

OPERATING SYSTEMS

CECSC09 - 1



Submitted by :-

NAME: Harshit Gupta

ROLL NO: 2019UCO1580

Program 2

Program to show the concept of multithreading

- Threads are the basic unit of computation for a CPU executing a program
- A Process can be thought of as analogous to a building construction being done and Threads can be thought of as the numerous workers executing different tasks like painting, electricity fittings, laying bricks etc.
- Just the difference in a computer system is that the process actually gets converted to viz. A web browser being opened and threads are the workers which execute loading of graphics, fetching of text content, processing server requests, etc.
- **PThreads** or POSIX threads is an implementation of the threading features in different operating systems and its use in Windows operating system is depicted by the following program
- The objective of this experiment is to copy the contents of one file into another through multiple threads where it creates 3 different threads for **input names**, **copying** and **termination** of the program

CODE - parallel execution achieved in the context of **pre opening the output file in a different thread.**

```
#include<bits/stdc++.h>
#include<thread>
#include<chrono>
#include<fstream>
using namespace std;
char source[20];
char destination[20];
FILE *fs, *ft;
int res;
string output;
int main() {
    // making three functions for
```

```

// 1. input
// 2. copying
// 3. termination message

// for input names
output="";
res = 0;
auto f1 = []() {
    cout<<"\nEnter the Name of Source File: ";
    cin>>source;
    fs = fopen(source,"r"); // read pointer
    if(fs==NULL) {
        cout<<"\nCould not open source file!";
        res = -1;
    }
    else
        cout<<"***INPUT OPENED SUCCESSFULLY***\n";
};

// for copying files
auto f2 = []() {
    ft = fopen("output.txt","w"); // write pointer
    cout<<"***OUTPUT OPENED SUCCESSFULLY***\n";
    return;
};

// for termination
auto f3 = []() {
    char ch = fgetc(fs); // get character
    while(ch != EOF)
    {
        fputc(ch, ft); // put character
        ch = fgetc(fs);
    }
    cout<<"\nFile copied successfully.";
    fclose(fs);
    fclose(ft);
    cout<<endl;
};

```

```

thread t2(f2);
thread::id t2_id = t2.get_id();
thread t1(f1);
thread::id t1_id = t1.get_id();

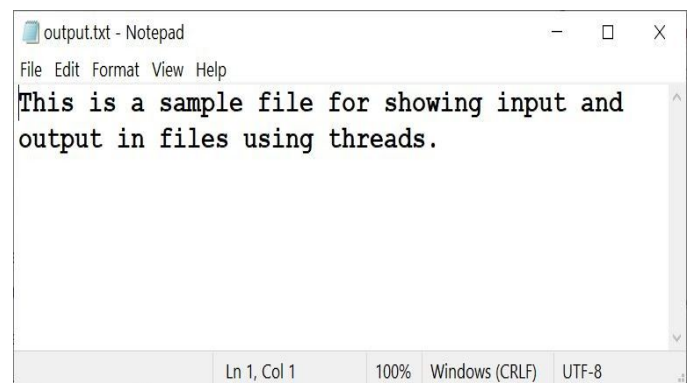
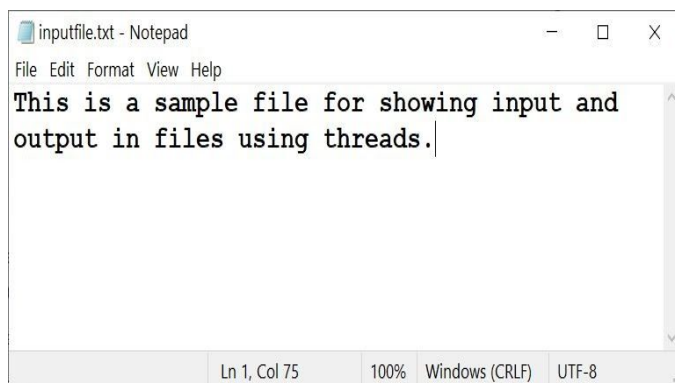
t1.join();
t2.join();
cout<<"\nInput Thread with id "<<t1_id<<" completed";
cout<<"\nOutput Thread with id "<<t2_id<<" completed";

if(res!=-1)
{
thread t3(f3);
thread::id t3_id = t3.get_id();
t3.join();
cout<<"\nCopy & Termination Thread with id "<<t3_id<<" completed";
// cout<<"Output : "<<output<<endl;
}
else
{cout<<"\nCould not open file name "<<source<<endl;
cout<<"\nIncomplete execution.";}

return 0;
}

```

FILES



TERMINAL : contains successful and unsuccessful execution

```
PS D:\IV Semester\OS\LAB\Lab2> g++ -std=c++11 lab2.cpp -pthread -o lab2
PS D:\IV Semester\OS\LAB\Lab2> ./lab2
***OUTPUT OPENED SUCCESSFULLY***

Enter the Name of Source File: inputfile.txt
***INPUT OPENED SUCCESSFULLY***

Input Thread with id 3 completed
Output Thread with id 2 completed
File copied successfully.

Copy & Termination Thread with id 4 completed
PS D:\IV Semester\OS\LAB\Lab2> ./lab2
***OUTPUT OPENED SUCCESSFULLY***

Enter the Name of Source File: input.txt

Could not open source file!
Input Thread with id 3 completed
Output Thread with id 2 completed
Could not open file name input.txt

Incomplete execution.
PS D:\IV Semester\OS\LAB\Lab2> █
```