

네트워크 프로그래밍 과제 #09

1. 멀티 쓰레드를 이용한 OTT(Over-The-Top media service)서비스 구현 (총 15점)

- 제출파일: ott_server.c ott_client.c

- 서버와 클라이언트에서 쓰레드를 생성하지 않고 구현하는 경우 0점 처리함
- 클라이언트의 가입자 유형에 따라 서버는 전송 버퍼의 크기를 달리하여 hw09.mp4 파일을 전송함
- 동영상에서 전송한 파일과 제공한 파일의 크기가 다르기 때문에 아래 결과와 다를 수 있음
- 문제에서 제시한 조건과 맞지 않을 경우, 각 항목당 -2점 감점함

구조체 내용

<pre>// Buffer size #define BASIC_BUF 10 #define STANDARD_BUF 100 #define PREMIUM_BUF 1000 #define MAX_SIZE PREMIUM_BUF typedef struct { int command; int type; char buf[MAX_SIZE]; int len; } PACKET;</pre>	<p>int command: 동작 명령</p> <p>0: 파일 요청(FILE_REQ): client -> server</p> <p>1: 파일 전송 중(FILE_SENDING): server -> client</p> <p>2: 파일 마지막(FILE_END): server -> client</p> <p>3: 파일 마지막 ack(FILE_END_ACK): client -> server</p> <p>int type: 가입자 유형</p> <p>1: Basic 가입자 (전송 버퍼 크기: 10 bytes)</p> <p>2: Standard 가입자 (전송 버퍼 크기: 100 bytes)</p> <p>3: Premium 가입자 (전송 버퍼 크기: 1000 bytes)</p> <p>int len: 파일에서 실제 읽고 전송한 바이트 수</p> <p>MAX_SIZE: 가입자 유형에 따라 최대 전송 버퍼의 크기는 달라짐</p>
---	---

■ 서버 기능 (7점)

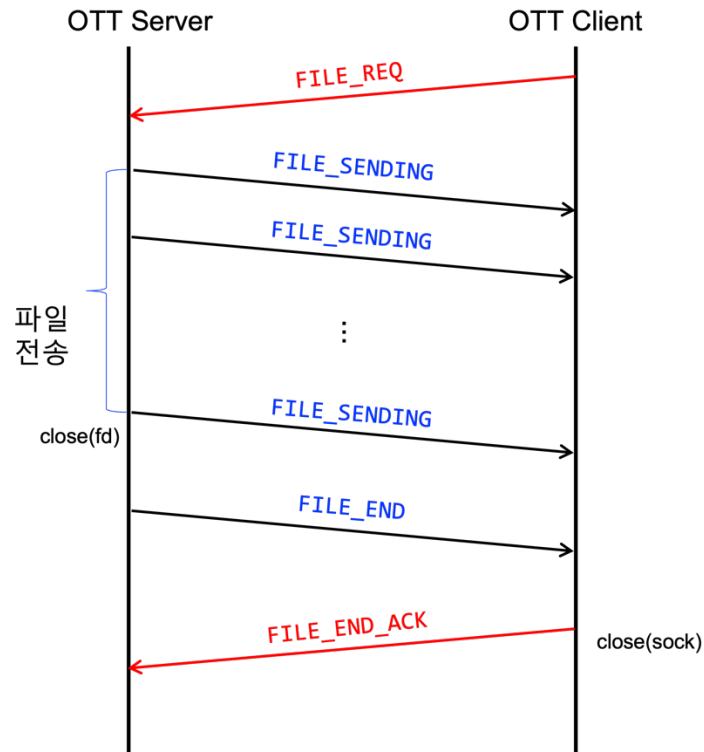
- ✓ 서버는 다중 클라이언트들의 파일 전송 요청에 대해 제공된 hw09.mp4 파일을 가입자 유형에 맞게 버퍼 크기를 변경하여 전송함: (1점)
 - 동기화 기능 구현은 옵션 사항임: 파일 read() 및 파일 전송 부분
- ✓ 메시지 프로토콜에 맞게 파일 전송 기능 구현: (4점)
 - FILE_SENDING, FILE_END 메시지 클라이언트에 전송
 - FILE_END_ACK 메시지 수신시 클라이언트 제거
- ✓ 파일 전송이 종료되고 클라이언트로 부터 FILE_END_ACK 메시지를 수신한 다음 전체 파일 전송 바이트 수 출력 및 가입자 유형 출력: (2점)

