Network Assignment 2

2014313303

홍태하

개발환경 : Ubuntu 16.04, python 3.5.2, gcc 5.4.0

1. Code 실행방법

리눅스 터미널에 python3 2014313303.py 를 입력한다.

```
$ python3 2014313303.py
```

2. Code 설명

```
import time
from threading import Thread

start = time.time()
chunk_size = 1024 * 4096

def copy_file():
    file_name = input("Input the file name: ")
    new_name = input("Input the new name: ")

    print()

    f2 = open("log.txt", 'a')
    tm = int(time.time() - start)
    f2.write('{:02d}:{:02d}'.format((tm%3600 // 60), tm % 60)
          + "\tStart copying " + file_name + " to " + new_name + "\n")
    f2.close()

    t = Thread(target = copy_file)
    t.start()

    with open(file_name, 'rb') as f:
        with open(new_name, 'wb') as f1:
            with open("log.txt", 'a') as f2:
                while True:
                    buffer = f.read(chunk_size)
                    if not buffer: break
                    f1.write(buffer)

                tm = int(time.time() - start)
                f2.write('{:02d}:{:02d}'.format((tm%3600 // 60), tm % 60)
                        + "\t" + new_name + " is copied completely\n")

                f.close()
                f1.close()
                f2.close()

if __name__ == "__main__":
    copy_file()
```

읽어들일 파일의 이름을 file_name에 저장하고 copy할 파일의 이름을 new_name에 저장한다.

log.txt 를 이름으로 하는 파일을 a mode로 open한다.

프로그램 시작을 0초로 하여 지난 시간과 copy를 시작한다는 log를 log.txt에 기록한다. 시간은 분:초 로 나타난다.

open했던 log.txt 파일을 close한 후에 쓰레드를 생성한다. 새로 생긴 쓰레드는 copy_file 함수를 다시 돌게 되기 때문에 input을 다시 받는다. 그와 같이 부모 쓰레드는 input으로 받았던 파일들을 open하고 copy를 시작한다. Copy가 끝나면 log.txt파일에 시간과 copy가 끝났다는 log를 남기고 열렸던 파일 모두를 close해준다.

3. 실행 결과

```
taeha@ubuntu:~/Desktop/Network/Assignment2$ python3 2014313303.py
Input the file name: input/large.mp4
Input the new name: new_large.mp4

Input the file name: input/Sample.txt
Input the new name: new_Sample.txt

Input the file name: input/Video1.mov
Input the new name: new_Video1.mov

Input the file name: ^CException ignored in: <module 'threading' from '/usr/lib/python3.5/threading.py'
>
Traceback (most recent call last):
  File "/usr/lib/python3.5/threading.py", line 1288, in _shutdown
    t.join()
  File "/usr/lib/python3.5/threading.py", line 1054, in join
    self._wait_for_tstate_lock()
  File "/usr/lib/python3.5/threading.py", line 1070, in _wait_for_tstate_lock
    elif lock.acquire(block, timeout):
KeyboardInterrupt
```

```
1 00:08   Start copying input/large.mp4 to new_large.mp4
2 00:15   Start copying input/Sample.txt to new_Sample.txt
3 00:15   new_Sample.txt is copied completely
4 00:26   Start copying input/Video1.mov to new_Video1.mov
5 00:27   new_Video1.mov is copied completely
6 01:10   new_large.mp4 is copied completely
```

파일을 실행시켜 large.mp4, Sample.txt, Video1.mov 3가지 파일을 copy하였다. 여기서 large.mp4는 용량이 커서 copy에 오랜 시간이 걸리므로 8초에 copy를 시작하여 1분 10초에 끝났지만 나머지 2파일은 용량이 작아 copy 시작과 거의 동시에 완료가 된 모습이다. 이 프로그램은 실행한 사람이 ctrl C 와 같은 것으로 종료시켜 줄때까지 동작한다.