[Final Project]

다음 과제를 12 월 01 일 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.)

   ** 테스트를 위해 직접 구현하는 서버의 경우 45000 값이외의 포트 번호를 사용할 것

   ** 과제용 서버가 사용하는 port 번호 45000 값을 사용할 경우 감점함

   ** well-known port 및 45000 값 이외의 포트 번호로 테스트할 것

   - 서버 오픈 시간은 추후 공지할 예정이며, 서버 오픈 마감은 12 월 01 일 23:59 까지이다.
   - 서버 오픈 시간부터 서버 마감 시간까지 3 개의 SLOT 으로 분류하여 전체 SLOT 기간내 최소 한번이상 서버에게 메시지를 전달하면 된다. SLOT 별 마감 시간은 다음과 같다.
     * SLOT 1: 11 월 24 일 20 시 00 분까지
     * SLOT 2: 11 월 29 일 20 시 00 분까지
     * SLOT 3: 제출 마감시간까지

     ** 서버 오픈 시간, 서버 마감시간, 각 SLOT 별 마감시간은 linux 서버의 시간을 기준으로 처리함

     ** SLOT 1 에 메시지를 전달한 학생은 SLOT 2~3 에 해당하는 기간에 제출 할 필요 없음

   - 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.
   - 소켓을 연결하거나 메시지를 받을 때 쓰레드를 구현하여야 함

   - 서비스 기간동안 서버가 전달받은 모든 메시지는 서버에 저장됨
   - 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: [$USER]|[MSG_TYPE]|[DATA_LEN]|[MSG_END]|[DATA]
  메시지 전달 예: $USER|0x10|5|0x00|Hello

- 서버로 전송하는 모든 메시지 길이는 256 Bytes 로 제한한다.
  1) 길이 계산
  [$USER 문자열의 길이] +
  [MSG_TYPE 문자열 길이] +
  [DATA_LEN 정수 값] +
  ['|' 문자의 길이 x 4] +
  [DATA 문자열의 길이]

- 서버로 보내는 메시지는 다음과 같이 전송하며 각 변수를 구별하기 위해 '| ' 문자를 사용한다.
  ('|' 문자는 Shift 키 + ₩ 키를 입력하여 나오는 특수문자이다.)

  1) $USER: 프로그램 실행할 때 입력한 옵션으로 프로그램 내부에서 다음과 같은 구문으로 argument 값을 변수에 저장할 수 있다.(linux.mme.dongguk.edu 접속한 계정 이름)

```
〉./client $USER → 프로그램 실행 예
int main(int argc, char* argv[]){
        char num[12];
        … 생략 …
        strcpy(num, argv[1]); → 프로그램 실행시 $USER 변수 값이 argv[1]에 값 저장
        … 생략 …
}
```

  2) 메시지 유형별 전달해야 하는 MSG_TYPE, DATA_LEN, DATA 는 별첨을 반드시 참고할 것!

- 메시지 tokenizer 는 strtok() 함수를 사용한다.

- 파일 업로드 및 다운로드 기능구현에서 바이너리 데이터를 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

- 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) 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

