 00:00:00 ps

```

```

• student@D001-38:~/Anurag/DOS_2141010002/ass4/Q1$ cd ~/home/studen
Parent (PID 10143)
waiting...
Child (PID 10144)

```

2. Trace the output of the following codes:

```
C Q2a.c X
C Q2a.c > main()
1 #include "allheader.h"
2
3 int main()
4 {
5     if(fork()==0)
6     {
7         printf("1");
8     }
9     else
10    {
11        printf("2");
12    }
13    printf("3");
14    return 0;
15 }
```

student@001-38:~/Anurag/DOS_2141010002/ass4/Q2a\$./Q2a

2313student@001-38:~/Anurag/DOS_2141010002/ass4\$

```
C Q4.c C Q2b.c X
Q2 > C Q2b.c > ...
1 #include "allheader.h"
2
3 int main()
4 {
5     if (vfork() == 0)
6     {
7         printf("1");
8         _exit(0);
9     }
10    else
11    {
12        printf("2");
13        printf("3");
14    }
15 }
```

student@001-38:~/Anurag/DOS_2141010002/ass4/Q2b\$ cd ~/home/student/Anurag/DOS_2141010002/ass4/Q2b\$./Q2b

123student@001-38:~/Anurag/DOS_2141010002/ass4/Q2b\$

```
C Q4.c C Q2c.c X
Q2 > C Q2c.c > ...
1 #include "allheader.h"
2
3 int main()
4 {
5     pid_t pid;
6     int i = 5;
7     pid = fork();
8     i = i + 1;
9     if (pid == 0)
10    {
11        printf("Child: %d", i);
12    }
13    else
14    {
15        wait(NULL);
16        printf("Parent: %d", i);
17    }
18    return 0;
19 }
```

student@001-38:~/Anurag/DOS_2141010002/ass4/Q2c\$ cd ~/home/student/Anurag/DOS_2141010002/ass4/Q2c\$./Q2c

Child: 6Parent: 6student@001-38:~/Anurag/DOS_2141010002/ass4/Q2c\$

```
C Q4.c C Q2d.c X
Q2 > C Q2d.c > ...
1 #include "allheader.h"
2
3 int main()
4 {
5     pid_t pid;
6     int i = 5;
7     pid = vfork();
8     i = i + 1;
9     if (pid == 0)
10    {
11        printf("Child: %d", i);
12        _exit(0);
13    }
14    else
15    {
16        printf("Parent: %d", i);
17    }
18    return 0;
19 }
```

student@001-38:~/Anurag/DOS_2141010002/ass4/Q2d\$ cd ~/home/student/Anurag/DOS_2141010002/ass4/Q2d\$./Q2d

Child: 6Parent: 7student@001-38:~/Anurag/DOS_2141010002/ass4/Q2d\$

```
C Q4.c C Q2e.c X
Q2 > C Q2e.c > main()
1 #include "allheader.h"
2
3 int main()
4 {
5     pid_t pid;
6     int i = 5;
7     pid = fork();
8     if (pid == 0)
9     {
10        i = i + 1;
11        printf("Child: %d", i);
12    }
13    else
14    {
15        wait(NULL);
16        printf("Parent: %d", i);
17    }
18    return 0;
19 }
```

student@001-38:~/Anurag/DOS_2141010002/ass4/Q2e\$ cd ~/home/student/Anurag/DOS_2141010002/ass4/Q2e\$./Q2e

Child: 6Parent: 5student@001-38:~/Anurag/DOS_2141010002/ass4/Q2e\$

```
C Q4.c C Q2f.c X
Q2 > C Q2f.c > ...
1 #include "allheader.h"
2
3 int main()
4 {
5     pid_t pid;
6     int i = 5;
7     pid = vfork();
8     if (pid == 0)
9     {
10        i = i + 1;
11        printf("Child: %d", i);
12        _exit(0);
13    }
14    else
15    {
16        printf("Parent: %d", i);
17    }
18    return 0;
19 }
```

student@001-38:~/Anurag/DOS_2141010002/ass4/Q2f\$ cd ~/home/student/Anurag/DOS_2141010002/ass4/Q2f\$./Q2f

Child: 6Parent: 6student@001-38:~/Anurag/DOS_2141010002/ass4/Q2f\$

