

# 컴퓨터 네트워크

2018년 4월 13일

컴퓨터공학과

2015061194 김선국

## 1. 동작 설명

### 1) Makefile

```
jimmyui-MacBook-Pro:ClientServer_Example jimmykim$ make
gcc -c server.c
gcc -o server server.o
rm server.o
```

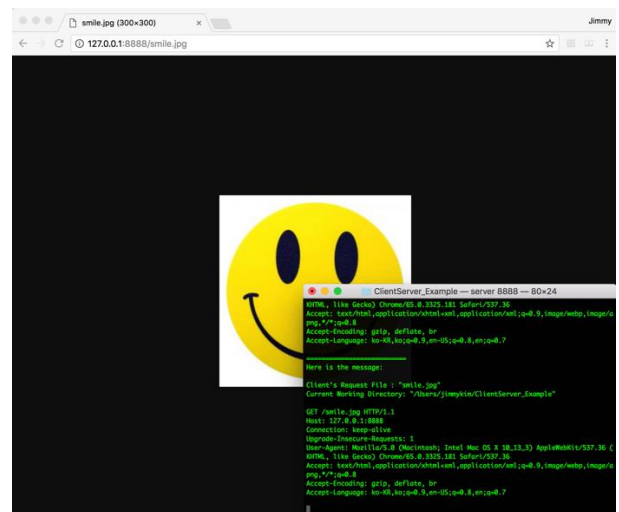
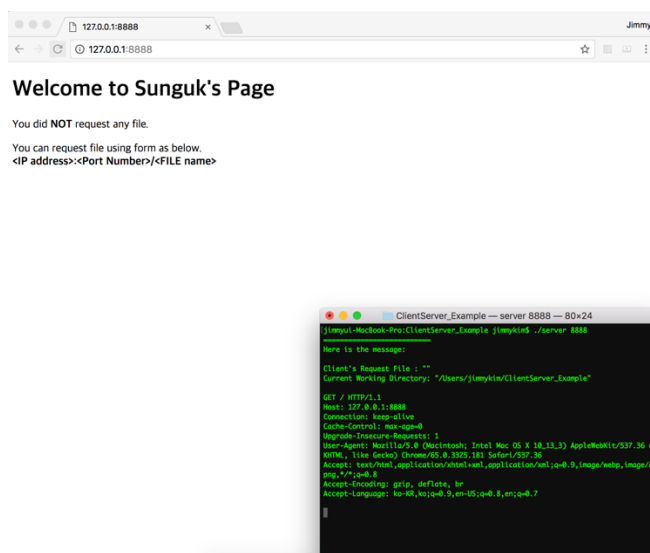
make를 입력하면 object파일이 만들어지고 server라는 실행파일이 생성됩니다. object파일은 자동으로 삭제됩니다.

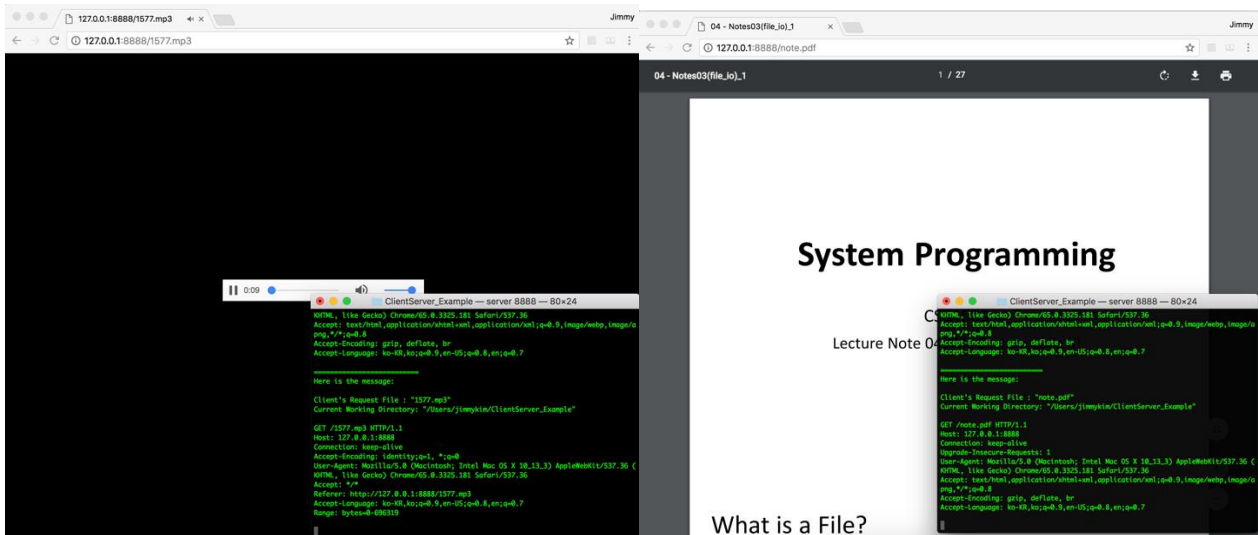
### 2) 실행

```
jimmyui-MacBook-Pro:ClientServer_Example jimmykim$ ./server 8888
```