[별첨 1] 메시지 형태 설명

메시지 형태 구분	값	설명
USER_ID	$USER	클라이언트 프로그램을 실행할 때 argv[1] 값에 대응되는 이름 (리눅스 계정 이름) 예) 다음과 같이 프로그램을 실행할 때 argv[1] 값으로 assign된 문자열을 사용함 ./clinet $USER 리눅스 접속 계정이 2020127022일 경우 argv[1] 값은 2020127022이여야 함
MSG_TYPE	0x10	채팅 메시지 전송 요청 (Client -> Server)
	0x11	채팅 메시지 전송 요청에 대한 응답(Server -> Client) • Client가 보낸 메시지를 브로드캐스트함 • 다른 클라이언트가 보낸 채팅 메시지를 구현한 클라이언트가 수신할 수 있는 type임
	0x20	파일 업로드 요청 (Client -> Server)
	0x21	파일 업로드 요청에 대한 응답 (Server -> Client) • 서버가 보내는 메시지의 DATA 값은 다음과 같음 0x00 : 파일 업로드를 허가함 (클라이언트는 0x22, 0x23 메시지 타입으로 서버에게 메시지 전송을 해야함) 0x01 : 파일 업로드를 허가하지 않음 (클라이언트는 파일 데이터 전송을 하지 않음)
	0x22	업로드할 데이터 전송 (Client -> Server) • MSG_END 값은 다음과 같이 구분함 0x00 : 서버에게 보낼 파일의 데이터가 마지막 메시지임을 의미함 (파일 데이터를 마지막까지 전달함) 0x01 : 서버에게 보낼 파일의 데이터가 마지막이 아님 (파일 데이터를 전달 중임)
	0x23	파일 업로드 결과에 대한 응답 (Server->Client)
	0x30	파일 다운로드 요청 (Client -> Server) • 클라이언트가 전달하는 DATA 값은 서버에 업로드된 파일 이름 (서버에 업로드된 파일 이름은 0x40, 0x41 메시지를 통해 확인 가능함)
	0x31	파일 다운로드 요청에 대한 응답 (Server -> Client) • 서버가 보내는 메시지의 DATA 값은 다음과 같음 0x00 : 파일 다운로드를 허가함 (서버는 0x22, 0x23 메시지 타입으로 클라이언트에게 메시지 전송을 함) 0x01 : 파일 다운로드를 허가하지 않음 (서버는 파일 데이터 전송을 하지 않음, 클라이언트가 요청한 파일명이 서버에 저장되어 있지 않은 경우가 대부분임)
	0x32	다운로드할 데이터 전송 (Server -> Client) • MSG_END 값은 다음과 같이 구분함 0x00 : 서버로부터 전달받은 메시지의 DATA가 파일의 마지막 메시지임을 의미함 (파일 데이터를 마지막까지 전달받음) 0x01 : 서버로부터 전달받은 메시지의 DATA가 파일의 마지막이 아님 (파일 데이터를 전달 중임)
	0x33	파일 다운로드 결과에 대한 응답 (Client -> Server)
	0x40	업로드된 파일 리스트 요청 (Client -> Server)
	0x41	업로드된 파일 리스트 요청에 대한 응답 (Server -> Client) • MSG_END 값은 다음과 같이 구분함 0x00 : 서버로부터 전달받은 메시지의 DATA가 마지막 메시지임을 의미함 (업로드된 파일 리스트의 정보를 마지막까지 전달받음) 0x01 : 서버로부터 전달받은 메시지의 DATA가 끝이 아님을 의미함(업로드된 파일 리스트의 정보 일부만 전달 받음)
	0x50	접속 종료 메시지 요청 (Clinet -> Server) • 서버에게 접속 종료 메시지를 요청하며 DATA값은 EXIT로 전달함
	0x51	접속 종료 메시지에 대한 응답 (Server -> Client) • 서버로부터 받은 DATA 값이 0x00이면 클라이언트를 종료함
DATA_LEN	int형 정수	Type [DATA]의 길이에 해당하는 byte 수
MSG_END	0x00	분할된 메시지의 마지막 메시지인지 확인하는 여부 메시지의 데이터가 분할된 경우 마지막 데이터를 의미함
	0x01	분할된 메시지의 마지막 메시지인지 확인하는 여부 메시지의 데이터가 분할된 경우 데이터를 받고 있는 중임
DATA	byte 처리된 문자열	메시지 타입에 해당하는 데이터 문자열

[별첨 2] 메시지 예시

MSG_TYPE	DATA_LEN	MSG_END	DATA	Client -> Server	Server -> Client
0x10	5	0x00	Hello	$USER|0x10|5|0x00|Hello	
0x11	5	0x00	Hello		$USER|0x10|5|0x00|Hello
0x20	8	0x00	test.txt	$USER|0x20|8|0x00|test.txt	
0x21	1	0x00	0x00		$USER|0x21|1|0x00|0x00
0x22	가변	0x01	업로드데이터일부1	$USER|0x22|가변|0x01|데이터일부1	
0x22	가변	0x00	업로드데이터마지막	$USER|0x22|가변|0x00|업로드데이터마지막	
0x23	1	0x00	0x00		$USER|0x23|1|0x00|0x00
0x30	8	0x00	test.txt	$USER|0x30|8|0x00|test.txt	
0x31	1	0x00	0x00		$USER|0x31|1|0x00|0x00
0x32	가변	0x01	다운로드데이터일부1		$USER|0x32|가변|0x01|다운로드데이터일부1
0x32	가변	0x00	다운로드데이터마지막		$USER|0x32|가변|0x00|다운로드데이터마지막
0x33	1	0x00	0x00	$USER|0x33|1|0x00|0x00	
0x40	1	0x00	0x00	$USER|0x40|1|0x00|0x00	
0x41	가변	0x01	리스트문자열일부1		$USER|0x01|가변|0x01|리스트문자열일부1
0x41	가변	0x00	리스트문자열마지막		$USER|0x01|가변|0x00|리스트문자열마지막
0x50	4	0x00	EXIT	$USER|0x50|1|0x00|EXIT	
0x51	1	0x00	0x00		$USER|0x51|1|0x00|0x00

[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.