```
C Q4.c C Q2g.c X
Q2 > C Q2g.c > ...
1 #include "allheader.h"
2
3 int main()
4 {
5     int i = 5;
6     if (fork() == 0)
7     {
8         printf("Child: %d", i);
9     }
10    else
11    {
12        printf("Parent: %d", i);
13    }
14    return 0;
15 }
```

student@001-38:~/Anurag/DOS_2141010002/ass4/Q2g\$ cd ~/home/student/Anurag/DOS_2141010002/ass4/Q2g\$./Q2g

Parent: 5Child: 5student@001-38:~/Anurag/DOS_2141010002/ass4/Q2g\$

```
C Q4.c C Q2h.c X
Q2 > C Q2h.c > main()
1 #include "allheader.h"
2
3 int main()
4 {
5     int i = 5;
6     if (vfork() == 0)
7     {
8         printf("Child: %d", i);
9         _exit(0);
10    }
11    else
12    {
13        printf("Parent: %d", i);
14    }
15    return 0;
16 }
```

student@001-38:~/Anurag/DOS_2141010002/ass4/Q2h\$ cd ~/home/student/Anurag/DOS_2141010002/ass4/Q2h\$./Q2h

Child: 5Parent: 5student@001-38:~/Anurag/DOS_2141010002/ass4/Q2h\$

```
C Q4.c C Q2i.c X
Q2 > C Q2i.c > ...
1 #include "allheader.h"
2
3 int main()
4 {
5     if (fork() == 0)
6     {
7         printf("1");
8     }
9     else
10    {
11        wait(NULL);
12        printf("2");
13        printf("3");
14    }
15    return 0;
16 }
```

student@001-38:~/Anurag/DOS_2141010002/ass4/Q2i\$ cd ~/home/student/Anurag/DOS_2141010002/ass4/Q2i\$./Q2i

123student@001-38:~/Anurag/DOS_2141010002/ass4/Q2i\$

```
C Q4.c C Q2j.c X
Q2 > C Q2j.c > ...
1 #include "allheader.h"
2
3 int main()
4 {
5     if (vfork() == 0)
6     {
7         printf("1");
8         _exit(0);
9     }
10    else
11    {
12        printf("2");
13        printf("3");
14    }
15    return 0;
16 }
```

student@001-38:~/Anurag/DOS_2141010002/ass4/Q2j\$ cd ~/home/student/Anurag/DOS_2141010002/ass4/Q2j\$./Q2j

123student@001-38:~/Anurag/DOS_2141010002/ass4/Q2j\$

```
C Q4.c C Q2k.c X
Q2 > C Q2k.c > ...
1 #include "allheader.h"
2
3 int main()
4 {
5     pid_t cl;
6     int n = 10;
7     cl = fork();
8     if (cl == 0)
9     {
10        printf(" Child\n");
11        n = 20;
12        printf("n=%d \n", n);
13    }
14    else
15    {
16        wait(NULL);
17        printf("Parent\n");
18    }
19 }
```

student@001-38:~/Anurag/DOS_2141010002/ass4/Q2k\$ cd ~/home/student/Anurag/DOS_2141010002/ass4/Q2k\$./Q2k

Child
n=20
Parent
n=10

```
C Q4.c C Q2l.c X
Q2 > C Q2l.c > ...
1 #include "allheader.h"
2
3 int main()
4 {
5     pid_t cl;
6     int n = 10;
7     cl = vfork();
8     if (cl == 0)
9     {
10        printf(" Child\n");
11        n = 20;
12        printf("n=%d \n", n);
13        _exit(0);
14    }
15    else
16    {
17        printf("Parent\n");
18    }
19 }
```

student@001-38:~/Anurag/DOS_2141010002/ass4/Q2l\$ cd ~/home/student/Anurag/DOS_2141010002/ass4/Q2l\$./Q2l

Child
n=20
Parent
n=20