실행파일과 함께 포트번호를 입력하면 그림과 같이 accept함수에서 차단(blocking)이 되며 client의 요청을 기다립니다.

### 3) 파일불러오기(파일이 있을 시)

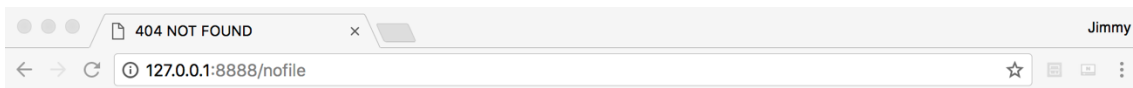




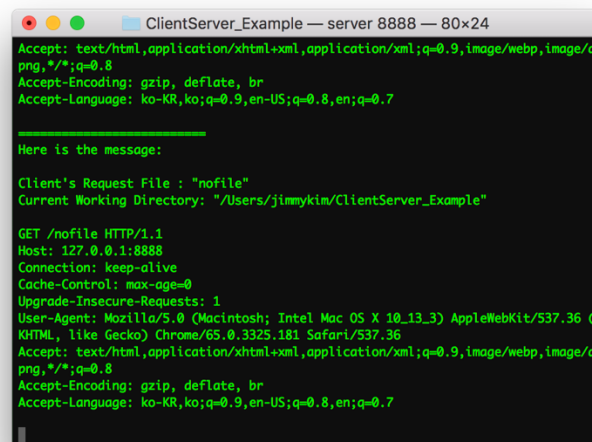
- ip번호와 port번호만 입력했을 시 default 페이지인 index.html 파일을 보내도록 설정하였습니다.(좌측상단)
- jpg확장자를 처리한 모습입니다.(우측상단)
- mp3확장자를 처리한 모습입니다. (좌측하단)
- pdf확장자를 처리한 모습입니다.(우측하단)

터미널(server side)에는 client가 request한 파일이 무엇인지 보여주고 현재 작업하고 있는 경로와 함께 보여줍니다. 그 후 사용자의 header information을 보여줍니다.

#### 4) 파일 불러오기(파일이 없을 시)



**Sorry, your requested file does NOT exist.  
TRY another file.**



폴더안에 없는 파일을 요청했을 시, "HTTP/1.1 404 Not Found\r\nContent-type: text/html\r\n\r\n" 헤더를 넘기고 404.html 파일을 보내도록 설정하였습니다.

