

# Introduction to Computer Networks

## Assignment 1: Concurrent File Copy Program

### 1. Goal

- Be able to handle a new copy request while copying files in parallel.
  - Using multi-threads ( one thread for each file transfer )

### 2. Development environments

- TA will evaluate your results on Linux.
- You can use C++ or Python (version 3.6+)
- You have to describe your development environments in detail in the report.

**\* This assignment is to give you an experience developing concurrent processing. You can use only basic libraries (ex: `stdlib.h`, `os`, `thread` etc. ), and must implement the main functionalities by yourself. If you want to use a special library, you must be confirmed via our anonymous google sheet.**

### 3. Functionalities to implement

- Assume that there are both text and binary files (text files, image files, and video files) in the directory where the program is located.
- Get two input strings;  
one for the source (existing) file name, and one for the destination (non-existent) name to be copied.
- The file must be read in 10 Kbytes increments using a loop.
- Must allow a new copy request while copying files in parallel
  - TA will evaluate up to 10 simultaneous multiple file copies
- If a file copy is completed, write the **file name** with the **completion time** in "log.txt".
  - The time is assumed to be zero seconds when the program starts.
- Enter 'exit' to terminate this program ( instead of a file name )

#### 4. Sample Results

```
Input the file name: GoodMovie.avi
Input the new name: GreateMovie.avi

Input the file name: BadMovie.avi
Input the new name: NotBadMovie.avi

Input the file name: exit
```

Figure 1. Console screen

```
3.00  Start copying GoodMovie.avi to GreateMovie.avi
10.00 Start copying BadMovie.avi to NotBadMovie.avi
32.00 NotBadMovie.avi is copied completely
47.00 GreateMovie.avi is copied completely
```

Figure 2. A sample of log.txt

#### 5. Submission

- The deadline is 4.5 (Sun) 23:59.
  - For delayed submissions, a penalty of -15 points applies every 24 hours. After 72 hours, you get zero points.
  - In the case of plagiarism, you will receive 0 points for the first time and **F** for the second.
- Submit a zip file including a **report** file and a **source code** for your program to iCampus
  - For C++, submit both a source code and **Makefile** to produce StudentID.out (ex: 2018001.out)
  - For Python, submit a source code named StudentID.py (ex: 2018001.py)
  - The report file must be the **PDF** format and include as following;
    - 1) Describe your development environments in detail:  
(version of operating system, programming language, compilers/interpreter version, compiler option...)
    - 2) Present how to design your assignment such as data structures and algorithms.
    - 3) Explain how to test your program by showing the screen capture.

## 6. Scoring criteria

- Total 100 points
  - 20 points: Get source and destination file names from keyboard
  - 20 points: Allow to copy multiple files concurrently
  - 20 points: Append every copy completion information in 'log.txt'
  - 20 points: The new (copied) files must be valid. (e.g. possible to play videos)
  - 10 points: Termination when entering 'exit'
  - 10 points: The report file should describe your development environment and how to run your programs.

## 7. Q&A

- Leave your questions on the google sheet