

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ЛЬВІВСЬКИЙ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ІВАНА ФРАНКА**

Факультет електроніки і комп'ютерних технологій

Кафедра радіоелектронних і
комп'ютерних систем

Звіт

про виконання лабораторної роботи №7

«Реалізація міжпроцесової взаємодії на основі інтерфейсу файлової системи»

Виконав:

студент групи ФеІ-23

Ковальчук Дмитро

Викладач: Сінькевич О. О.

Львів 2018

Тема: Реалізація міжпроцесової взаємодії на основі інтерфейсу файлової системи.

Мета: реалізувати міжпроцесову взаємодію на основі інтерфейсу файлової системи.

Завдання №1.

Розробіть систему обміну даними про поточну температуру повітря для Linux і Windows XP з використанням відображуваної пам'яті

```
#include <pthread.h>
#include <unistd.h>
#include <sys/mman.h>
#include <fcntl.h>
#include <stdio.h>

void* server(void* par)
{
    int fdl;
    int* map;
    fdl = open("tmp", O_RDWR | O_CREAT, 0644);
    lseek(fdl, sizeof(int), SEEK_SET);
    write(fdl, "", 1);
    map = (int*)mmap(0, sizeof(int), PROT_WRITE | PROT_READ, MAP_SHARED, fdl, 0);
    close(fdl);
    map[0] = 5;
    while(1)
    {
        printf("Server temperature is : %i\n", map[0]);
        if (map[0] == 0) break;
        usleep(200);
        map[0]--;
    }
}

int index = 0;
pthread_mutex_t i = PTHREAD_MUTEX_INITIALIZER;
int getIndex(){

    index++;

    return index;
}

void* client(void* par) {
    usleep(200);
    int fdl;
    int* map;

    pthread_mutex_lock(&i);
    int index = getIndex();
    pthread_mutex_unlock(&i);

    fdl = open("./tmp", O_RDONLY);
    map = (int*)mmap(0, sizeof(int), PROT_READ, MAP_SHARED, fdl, 0);
    close(fdl);

    while(1){
        printf("client %i temperature is : %i\n", index, map[0]);

        if (map[0] == 0)

            break;
        usleep(200);
    }
}
```

```

}
}

int main() {

int clientCount = 3;
pthread_t serverThread;
pthread_t clientThreads[clientCount];

pthread_create(&serverThread, NULL, server, NULL);
for (int i = 0; i < clientCount; ++i)
pthread_create(&clientThreads[i], NULL, client, NULL);

for (int i = 0; i < clientCount; ++i)
pthread_join(clientThreads[i], NULL);
pthread_join(serverThread, NULL);

return 0;
}

```

У даній програмі створено один потік-менеджер та три потоки-клієнти. Потік-менеджер задає значення температури від 5 до 1, а кожен з трьох потоків-клієнтів відображає ці зміни. Дійшовши до нуля, робота завершується.

```

Server temperature is : 5
client 1 temperature is : 5
client 2 temperature is : 5
client 3 temperature is : 5
client 1 temperature is : 5
client 2 temperature is : 5
client 3 temperature is : 5
Server temperature is : 4
client 1 temperature is : 4
client 2 temperature is : 4
client 3 temperature is : 4
Server temperature is : 3
client 1 temperature is : 3
client 2 temperature is : 3
client 3 temperature is : 3
Server temperature is : 2
client 1 temperature is : 2
client 2 temperature is : 2
client 3 temperature is : 2
Server temperature is : 1
client 1 temperature is : 1
client 2 temperature is : 1
client 3 temperature is : 1
Server temperature is : 0
client 1 temperature is : 0
client 2 temperature is : 0
client 3 temperature is : 0

```

Завдання №2

Розробіть просту клієнт-серверну систему для Linux і Windows XP з використанням поіменованих каналів.

```

#include <sys/types.h>
#include <sys/stat.h>
#include <pthread.h>

