Lab Report 06

**Systems Programming Lab**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**302L-Systems Programming Lab**

**Department of Computer System Engineering**

**University of Engineering and Technology Peshawar**

Submitted to: **Engr. Abdullah Hamid**

Submitted by: **Muhammad Saad**

Reg ID: **21PWCSE1997**

**DCSE, Batch 23, Section “B”**

**Lab 06: Unix I/O**

**Task 01:**Write a program for parallel file copying using multiple processes. (First check if the no of files entered are even and if yes then create a child process for each pair and read from one and write to other.

#include *<stdio.h>*

#include *<stdlib.h>*

#include *<unistd.h>*

#include *<sys/types.h>*

#include *<sys/wait.h>*

#include *<fcntl.h>*

void copyFiles(**const** char \*source, **const** char \*destination) {

int sourceFile, destinationFile;

char buffer[1024];

int bytesRead;

sourceFile = open(source, O\_RDONLY);

**if** (sourceFile == -1) {

perror("Error opening source file");

exit(EXIT\_FAILURE);

}

destinationFile = open(destination, O\_WRONLY | O\_CREAT | O\_TRUNC, S\_IRUSR | S\_IWUSR | S\_IRGRP | S\_IROTH);

**if** (destinationFile == -1) {

perror("Error opening destination file");

close(sourceFile);

exit(EXIT\_FAILURE);

}

**while** ((bytesRead = read(sourceFile, buffer, **sizeof**(buffer))) > 0) {

**if** (write(destinationFile, buffer, bytesRead) != bytesRead) {

perror("Error writing to destination file");

close(sourceFile);

close(destinationFile);

exit(EXIT\_FAILURE);

}

}

**Task 02:**

Implement cat utility.

**Code:**

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

**if** (argc % 2 != 1) {

fprintf(stderr, "Usage: %s source1 dest1 source2 dest2 ...", argv[0]);

exit(EXIT\_FAILURE);

}

int numPairs = (argc - 1) / 2;

**for** (int i = 0; i < numPairs; i++) {

pid\_t pid = fork();

**if** (pid == -1) {

perror("Error forking process");

exit(EXIT\_FAILURE);

} **else** **if** (pid == 0) {

copyFiles(argv[i \* 2 + 1], argv[i \* 2 + 2]);

exit(EXIT\_SUCCESS);

}

}

**for** (int i = 0; i < numPairs; i++) {

wait(NULL);

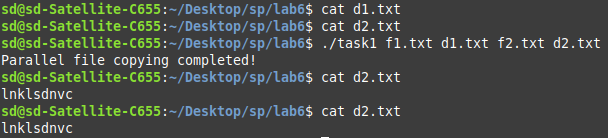
}

printf("Parallel file copying completed!**\n**");

**return** 0;

}

**Output:**



**Task 2:** Implement cat utility in C programming language.

#include *<stdio.h>*

#include *<unistd.h>*

#include *<string.h>*

#include *<fcntl.h>*

#include *<sys/stat.h>*

#include *<stdlib.h>*

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

{

int fd1, fd2, rd, wr;

char buffer[100];

**if** (argc == 1)

{

fd1 = STDIN\_FILENO;

fd2 = STDOUT\_FILENO;

rd = read(fd1, buffer, 100);

wr = write(fd2, buffer, rd);

}

**if** (argc == 2)

{

fd1 = open(argv[1], O\_RDONLY);

fd2 = STDOUT\_FILENO;

**for** (;;)

{

rd = read(fd1, buffer, 100);

**if** (rd == 0)

**break**;

wr = write(fd2, buffer, rd);

}

}

**if** (argc == 3)

{

fd1 = STDIN\_FILENO;

fd2 = open(argv[2], O\_APPEND, S\_IRWXU);

rd = read(fd1, buffer, 100);

wr = write(fd2, buffer, rd);

}

**if** (argc == 4)

{

fd1 = open(argv[1], O\_RDONLY);

fd2 = open(argv[2], O\_APPEND, S\_IRWXU);

**for** (;;)

{

rd = read(fd1, buffer, 100);

**if** (rd == 0)

**break**;

wr = write(fd2, buffer, rd);

}

}

close(fd1);

close(fd2);

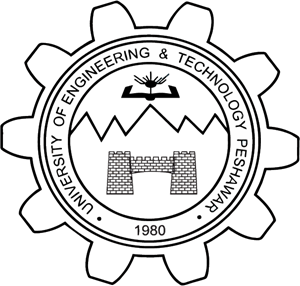
**return** 0;

}

**Output:**

A screenshot of a computer program

Description automatically generated



Lab Report 07

**Systems Programming Lab**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**302L-Systems Programming Lab**

**Department of Computer System Engineering**

**University of Engineering and Technology Peshawar**

Submitted to: **Engr. Abdullah Hamid**

Submitted by: **Muhammad Saad**

Reg ID: **21PWCSE1997**

**DCSE, Batch 23, Section “B”**

**Lab 07: Multiple File copying**

**Task 01:**

Write a program that copies 2 files sequentially.

