

# Computer Network

## Project 1

2015313697

황영준

### 1. Development environments

<OS version>

Ubuntu 18.04.3 LTS

<Programming language and Compiler version>

C++ / g++ 7.4.0

```
hwang@hwang-VirtualBox:~/3-2/network/proj1$ cat /etc/issue
Ubuntu 18.04.3 LTS \n \l

hwang@hwang-VirtualBox:~/3-2/network/proj1$ g++ --version
g++ (Ubuntu 7.4.0-1ubuntu1~18.04.1) 7.4.0
Copyright (C) 2017 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

hwang@hwang-VirtualBox:~/3-2/network/proj1$ g++ ans.cpp -pthread
```

### 2. Algorithm and data structures

<main function>

```
52 ▼ int main(){
53     /// var
54     int temp=1;
55     string input,output;
56     vector<thread> working_thread;
57
58 ▼     while(temp++){
59         /// get file names
60         cout << "Input the file name: ";
61         cin >> input;
62         cout << "Input the new name: ";
63         cin >> output;
64         /// create thread
65         working_thread.push_back(thread(copy_file,input,output));
66         working_thread[temp-2].detach();
67     }
68 }
```

1. while문을 반복하며 input, output 파일이름을 string으로 받는다.
2. 두 개의 파일이름 string을 받으면 'copy\_file' function을 수행하는 thread를 생성하고 string 2개를 인자로 전달한다.

3. Vector<thread> 배열에 thread를 추가하고, 해당 thread가 master thread와 상관없이 실행, 종료 될 수 있게 detach한다.

<copy\_file/thread>

```
17 void* copy_file(std::string input, std::string output){
18     /// var
19     FILE *input_F,*output_F;
20     char buffer[buffer_size];
21     size_t len;
22     system_clock::time_point end;
23     duration<double> sec;
24     /// record start log
25     log_lock.lock();
26     end=system_clock::now();
27     sec=end-start;
28     fprintf(log_F,"%2f start copying %s to %s\n",
29         sec,input.c_str(),output.c_str());
30     fflush(log_F);
31     log_lock.unlock();
32     /// open input and output files
33     input_F=fopen(input.c_str(),"rb");
34     output_F=fopen(output.c_str(),"wb");
35     /// read input -> copy to buffer -> write to new file
36     while((len=fread(buffer,sizeof(char),sizeof(buffer),input_F))>0){
37         fwrite(buffer,sizeof(char),len,output_F);
38     }
39     /// close input and output file
40     fclose(input_F);
41     fclose(output_F);
42     /// record end log
43     log_lock.lock();
44     end=system_clock::now();
45     sec=end-start;
46     fprintf(log_F,"%2f %s is copied completely\n",
47         sec,output.c_str());
48     fflush(log_F);
49     log_lock.unlock();
50 }
51
```

1. Thread를 통해 file copy가 시작됨을 기록하기 위해 system\_clock으로부터 시간을 받아온 후 log.txt에 시간, input, output 파일명을 기록한다.
  - 이 때 프로그램의 종료조건이 따로 없고 강제로 종료될 경우에도 log의 기록을 보장하기 위해 fflush를 통해 바로 log파일에 기록한다.
  - 각 thread가 동시에 log파일에 접근을 하면 제대로 write 되지 않을 수 있으므로 log파일을 global variable로 shared memory로 관리하고, mutex lock을 통해 log파일의 접근을 critical section으로 설정한다.
2. Input file의 내용을 thread의 buffer에 가져온 후, buffer의 내용을 output file에 fwrite를 통해 기록한다.