```
C Q4.c C Q2m.c x
Q2 > C Q2m.c > main()
1 #include "allheader.h"
2
3 int main()
4 {
5     int i = 5;
6     fork();
7     i = i + 1;
8     fork();
9     fprintf(stderr, "%d", i);
10    return 0;
11 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

student@D001-38:~/Anurag/DOS_2141010002/ass4/Q2\$./Q2m
6 6 6 6 student@D001-38:~/Anurag/DOS_2141010002/ass4/Q2\$

```
C Q4.c C Q2m.c C Q2nc x
Q2 > C Q2nc > ...
1 #include "allheader.h"
2
3 int main()
4 {
5     pid_t pid;
6     int i = 5;
7     pid = vfork();
8     if (pid == 0)
9     {
10        printf("Child: %d", i);
11        _exit(0);
12    }
13    else
14    {
15        i = i + 1;
16        printf("Parent: %d", i);
17    }
18    return 0;
19 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

student@D001-38:~/Anurag/DOS_2141010002/ass4/Q2\$./Q2n
Child: 5 Parent: 6 student@D001-38:~/Anurag/DOS_2141010002/ass4/Q2\$

```
C Q4.c C Q2m.c C Q2o.c x
Q2 > C Q2o.c > ...
1 #include "allheader.h"
2
3 int main()
4 {
5     int i = 5;
6     if (fork() == 0)
7     {
8         i = i + 1;
9     }
10    else
11    {
12        i = i - 1;
13    }
14    fprintf(stderr, "%d", i);
15    return 0;
16 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

student@D001-38:~/Anurag/DOS_2141010002/ass4/Q2\$./Q2o
46 student@D001-38:~/Anurag/DOS_2141010002/ass4/Q2\$

```
C Q4.c C Q2m.c C Q2p.c x
Q2 > C Q2p.c > ...
1 #include "allheader.h"
2
3 int main()
4 {
5     int i = 5;
6     if (vfork() == 0)
7     {
8         i = i + 1;
9         _exit(0);
10    }
11    else
12    {
13        i = i - 1;
14        fprintf(stderr, "%d", i);
15    }
16 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

student@D001-38:~/Anurag/DOS_2141010002/ass4/Q2\$./Q2p
5 student@D001-38:~/Anurag/DOS_2141010002/ass4/Q2\$

```
C Q4.c C Q2m.c C Q2qc x
Q2 > C Q2qc > ...
1 #include "allheader.h"
2
3 int main()
4 {
5     int j, i = 5;
6     for (j = 1; j < 3; j++)
7     {
8         if (fork() == 0)
9         {
10            i = i + 1;
11            break;
12        }
13        else
14        {
15            wait(NULL);
16        }
17        fprintf(stderr, "%d", i);
18    }
19 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

student@D001-38:~/Anurag/DOS_2141010002/ass4/Q2\$./Q2q
665 student@D001-38:~/Anurag/DOS_2141010002/ass4/Q2\$

```
C Q4.c C Q2m.c C Q2rc x
Q2 > C Q2rc > ...
1 #include "allheader.h"
2
3 int main()
4 {
5     int j, i = 5;
6     for (j = 1; j < 3; j++)
7     {
8         if (fork() != 0)
9         {
10            i = i - 1;
11            break;
12        }
13    }
14    fprintf(stderr, "%d", i);
15    return 0;
16 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

student@D001-38:~/Anurag/DOS_2141010002/ass4/Q2\$./Q2r
445 student@D001-38:~/Anurag/DOS_2141010002/ass4/Q2\$