**Code:**

#include *<stdio.h>*

#include *<unistd.h>*

#include *<string.h>*

#include *<fcntl.h>*

#include *<sys/stat.h>*

#include *<stdlib.h>*

#include *<errno.h>*

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

{

int fd1 = open(argv[1], O\_RDONLY), fd2 = open(argv[2], O\_RDONLY);

**if**(fd1==-1 || fd2==-1)

perror("Error: ");

char bff1[100], bff2[100];

int rd1 = read(fd1, bff1, 100);

int rd2 = read(fd2, bff2, 100);

**if**(rd1==-1 || rd2==-1)

perror("Error: ");

int wr1 = write(1, bff1, rd1), wr2 = write(1, bff2, rd2);

**if**(wr1==-1 || wr2==-1)

perror("Error: ");

**return** 0;

}

**Output:**

A screenshot of a computer program

Description automatically generated

**Task 02:**

Write a program that monitors 2 files using select.

**Code:**

#include *<stdio.h>*

#include *<unistd.h>*

#include *<string.h>*

#include *<fcntl.h>*

#include *<sys/stat.h>*

#include *<stdlib.h>*

#include *<errno.h>*

#include *<sys/select.h>*

int main()

{

int fd1 = open("f1.txt", O\_RDONLY);

int fd2 = open("f2.txt", O\_RDONLY);

char buffer[100];

fd\_set readset;

FD\_ZERO(&readset);

FD\_SET(fd1, &readset);

FD\_SET(fd2, &readset);

int maxfd = fd1 > fd2 ? fd1 : fd2;

int nrfds = select(maxfd + 1, &readset, NULL, NULL, NULL);

**if** (FD\_ISSET(fd1, &readset))

{

printf("f1.txt is ready!!**\n**");

}

**if** (FD\_ISSET(fd2, &readset))

{

printf("f2.txt is ready!!**\n**");

}

**return** 0;

}

**Output:**



**Task 03:**

Write a program that monitors N files using Select.

**Code:**

#include *<stdio.h>*

#include *<unistd.h>*

#include *<string.h>*

#include *<fcntl.h>*

#include *<sys/stat.h>*

#include *<stdlib.h>*

#include *<errno.h>*

#include *<sys/select.h>*

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

{

int fds[argc - 1];

**for** (int i = 1; i <= argc - 1; i++)

{

fds[i - 1] = open(argv[i], O\_RDONLY);

}

char buffer[100];

fd\_set readset;

FD\_ZERO(&readset);

int maxfd = 0;

**for** (int i = 0; i < argc - 1; i++)

{

FD\_SET(fds[i], &readset);

**if** (maxfd < fds[i])

maxfd = fds[i];

}

int nrfds = select(maxfd, &readset, NULL, NULL, NULL);

**for** (int i = 0; i < argc - 1; i++)

{

**if** (FD\_ISSET(fds[i], &readset))

{

printf("File No %d is ready.**\n**", i+1);

}

}

**return** 0;

}

**Output:**



**Task 04:**

Write a program that monitors using fork () 2 times.

**Code:**

#include *<stdio.h>*

#include *<unistd.h>*

#include *<string.h>*

#include *<fcntl.h>*

#include *<sys/stat.h>*

#include *<stdlib.h>*

#include *<errno.h>*

#include *<sys/select.h>*

int main()

{

int fd1 = open("f1.txt", O\_RDONLY);

int pid1 = fork(), pid2;

**if** (pid1 > 0)

pid2 = fork();

char buffer[100];

**if** (pid1 == 0 || pid2 == 0)

{

fd\_set readset;

FD\_ZERO(&readset);

FD\_SET(fd1, &readset);

int nrfds = select(fd1+1, &readset, NULL, NULL, NULL);

**if** (FD\_ISSET(fd1, &readset))

{

int rd = read(fd1, buffer, 100);

int wr = write(1, buffer, rd);

}

}

**return** 0;

}

**Output:**



**Task 05:**

Write function that creates delay of N seconds using select function call.

**Code:**

#include <stdio.h>

#include <sys/select.h>

#include <sys/time.h>

#include <stdlib.h>

#include <signal.h>

void delay\_using\_select(int seconds)

{

struct timeval timeout;

timeout.tv\_sec = seconds;

timeout.tv\_usec = 0;

fd\_set empty\_set;

FD\_ZERO(&empty\_set);

select(0, &empty\_set, NULL, NULL, &timeout);

}

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

{

int x = alarm(100);

delay\_using\_select(atoi(argv[1]));

printf("Delayed For %d seconds\n", atoi(argv[1]));

int y = alarm(2);

printf("%d", 100 - y);

}

**Ouput:**



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_