

# [Final Project]

다음 과제를 12월 06일 23:59 까지 linux.mme.dongguk.edu 서버에서 gm\_turnin 프로그램을 통해서 제출하세요.

\* 다음의 조건을 만족하는 TCP 통신이 가능한 Client 프로그램을 구현하고자 한다. Client 프로그램과 Server 프로그램의 조건은 아래와 같다.

## 1. 서버

- linux.mme.dongguk.edu 서버내에서 port 번호 45000 으로 서비스한다. (외부에서는 접근할 수 없으므로 linux.mme.dongguk.edu 서버에 접속하여 구현한 Client 프로그램을 실행한다.)
- 서버 오픈 시간은 추후 공지할 예정이며, 서버 오픈 마감은 12월 06일 23:59 까지이다.
- 서버 오픈 시간부터 서버 마감 시간까지 4 개의 SLOT 으로 분류하여 전체 SLOT 기간내 최소 한번이상 서버에게 메시지를 전달하면 된다. SLOT 별 마감 시간은 다음과 같다.
  - \* SLOT 1: 12월 04일 20시 00분까지
  - \* SLOT 2: 12월 05일 20시 00분까지
  - \* SLOT 3: 12월 06일 10시 00분까지
  - \* SLOT 4: 제출 마감시간까지
- \*\* 서버 오픈 시간, 서버 마감시간, 각 SLOT 별 마감시간은 linux 서버의 시간을 기준으로 처리한다.
- 서버는 접속한 사용자(클라이언트)간 채팅, 파일 업로드 및 파일 다운로드 서비스를 제공한다.
- 서버는 사용자(클라이언트)가 실행한 클라이언트 프로그램 파일을 업로드 및 다운로드할 수 있도록 구현된다.
- 서버 프로그램은 C 언어로 구현된다.
- 서비스 기간동안 서버가 전달받은 모든 메시지는 서버에 저장된다.
- 서비스 기간동안 DDOS 공격 형태의 loop 문 사용 등 서버 프로그램에 문제를 일으키는 프로그램이 있을 경우 감점함

## 2. 클라이언트

- 메시지의 형태와 예시는 다음과 같다.
    - 메시지 형태: [FLAG]|[\$USER]|[DATA]
    - 메시지 전달 예: FLAG\_CHAT|2020123456|Hello WorldWn
  - 서버로 전송하는 모든 메시지 길이는 256 Bytes 로 제한한다.
    - (FLAG\_TYPE 문자열 길이 + '!' 문자의 길이 x 2 + \$USER 문자열의 길이 + DATA 문자열의 길이)
    - \* FLAG\_TYPE 은 FLAG\_CHAT, FLAG\_UPLOAD, ..., FLAG\_EXIT 로 2-2)의 A~F 에 해당한다.
  - 서버로 전송하는 메시지에서 DATA 영역의 길이는 200 Bytes 로 제한한다.
  - 서버로 보내는 메시지는 다음과 같이 전송하며 각 변수를 구별하기 위해 '!' 문자를 사용한다.
    - ('!' 문자는 Shift 키 + W 키를 입력하여 나오는 특수문자이다.)
- 1) \$USER: 프로그램 실행할 때 입력한 옵션으로 프로그램 내부에서 다음과 같은 구문으로 argument 값을 변수에 저장할 수 있다.('s 학번' 또는 'ss 학번'을 값으로 가지며 linux.mme.dongguk.edu 접속한 이름)

> ./2020123456 → 프로그램 실행 예

```
int main(int argc, char* argv[]){
    char num[12];
    ... 생략 ...
    strcpy(num, argv[0]); → 프로그램 실행시 파일이름이 argv[0]에 값 저장
    ... 생략 ...
}
```

- 2) FLAG: 메시지 유형(메시지 구분을 위한 플래그 0x는 16 진수)
  - A. FLAG\_CHAT: 채팅을 위한 플래그
  - B. FLAG\_UPLOAD: 파일 업로드를 위한 플래그
  - C. FLAG\_DOWNLOAD: 파일 다운로드를 위한 플래그
  - D. FLAG\_LIST\_FILE: 다운로드할 수 있는 파일 리스트를 요청하기 위한 플래그
  - E. FLAG\_LIST\_USER: 접속중인 사용자 리스트를 요청하기 위한 플래그
  - F. FLAG\_EXIT: 접속 종료를 서버에게 알리는 플래그

3) DATA: 메시지 유형별 전달해야 하는 데이터는 별첨을 반드시 참고하여 다음 사항을 만족해야 한다.

