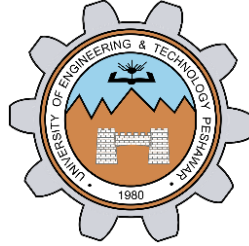


MULTIPLE FILE COPYING

LAB # 07



Fall 2023

CSE-302L Systems Programming Lab

Submitted by: **Ali Asghar**

Registration No. : **21pwcse2059**

Class Section: **C**

“On my honor, as student of University of Engineering and Technology, I have neither given nor received unauthorized assistance on this academic work.”

Submitted to:

Engr. Abdullah Hamid

DATED: 28 \01 \2024

Department of Computer Systems Engineering
University of Engineering and Technology, Peshawar

OBJECTIVES:

- To learn about UNIX input and output.
- To learn about File Handling.

Tasks:**Task 01:**

Write a program that copies two files sequentially in a single process.

Code :

```
C T1.c > main(int, char * [])
1  #include <unistd.h>
2  #include <stdio.h>
3  #include <stdlib.h>
4  #include <fcntl.h>
5  #include <string.h>
6  int main(int argc, char *argv[])
7  {
8      if (argc < 5)
9      {
10         printf("Error: Invalid number of arguments\n");
11         return -1;
12     }
13     int op1 = open(argv[1], O_RDONLY);
14     if (op1 == -1)
15     {
16         perror("Open 1 Failed");
17         return -1;
18     }
19     int op2 = open(argv[2], O_RDONLY);
20     if (op2 == -1)
21     {
22         perror("Open 2 Failed");
23         return -1;
24     }
25     int opx1 = open(argv[3], O_RDWR | O_CREAT, S_IRWXU | S_IRWXG | S_IRWXO);
26     if (opx1 == -1)
27     {
28         perror("New file 1 Open Failed");
29         return -1;
30     }
```

```
30     }
31     int opx2 = open(argv[4], O_RDWR | O_CREAT, S_IRWXU | S_IRWXG | S_IRWXO);
32     if (opx2 == -1)
33     {
34         perror("New file 2 Open Failed");
35         return -1;
36     }
37     char buff[10];
38     int size = 10;
39     while (1)
40     {
41         int rd = read(op1, &buff, size);
42         if (rd == -1)
43         {   perror("Read failed");   return -1;   }
44         if (rd > 0)
45         {
46             int wr = write(opx1, buff, strlen(buff));
47             if (wr == -1)
48             {   perror("Write Failed");   }
49         }
50         if (rd == 0)
51         {   break;   }
52     }
53     while (1){
54         int rd = read(op2, &buff, size);
55         if (rd == -1)
56         {   perror("Read failed");   return -1;   }
57         if (rd > 0)
58         {
59             int wr = write(opx2, buff, strlen(buff));
60             if (wr == -1)
61             {perror("Write Failed");}
62         }
63         if (rd == 0)
64         {   break;   }
65     }
66     unlink(argv[1]);
67     unlink(argv[2]);
68 }
```

Ouput :

SP LAB REPROT

```
DEBUG CONSOLE  TERMINAL  JUPYTER  COMMENTS
~/Documents/SP_Lab/Lab 07$ gcc T1.c -o t1
~/Documents/SP_Lab/Lab 07$ ./t1 f1 f2 ./f1x ./f2x
~/Documents/SP_Lab/Lab 07$ ./t1 f1x f2x ../f1x ../f2x
~/Documents/SP_Lab/Lab 07$ ./t1 file1 file2 ../file1 ../file2
~/Documents/SP_Lab/Lab 07$
```

Task 02:

Write a program that monitors two files by forking a child .

Code :

```
C T2.c  x
C T2.c > main()
1  #include <stdio.h>
2  #include <unistd.h>
3  #include <wait.h>
4  #include <sys/wait.h>
5
6  int main()
7  {
8      int pid;
9      int status;
10     pid = fork();
11     if (pid == 0)
12     {
13         fprintf(stderr, "Child Process\n");
14         sleep(5);
15         fprintf(stderr, "Child Process\n");
16         return 3; // exit status is 3 from child process
17     }
18     else
19     {
20         sleep(3);
21         wait(&status);
22         printf("In parent process: exit status from child is %d, \n", status);
23     }
24     return 0;
25 }
```

Output :

SP LAB REPROT

```
DEBUG CONSOLE  TERMINAL  JUPYTER  COMMENTS

~/Documents/SP_Lab/Lab 07$ gcc T2.c -o t2
~/Documents/SP_Lab/Lab 07$ ./t2

s: exit status from child is 768,
~/Documents/SP_Lab/Lab 07$
```

Task 03:

Write a program that monitors two files using 'select' .