## 2. 어려웠던 점

### 1) Header Information이 Client-side(웹 브라우저)에 보이는 현상

기본적인 read와 write함수에 대한 동작을 알아보기 위해 rough코드를 짜서 테스트를 했습니다. 처음 client쪽에서 서버에 접속하면 정확한 값이 들어왔으나 새로고침을 하면 의도치 않게 header-information이 브라우저 상에 보이는 현상이 보였습니다. 여러 경우의 수들을 테스트 해본 결과 보내고자 하는 데이터 사이즈가 정확하지 않으면 쓰레기 값들이 축적되어 넘어가서 생기는 오류로 판단 할 수 있었습니다. 이 경험을 통해 write함수를 사용할 때 기본 1024바이트를 보내되 sndsize라는 변수를 선언하여 마지막 반복 횟수가 되면 파일의 나머지 사이즈만큼 정확하게 보내도록 하였습니다.(149 line)

### 2) AF\_INET과 PF\_INET 차이

교수님께서 socket함수를 설명해주실때 PF\_INET에 대해서 알려주셨는데 sample code에는 AF\_INET을 사용하고 있었습니다. 차이점을 알아보았고, 프로토콜 체계를 설정하는 부분은 PF로 시작하는 "상수"를 사용하고, 주소 체계를 설정하는 부분은 AF로 시작하는 "상수"를 사용하는 것을 알 수 있었습니다. 하지만 PF\_INET과 AF\_INET의 상수 값은 같았고, 사실상 socket함수에서 받는 인자 값의 차이는 없는 것을 알 수 있었습니다.

### 3) Listen() 함수의 정확한 역할:

수업 중 accept함수는 blocking을 하고 listen함수에서는 안한다고 들었지만, 정확히 listen함수는 어떤 동작을 하는지 알지 못했습니다. Listen함수를 호출하면 서버 소켓 상태는 close에서 listen상태로 변경됩니다. 모든 연결 요청은 서버가 미리 만들어 놓은 대기실로 들어가 순서대로 연결 요청이 수락될 때까지 기다릴 수 있는 상태가 됩니다.

### 4) Client-side에서 무한 로딩 문제:

서버를 지속적으로 가동하기 위해 while문을 사용했습니다. 하지만 Client쪽에서 화면이 뜨지않고 계속 로딩하는 모습을 보였습니다. 데이터 송신이 끝나면 소켓을 close()하여 서버와의 통신을 더이상 안하게 하여 무한 로딩 문제를 해결하였습니다.

### 5) favicon.ico 요청:

server 동작 중에 favicon.ico 라는 파일 request가 지속적으로 들어왔습니다. 이 파일은 일부 브라우저가 주소 표시줄 정보 및 즐겨찾기, 책갈피 목록의 표시를 향상하기 위해 사용하는 작은 그래픽 아이콘입니다. 브라우저는 사이트에 맞게 정의된 favicon.ico 파일을 찾으려고 시도하기 위해 요청을 지속적으로 보내는데, 저는 이 요청이 터미널 창에 계속 보이는 것이 거슬려서 continue로 처리하여 Client가 직접 요청한 파일이 아니면 터미널에 뜨지않게 처리하였습니다.

## <참조 문헌>

### **AF\_INET과 PF\_INET 차이:**

<http://mistrabbit.tistory.com/90>

### **Listen()함수의 역할:**

<http://mintnlatte.tistory.com/280>

### **Open()함수와 fopen()함수의 차이:**

<https://m.blog.naver.com/PostView.nhn?blogId=liebeguso&logNo=140037906132&proxyReferer=https%3A%2F%2Fwww.google.com%2F>

**MIME타입에 대한 정보:** Content-type에는 여러가지가 있고 각각을 처리해줘야하기때문에 대표적인 MIME타입들을 참조함

[https://developer.mozilla.org/ko/docs/Web/HTTP/Basics\\_of\\_HTTP/MIME\\_types](https://developer.mozilla.org/ko/docs/Web/HTTP/Basics_of_HTTP/MIME_types)

<https://www.sitepoint.com/mime-types-complete-list/>

### **favicon.ico에 대한 설명:**

[https://www.ibm.com/support/knowledgecenter/ko/SSELE6\\_8.0.1.2/com.ibm.isam.doc\\_8.0.1.2/wrp\\_config/task/tsk\\_handl\\_favicon.html](https://www.ibm.com/support/knowledgecenter/ko/SSELE6_8.0.1.2/com.ibm.isam.doc_8.0.1.2/wrp_config/task/tsk_handl_favicon.html)