- 메시지 tokenizer 는 strtok() 함수를 사용한다.
- 파일 업로드 및 다운로드 기능구현에서 **바이너리 데이터**를 제어할 때 다음과 같은 함수를 사용한다.
  - open(파일이름, O\_RDONLY): 읽기전용 열기(업로드)
  - open(파일이름, O\_WRONLY | O\_CREAT): 쓰기전용 열기(다운로드)
  - read() 또는 write() 메서드: 파일 또는 소켓에 파일 읽기 또는 쓰기
- 다음과 같이 커맨드를 입력하여 클라이언트 프로그램을 실행한다.
  - gcc client.c -o 2020123456 → 소스코드를 컴파일한 실행 파일의 이름은 학번으로 작성해야 한다.
  - ./2020123456 → 클라이언트 프로그램 실행
- 구현한 클라이언트가 다음 기능을 수행하는지 확인한다.
  - 1) 채팅 메시지를 서버에게 전달하고 서버로부터 전달받을 수 있다. (채팅)
  - 2) 서버에게 파일을 바이트데이터로 전달하여 서버가 전달받은 메시지로 파일 쓸 수 있도록 한다. (업로드)
  - 3) 서버로부터 파일을 바이트 데이터 형태로 메시지를 전달받아 파일 쓰기를 수행한다. (다운로드)
  - 4) 업로드한 파일 목록을 확인할 수 있다. (다운로드 가능한 파일 리스트 확인)
  - 5) 서버에 접속중인 사용자 목록을 확인할 수 있다. (서버 접속 클라이언트 리스트 확인)
  - 6) 서버에게 종료할 것을 알리고 프로그램을 종료한다. (종료)
- 구현한 클라이언트의 소스코드는 **gm\_turnin** 명령어를 통해 제출 기간내에 반드시 submission 하여야 한다.

### 3. 성적

- 제출한 소스코드 확인 및 실행 후 기준에 따른 항목별 점수 할당

#### [참고]

- \* turnin 프로그램으로 제출 후 hw.pdf 파일에 작성한 소스코드가 출력되었는지 확인하세요. 출력이 되어야 정상 제출입니다. 출력 확인은 제출자의 책임입니다.
- \* 프로그램 작성은 서버에서 vi 에디터를 이용하여 gcc 또는 g++ 로 컴파일 하기 바랍니다.
- \* 제출 마감 시간 이후, 추가 제출을 받지 않습니다.

Message Type	Client → Server	Server → Client
FLAG_CHAT	FLAG_CHAT 2020123456 Hello World#n	FLAG_CHAT 2020123456 Hello World#n  *Server로부터 받은 메시지의 데이터 영역(Hello World#n)을 Client 화면에 출력 *Server는 Client로 받은 FLAG_CHAT 유형의 메시지를 브로드캐스트함
FLAG_UPLOAD	FLAG_UPLOAD 2020123456 %PDF-1.4^M%âãö^M FLAG_UPLOAD 2020123456 12 0 obj^M ... (중략) ... FLAG_UPLOAD 2020123456 EOF  *Client는 정해진 PDF파일을 open() 함수를 호출하여 200byte씩 읽은 후 메시지의 데이터 영역에 붙여서 Server에 전달 *PDF 파일을 다 읽은 후 서버에게 "EOF" 문자열 데이터를 포함한 메시지를 전달 하여 파일 업로드가 종료됨을 알려줌	FLAG_UPLOAD 2020123456 [UPLOAD] Done.  *Client로부터 EOF 문자열 데이터를 받은 Server는 클라이언트에게 위 메시지를 전달함 *Client는 Server로부터 위 메시지를 받으면 화면에 출력함
FLAG_DOWNLOAD	FLAG_DOWNLOAD 2020123456 20201123_1.pdf  *Client는 데이터영역에 다운로드자 하는 파일명을 담아 메시지로 전달함 *파일명은 FLAG_LIST_FILE 메시지를 통해 미리 알고 있어야함 *결론된 파일명인 경우 서버는 FLAG_DOWNLOAD의 Server→Client 항목 2)번처럼 메시지를 전달함	1) FLAG_DOWNLOAD 2020123456 %PDF-1.4^M%âãö^M FLAG_DOWNLOAD 2020123456 12 0 obj^M ... (중략) ... FLAG_DOWNLOAD 2020123456 EOF FLAG_DOWNLOAD 2020123456 [Download] Done.  2) FLAG_DOWNLOAD 2020123456 [ERR] Not found file.  *Server는 다운로드 요청받은 PDF파일을 찾아 존재할 경우 1)번처럼 메시지를 Client에게 전달함 *만약, PDF파일이 서버에 없을 경우 2)번의 메시지를 Client에게 전달함 *1)과 2)는 메시지가 아니며 메시지 유형을 구분하기 위한 표시임
FLAG_LIST_USER	FLAG_LIST_USER 2020123456 NULL	FLAG_LIST_USER 2020123456 1. 2020127022#n2. 2020127023#n3. 2020127024#n  *Server로부터 받은 메시지의 데이터 영역 값을 FLAG_CHAT 메시지 유형과 같이 화면에 출력함
FLAG_LIST_FILE	FLAG_LIST_FILE 2020123456 NULL	FLAG_LIST_FILE 2020123456 1. 20201123_1.pdf#n2. 202011223_2.pdf#n3. 20201123_3.pdf#n  *Server로부터 받은 메시지의 데이터 영역 값을 FLAG_CHAT 메시지 유형과 같이 화면에 출력함
FLAG_EXIT	FLAG_EXIT 2020123456 EXIT	FLAG_EXIT 2020123456 BYE  *Server로부터 받은 메시지의 데이터 영역 값이 "BYE"이면 Client 프로그램 종료함