```
C Q4.c C Q2m.c C Q2s.c x
Q2 > C Q2s.c > ...
1 #include "allheader.h"
2
3 int main()
4 {
5     if (fork() == 0)
6     {
7         if (fork())
8         {
9             printf("1\n");
10        }
11    }
12    return 0;
13 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

student@D001-38:~/Anurag/DOS_2141010002/ass4/Q2\$./Q2s
1

```
C Q4.c C Q2m.c C Q2tc x
Q2 > C Q2tc > ...
1 #include "allheader.h"
2
3 void fun1()
4 {
5     fork();
6     fork();
7     printf("1\n");
8 }
9
10 int main()
11 {
12     fun1();
13     printf("1\n");
14     return 0;
15 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

student@D001-38:~/Anurag/DOS_2141010002/ass4/Q2\$./Q2t
1
1
1
1
1
1
1
1
1
1

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 Q3.c x file2.txt file1.txt
C Q3.c> @ main()
1 #include "allheader.h"
2
3 int main()
4 {
5     pid_t pid1, pid2, pid3;
6
7     pid1 = fork();
8     if (pid1 == 0)
9     {
10         printf("pid of child1: %ld\nppid of child1: %ld\n", (float)getpid(), (float)getppid());
11         execlp("cp", "cp", "file1.txt", "file2.txt", NULL);
12         exit(0);
13     }
14     sleep(1);
15
16     pid2 = fork();
17     if (pid2 == 0)
18     {
19         printf("pid of child2: %ld\nppid of child2: %ld\n", (float)getpid(), (float)getppid());
20         printf("Content of file2 before sort: \n");
21         execlp("cat", "cat", "file2.txt", NULL);
22         exit(0);
23     }
24     sleep(1);
25
26     pid3 = fork();
27     if (pid3 == 0)
28     {
29         printf("pid of child3: %ld\nppid of child3: %ld\n", (float)getpid(), (float)getppid());
30         printf("Content of file2 after reverse sort: \n");
31         execlp("sort", "sort", "-r", "file2.txt", NULL);
32         exit(0);
33     }
34     sleep(1);
35
36     waitpid(pid1, NULL, 0);
37     waitpid(pid2, NULL, 0);
38     waitpid(pid3, NULL, 0);
39
40     printf("Parent Process pid : %ld\n", (float)getpid());
41 }
```

```
student0001-38:~/Anurag/D05_2141010002/ass4$ cd ~/home/student/Anurag/D05_2141010002/ass4/ && gcc Q3.c -o Q3 && ./Q3
pid of child1: 12822.000000
ppid of child1: 12821.000000
pid of child2: 12831.000000
ppid of child2: 12821.000000
Content of file2 before sort:
3
5
2
4
1
pid of child3: 12854.000000
ppid of child3: 12821.000000
Content of file2 after reverse sort:
5
4
3
2
1
Parent Process pid : 12821.000000
```

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 Q4.c x
Q4 > C Q4.c > main()
1  #include "allheader.h"
2
3  void fib(int *a, int n){
4      for(int i = 2; i <= n; i++){
5          a[i] = a[i - 1] + a[i - 2];
6      }
7  }
8
9  int isPrime(int n){
10
11     for(int i = 2; i < n / 2; i++){
12         if(n % i == 0) return 0;
13     }
14     return 1;
15 }
16
17 int main()
18 {
19     int n = 15;
20
21     int *arr = (int*)malloc((n + 1) * sizeof(int));
22     arr[0] = 0; arr[1] = 1;
23
24     pid_t c1 = vfork();
25
26     if(c1 == 0){
27         fib(arr, n);
28         _exit(0);
29     }else{
30         wait(NULL);
31
32         for(int i = 0; i <= n; i++){
33             printf("%d ", arr[i]);
34         }
35
36         printf("\n");
37
38         for(int i = 2; i <= n; i++){
39             if(isPrime(arr[i])){
40                 printf("%d %d\n", i, arr[i]);
41             }
42         }
43     }
44     return 0;
45 }
```

```
● cd "/home/student/Anurag/DOS_2141010002/ass4/Q4/" && gcc Q4.c -o Q4 && "/home/student/Anurag/DOS_214s
/Q4/"Q4
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610
2 1
3 2
4 3
5 5
7 13
11 89
13 233
```