■ 클라이언트 기능 (8점)

- ✓ 가입자 유형 지정 (1. Basic, 2. Standard, 3. Premium) 및 Download 메뉴 및 Main menu로 돌아가기 메뉴 기능 구현: (1점)
- ✓ Receive 기능 쓰레드로 구현 (send 쓰레드 구현은 옵션 사항): (1점)
- ✓ 아래의 메시지 프로토콜에 맞추어 전송해야 됨: (4점)
 - 파일 요청(FILE_REQ), 파일의 마지막 수신(FILE_END_ACK) 전송
 - 서버의 FILE_SENDING 메시지를 수신할 때마다 화면에 "."을 출력함
- ✓ 요청한 파일에 대한 파일 다운로드 시간 화면 출력 (1점)

- clock_gettime()함수 사용: 강의교재 15장 참조
- 가입자 유형에 따라 다운로드 시간은 차이가 발생함
- ✓ 서버의 전송 바이트 수와 클라이언트의 수신 바이트 수는 동일해야 됨 (1점)

■ 파일 전송 프로토콜



■ 실행 방법 (제공된 동영상 참조)

- ✓ 각 가입자 유형을 최소 1개씩 선택하고 Download 메뉴를 선택함
- ✓ Basic, Standard, Premium 가입자 각 1개씩 선택

■ 서버 실행 화면

```

$ ./ott_server 9190
-----
K-OTT Service Server
-----
Connected client IP: 127.0.0.1 , clnt_sock=4
Connected client IP: 127.0.0.1 , clnt_sock=5
Connected client IP: 127.0.0.1 , clnt_sock=7

Total Tx Bytes: 17709659 to Client 7 (Premium) // 전송 완료 후 출력
[Rx] FILE_END_ACK from Client 7 => clnt_sock: 7 closed.

Total Tx Bytes: 17709659 to Client 5 (Standard)
[Rx] FILE_END_ACK from Client 5 => clnt_sock: 5 closed.

Total Tx Bytes: 17709659 to Client 4 (Basic)
[Rx] FILE_END_ACK from Client 4 => clnt_sock: 4 closed.
  
```

■ 클라이언트 메뉴 실행 화면

```

$ ./ott_client 127.0.0.1 9190
-----
K-OTT Service
-----
Choose a subscribe type
-----
1: Basic, 2: Standard, 3: Premium, 4: quit: 1
-----
1. Download, 2: Back to Main menu: 2
-----
K-OTT Service
-----
Choose a subscribe type
-----
1: Basic, 2: Standard, 3: Premium, 4: quit: 4
Exit program

```

■ 클라이언트 #1 실행 (Premium 가입자)

```

$ ./ott_client 127.0.0.1 9190
-----
K-OTT Service
-----
Choose a subscribe type
-----
1: Basic, 2: Standard, 3: Premium, 4: quit: 3
-----
1. Download, 2: Back to Main menu: 1
-----
중간 생략
.....
.....
.....
.....
.....
.....
.....
File Transmission Finished
Total received bytes: 17709659
Downloading time: 160 msec
Client closed

```

클라이언트 #2 실행 (Standard 가입자)

```

$ ./ott_client 127.0.0.1 9190
-----
K-OTT Service
-----
Choose a subscribe type
-----
1: Basic, 2: Standard, 3: Premium, 4: quit: 2
-----
1. Download, 2: Back to Main menu: 1
-----
중간 생략

```