Code :

```
C T3.c > main(int, char *[])
1  #include <stdio.h>
2  #include <unistd.h>
3  #include <fcntl.h>
4  #include <sys/select.h>
5
6  int main(int argc, char *argv[])
7  {
8      int op1 = open(argv[1], O_RDWR | O_CREAT, S_IRWXU | S_IRWXG | S_IRWXO);
9      if (op1 == -1)
10     {
11         perror("Error opening file 1");
12     }
13     int op2 = open(argv[2], O_RDWR | O_CREAT, S_IRWXU | S_IRWXG | S_IRWXO);
14     if (op2 == -1)
15     {
16         perror("Error opening file 1");
17     }
18     fd_set setone;
19     int size = 100;
20     char buff[size];
21     int maxfd;
22     int nfds;
23     FD_ZERO(&setone);
24     FD_SET(op1, &setone);
25     FD_SET(op2, &setone);
26     maxfd = (op1 > op2) ? op1 : op2;
27     nfds = select(maxfd + 1, &setone, NULL, NULL, NULL);
28     printf("Number of ready files is %d\n", nfds);
29     return 0;
30 }
```

Ouput :

SP LAB REPROT

```
DEBUG CONSOLE  TERMINAL  JUPYTER: VARIABLES  COMMENTS

~/Documents/SP_Lab/Lab 07$ gcc T3.c -o t3
~/Documents/SP_Lab/Lab 07$ ./t3 f1 f2
files is 2
~/Documents/SP_Lab/Lab 07$
```

Task 04:

Write a program that monitors N files using 'select'.

Code :

```
C T4.c > main(int, char *[])
1  #include <stdio.h>
2  #include <unistd.h>
3  #include <fcntl.h>
4  #include <sys/select.h>
5
6  int main(int argc, char *argv[])
7  {
8      int op[argc];
9      for (int i = 0; i < argc; i++)
10     {
11         op[i] = open(argv[i], O_RDWR | O_CREAT, S_IRWXU | S_IRWXG | S_IRWXO);
12     }
13     fd_set setone;
14
15     int maxfd = 0;
16     int nfd;
17     FD_ZERO(&setone);
18     for (int i = 0; i < argc; i++)
19     {
20         FD_SET(op[i], &setone);
21     }
22     for (int i = 0; i < argc; i++)
23     {
24         maxfd = (op[i] > op[i + 1]) ? op[i] : op[i + 1];
25     }
26     nfd = select(maxfd + 1, &setone, NULL, NULL, NULL);
27     printf("Number of ready files is %d\n", nfd);
28     return 0;
29 }
```

Ouput :

```
DEBUG CONSOLE  TERMINAL  JUPYTER: VARIABLES  COMMENTS

~/Documents/SP_Lab/Lab 07$ gcc T4.c -o t4
~/Documents/SP_Lab/Lab 07$ ./t4 file1 file2 file3 file4
files is 4
~/Documents/SP_Lab/Lab 07$
```

Task 05:

Write a function that creates a delay of N seconds using select function. Pass N as an argument to the function .

Code :

```
C T3.c  C T4.c  C T5.c  X
C T5.c > main(int, char * [])
1  #include <stdio.h>
2  #include <unistd.h>
3  #include <fcntl.h>
4  #include <sys/select.h>
5  #include <stdlib.h>
6
7  int main(int argc, char *argv[])
8  {
9      struct timeval timeout;
10     timeout.tv_sec = atoi(argv[1]);
11     int nfd = select(1, NULL, NULL, NULL, &timeout);
12     printf("Number of ready files is %d\n", nfd);
13     return 0;
14 }
```

Ouput :

```
DEBUG CONSOLE  TERMINAL  JUPYTER: VARIABLES  COMMENTS
~/Documents/SP_Lab/Lab 07$ gcc T5.c -o t5
~/Documents/SP_Lab/Lab 07$ ./t5 5
files is 0
~/Documents/SP_Lab/Lab 07$
```