3. Input file을 끝까지 읽으면 파일의 복사가 끝난 것 이므로 file copy가 종료되었음을 기록하는 log를 남긴다.

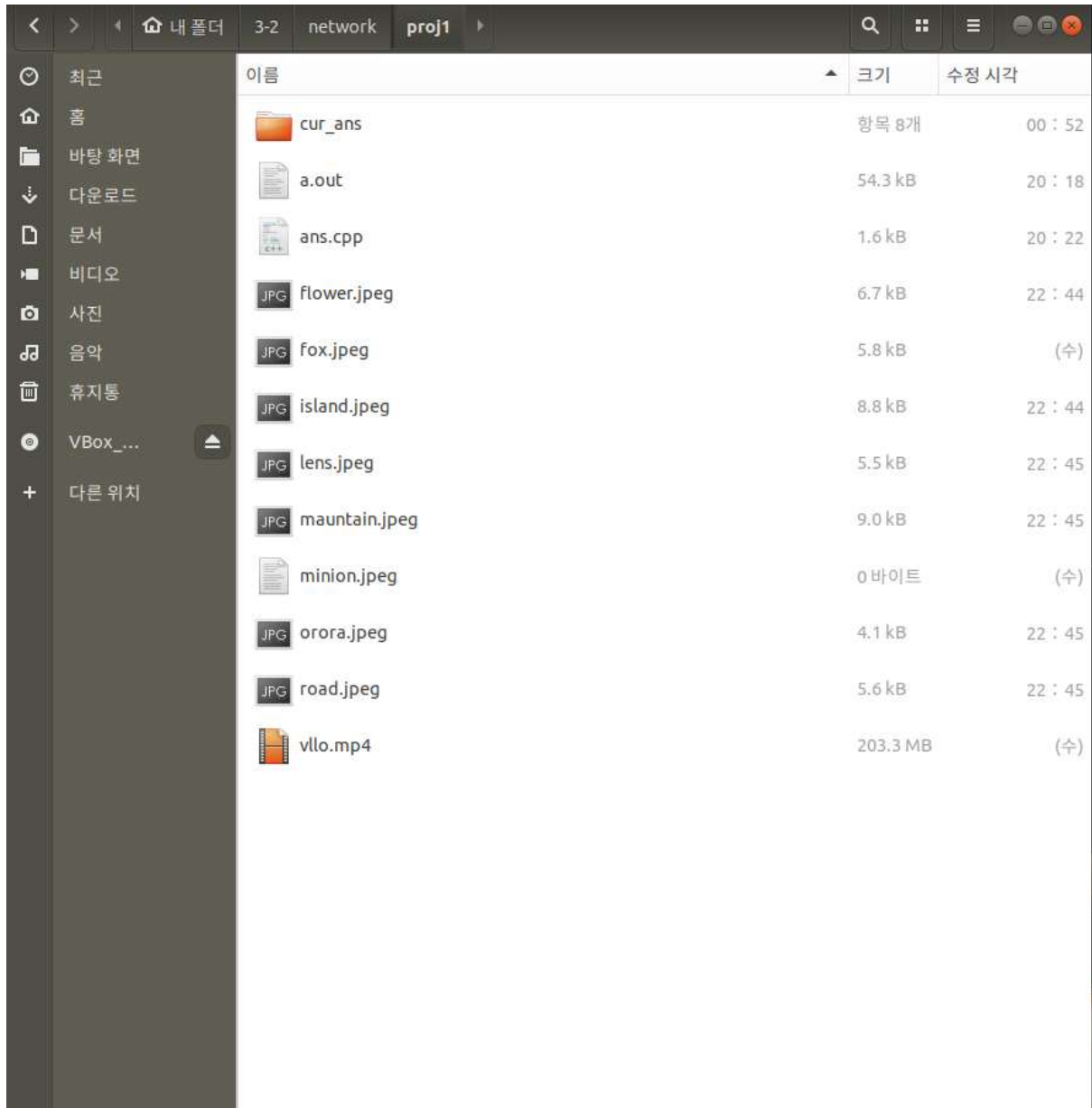
<STL and global variable>

```
1  #include <iostream>
2  #include <stdio.h>
3  #include <string.h>
4  #include <thread>
5  #include <mutex>
6  #include <vector>
7  #include <chrono>
8  #define buffer_size 1024
9
10 using namespace std;
11 using namespace chrono;
12 //// global var
13 mutex log_lock;
14 FILE *log_F=fopen("log.txt","wb");
15 system_clock::time_point start=system_clock::now();
```

1. Concurrency를 위한 <thread>와 log file에 접근할 때 critical section을 위한 <mutex>, 시간 계산을 위한 <chrono> library.
2. Thread의 buffer size는 1024로 설정.
3. 모든 thread가 log.txt의 파일 포인터를 공유하여 copy기록들이 순차적으로 작성되게 하기위해 log file의 포인터는 global variable로 선언.
4. 프로그램의 시작을 기록하기 위한 time\_point start.

