

Міністерство освіти та науки України
Львівський національний університет ім. Івана Франка
Факультет електроніки та 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
Речинський О.С.
Перевірив:
ас. Сінькевич О.О.

Львів – 2019

Додаткова інформація

Варіант: №6;
Дистрибутив: Ubuntu 19.04;
Девайс: Acer Aspire 5 A515-51G-58BE;
Процесор: Intel Core i5-8250U;
Графіка: NVIDIA GeForce MX130;
Оперативна пам'ять: 8 ГБ;
Постійна пам'ять: 1 TB HDD.

Лабораторна робота №7. Реалізація міжпроцесової взаємодії на основі інтерфейсу файлової системи

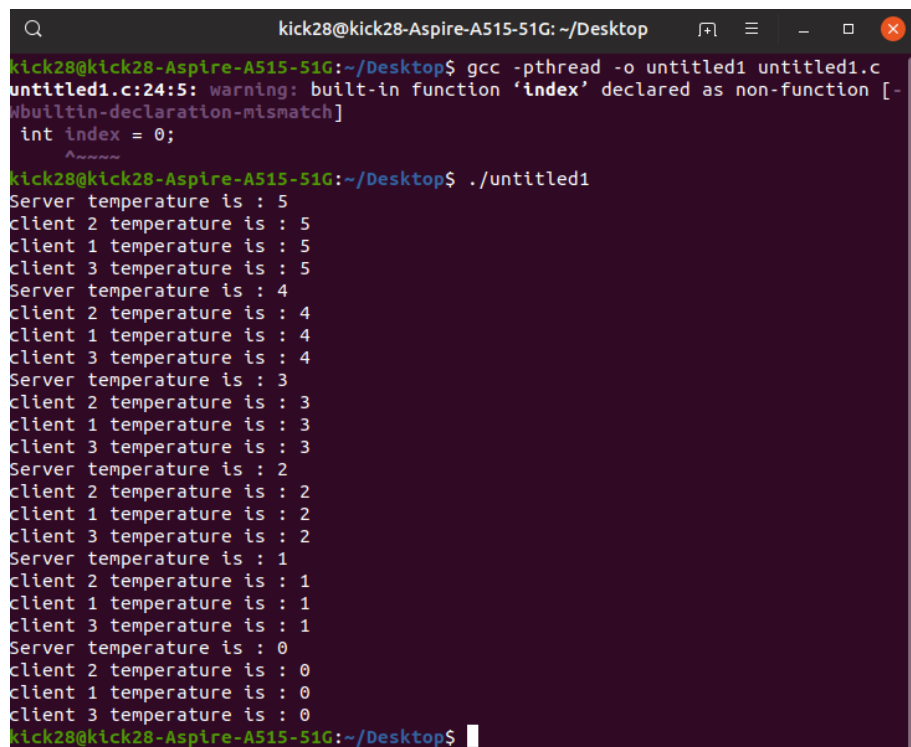
Мета

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

Результат



```
kick28@kick28-Aspire-A515-51G: ~/Desktop
kick28@kick28-Aspire-A515-51G:~/Desktop$ gcc -pthread -o untitled1 untitled1.c
untitled1.c:24:5: warning: built-in function 'index' declared as non-function [-Wbuiltin-declaration-mismatch]
    int index = 0;
    ^~~~~~
kick28@kick28-Aspire-A515-51G:~/Desktop$ ./untitled1
Server temperature is : 5
client 2 temperature is : 5
client 1 temperature is : 5
client 3 temperature is : 5
Server temperature is : 4
client 2 temperature is : 4
client 1 temperature is : 4
client 3 temperature is : 4
Server temperature is : 3
client 2 temperature is : 3
client 1 temperature is : 3
client 3 temperature is : 3
Server temperature is : 2
client 2 temperature is : 2
client 1 temperature is : 2
client 3 temperature is : 2
Server temperature is : 1
client 2 temperature is : 1
client 1 temperature is : 1
client 3 temperature is : 1
Server temperature is : 0
client 2 temperature is : 0
client 1 temperature is : 0
client 3 temperature is : 0
kick28@kick28-Aspire-A515-51G:~/Desktop$
```

Код програми

```
#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;
}

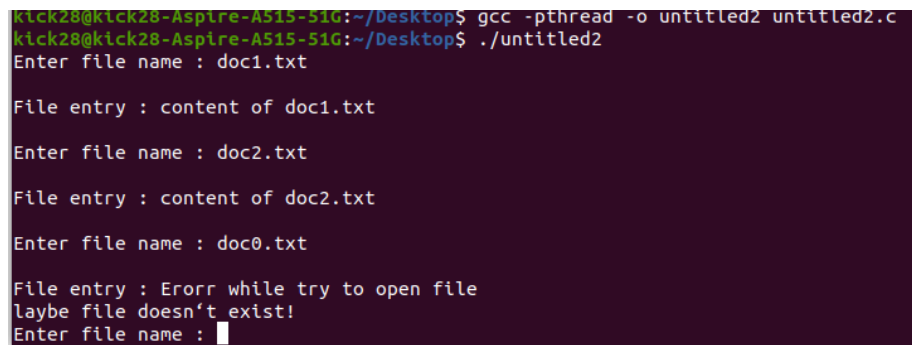
```

Ця програма має 1 потік-менеджер та 3 потоки-клієнти: перший потік задає значення температури в діапазоні від 5 до 1, а 3 інші потоки відображають зміни. Як тільки лічильник доходить до 0, програма завершує роботу

Завдання № 2

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

Результат



```

kick28@kick28-Aspire-A515-51G:~/Desktop$ gcc -pthread -o untitled2 untitled2.c
kick28@kick28-Aspire-A515-51G:~/Desktop$ ./untitled2
Enter file name : doc1.txt

File entry : content of doc1.txt

Enter file name : doc2.txt

File entry : content of doc2.txt

Enter file name : doc0.txt

File entry : Error while try to open file
laybe file doesn't exist!
Enter file name : 

```

Код програми

```

#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;
}

```

В цій програмі було створено 2 потоки: клієнт та сервер. Маючи файли (в нашому випадку 2 файли doc1.txt & doc2.txt) та ввівши їх назву, сервер зчитує вміст файлу і виводить вміст на екран. Якщо відкривається неіснуючий файл, то програма видає помилку. Також ввівши ключове слово exit, можна вийти з програми.

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.