```

```

#include <unistd.h>
#include <sys/mman.h>
#include <fcntl.h>
#include <stdio.h>
#include <string.h>

void* server(void* par) {
mkfifo("serverFIFO", 0644);

int file;
char fileName[100], fileEntry[10000];
int length;

while(1) {

file = open("serverFIFO", O_RDONLY);
length = read(file, fileName, sizeof(fileName));
close(file);

if (strcmp(fileName, "exit") == 0)
break;

file = open(fileName, O_RDONLY);
length = read(file, fileEntry, sizeof(fileEntry));
close(file);

if (length > 0) {
file = open("clientFIFO", O_WRONLY);
write(file, fileEntry, sizeof(fileEntry));
close(file);
} else {
file = open("clientFIFO", O_WRONLY);
char erorrStr[] = "Errorr while try to open file\nlaybe file doesn't exist!";
write(file, erorrStr, sizeof(erorrStr));
close(file);
}
}

void* client(void* par) {
mkfifo("clientFIFO", 0644);

int file;
char input[100], fileEntry[10000];
int length;

while(1) {
printf("Enter file name : ");
scanf("%s", input);
printf("\n");

file = open("serverFIFO", O_WRONLY);
length = write(file, input, sizeof(input));
close(file);

if (strcmp(input, "exit") == 0)
break;
file = open("clientFIFO", O_RDONLY);
length = read(file, fileEntry, sizeof(fileEntry));
close(file);
if (length > 0) {
fileEntry[length] = '\0';

```

```

printf("File entry : %s\n", fileEntry);

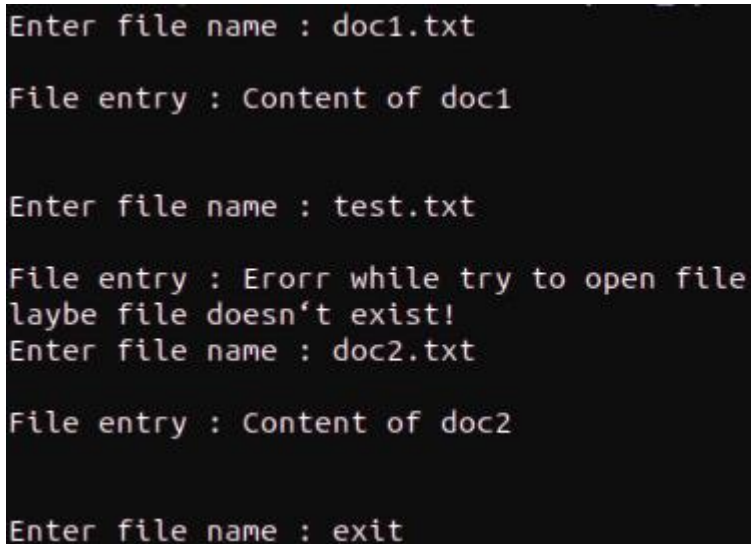
}
}
unlink("clientFIFO");
}
int main()
{
pthread_t serverThread;
pthread_t clientThread;

pthread_create(&serverThread, NULL, server, NULL);
pthread_create(&clientThread, NULL, client, NULL);

pthread_join(clientThread, NULL);
pthread_join(serverThread, NULL);

return 0;
}

```



```

Enter file name : doc1.txt
File entry : Content of doc1

Enter file name : test.txt
File entry : Errorr while try to open file
laybe file doesn't exist!
Enter file name : doc2.txt
File entry : Content of doc2

Enter file name : exit

```

Створено два потоки: сервер та клієнт. Ми маємо набір з двох файлів: file1 та file2. При вводі назви файлу, сервер зчитує вміст файлу та виводить його на екран. При вводі назви неіснуючого файлу видається помилка. При вводі слова exit робота завершається.

Висновок: на лабораторній роботі було розглянуто інтерфейси файлової системи, реалізовано клієнт серверну систему на основі цих інтерфейсів. Створив два потоки для сервера клієнта, зробив набір з двох файлів.