### 3. Program test

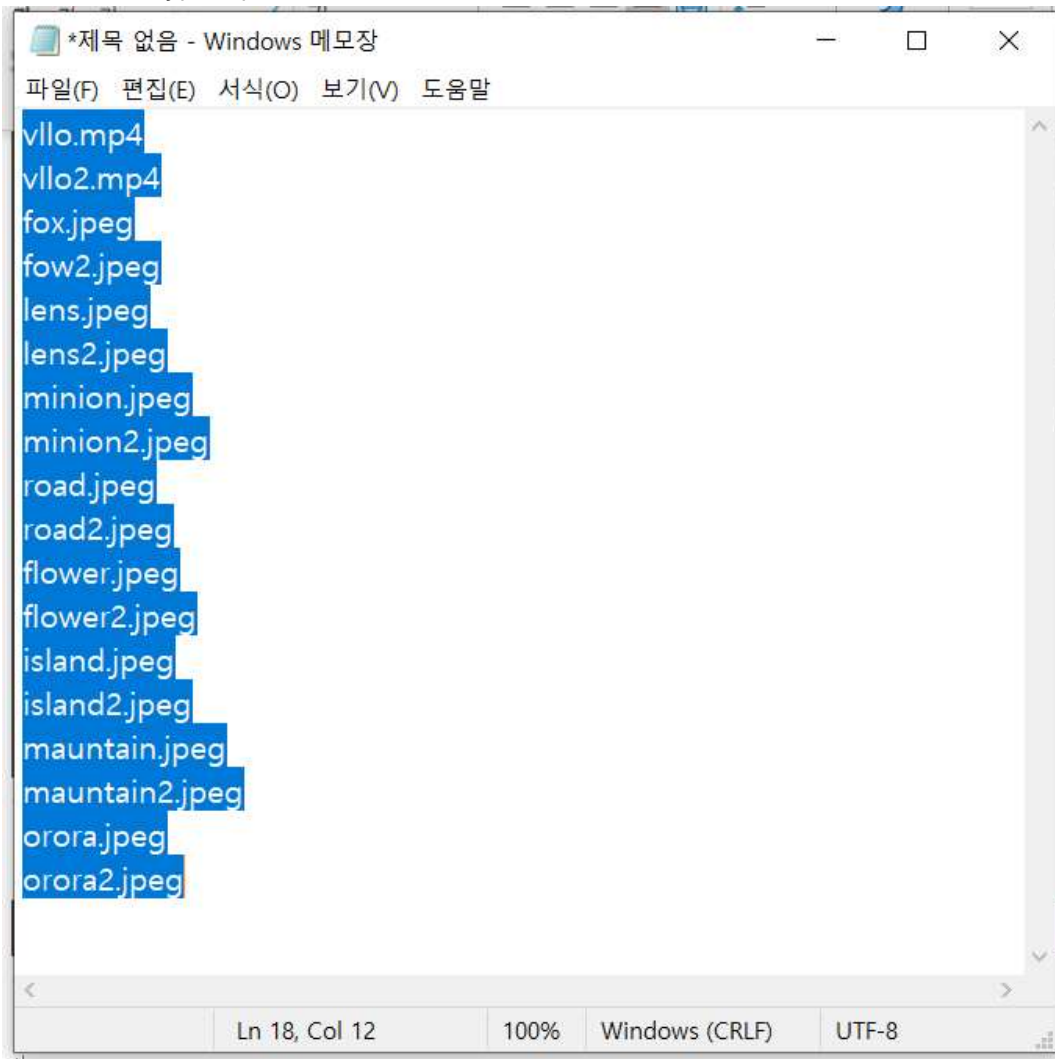
#### 1. <initial condition>



#### 2. <run program>

```
hwang@hwang-VirtualBox:~/3-2/network/proj1$ ./a.out
Input the file name: 
```

### 3. <type input>



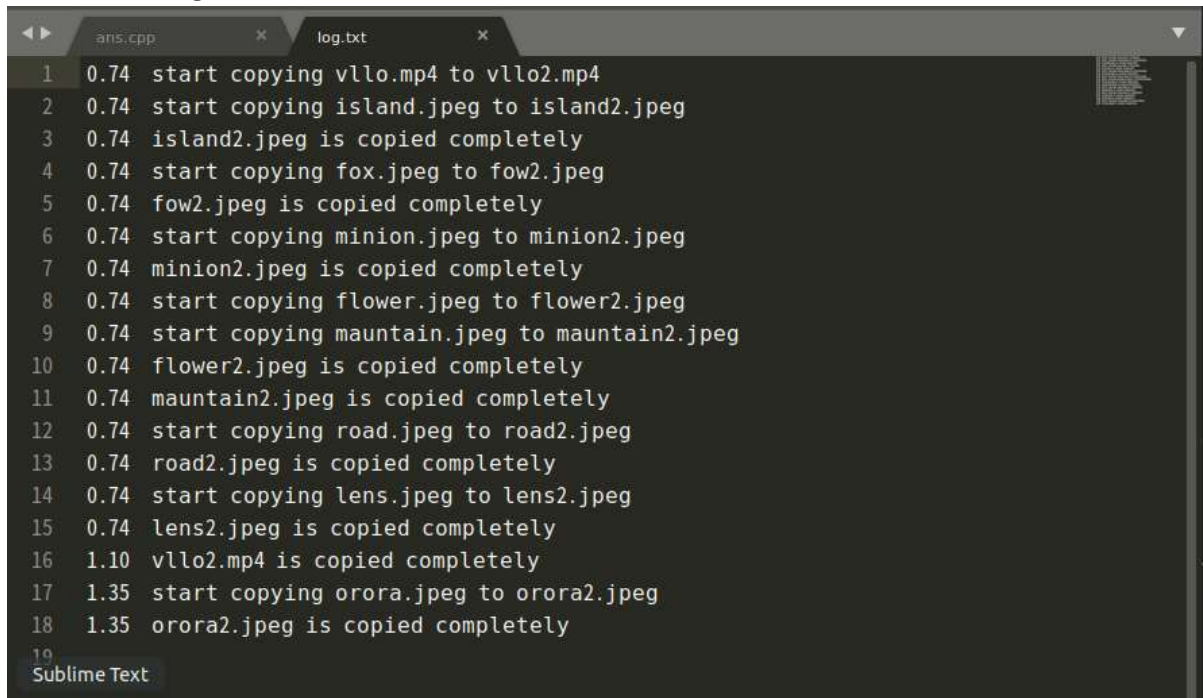
### <4. Ctrl + c & Ctrl + v>

```
hwang@hwang-VirtualBox:~/3-2/network/proj1$ ./a.out
Input the file name: vllo.mp4
vllo2.mp4
fox.jpeg
fow2.jpeg
lens.jpeg
lens2.jpeg
minion.jpeg
minion2.jpeg
road.jpeg
road2.jpeg
flower.jpeg
flower2.jpeg
island.jpeg
island2.jpeg
mauntain.jpeg
mauntain2.jpeg
orora.jpeg
orora2.jpeg
Input the new name: Input the file name: Input the new name: Input the file name: Inp
ut the new name: Input the file name: Input the new name: Input the file name: Input the new nam
e: Input the file name: Input the new name: Input the file name: Input the new name: Input the f
ile name: Input the new name: Input the file name: Input the new name: 
```

<5. Results>

< 3-2 network proj1 >				Q	≡	≡	⊞	⊞	⊞
	이름	크기	수정 시각						
최근	flower2.jpeg	6.7 kB	22 : 56						
홈	fow2.jpeg	5.8 kB	22 : 56						
바탕 화면	fox.jpeg	5.8 kB	(수)						
다운로드	island.jpeg	8.8 kB	22 : 44						
문서	island2.jpeg	8.8 kB	22 : 56						
비디오	lens.jpeg	5.5 kB	22 : 45						
사진	lens2.jpeg	5.5 kB	22 : 56						
음악	log.txt	767 바이트	22 : 56						
휴지통	mauntain.jpeg	9.0 kB	22 : 45						
VBox_...	mauntain2.jpeg	9.0 kB	22 : 56						
다른 위치	minion.jpeg	0 바이트	(수)						
	minion2.jpeg	0 바이트	22 : 56						
	orora.jpeg	4.1 kB	22 : 45						
	orora2.jpeg	4.1 kB	22 : 56						
	road.jpeg	5.6 kB	22 : 45						
	road2.jpeg	5.6 kB	22 : 56						
	vllo.mp4	203.3 MB	(수)						
	vllo2.mp4	203.3 MB	22 : 56						

<6. Log.txt>



```
1 0.74 start copying vllo.mp4 to vllo2.mp4
2 0.74 start copying island.jpeg to island2.jpeg
3 0.74 island2.jpeg is copied completely
4 0.74 start copying fox.jpeg to fow2.jpeg
5 0.74 fow2.jpeg is copied completely
6 0.74 start copying minion.jpeg to minion2.jpeg
7 0.74 minion2.jpeg is copied completely
8 0.74 start copying flower.jpeg to flower2.jpeg
9 0.74 start copying mauntain.jpeg to mauntain2.jpeg
10 0.74 flower2.jpeg is copied completely
11 0.74 mauntain2.jpeg is copied completely
12 0.74 start copying road.jpeg to road2.jpeg
13 0.74 road2.jpeg is copied completely
14 0.74 start copying lens.jpeg to lens2.jpeg
15 0.74 lens2.jpeg is copied completely
16 1.10 vllo2.mp4 is copied completely
17 1.35 start copying orora.jpeg to orora2.jpeg
18 1.35 orora2.jpeg is copied completely
19
Sublime Text
```