



# 2-3

## 금3,4

과목명	시스템프로그래밍
담당교수	김태석 교수님
학과	컴퓨터정보공학부
학년	3학년
학번	2019202009
이름	서여지
제출일	21.05.11 (화)

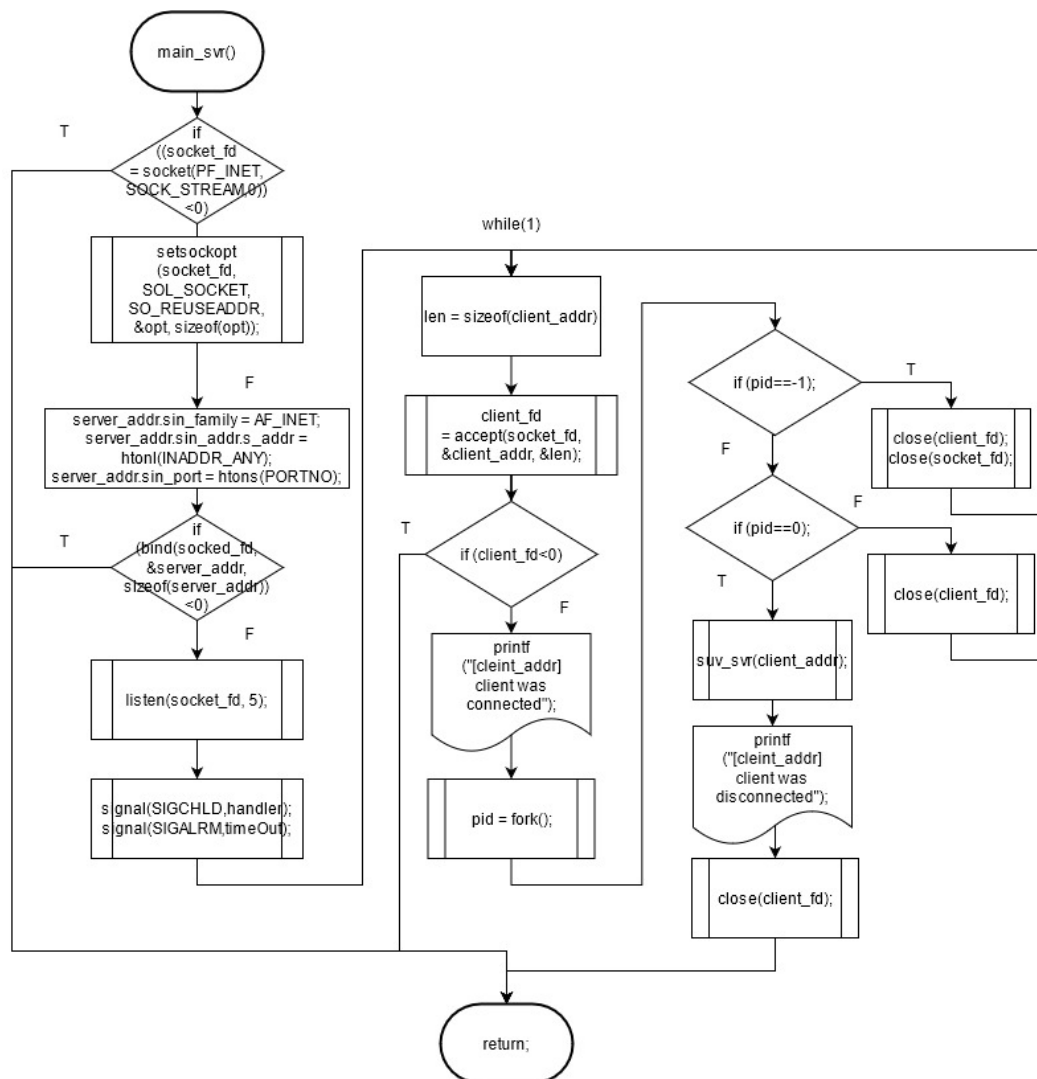
## 1. Introduction //과제 소개 - 5줄 내외(background 제외)

인터넷 브라우저에서 입력한 URL에 대해 cache 파일을 조회하고, 해당 내용을 브라우저에 전달한다. 해당하는 cache 파일이 없는 경우 web server에 client의 request message를 전송한다. web server가 전송한 response message를 인터넷 브라우저에 전달하고, 동시에 cache파일을 생성하여 그 내용을 저장한다. 인터넷이 연결되지 않아 web server와 연결이 되지 않거나 response를 받지 못하는 경우 timeOut message를 브라우저에 전달한다.

## 2. Flow Chart //코드 작성 순서도

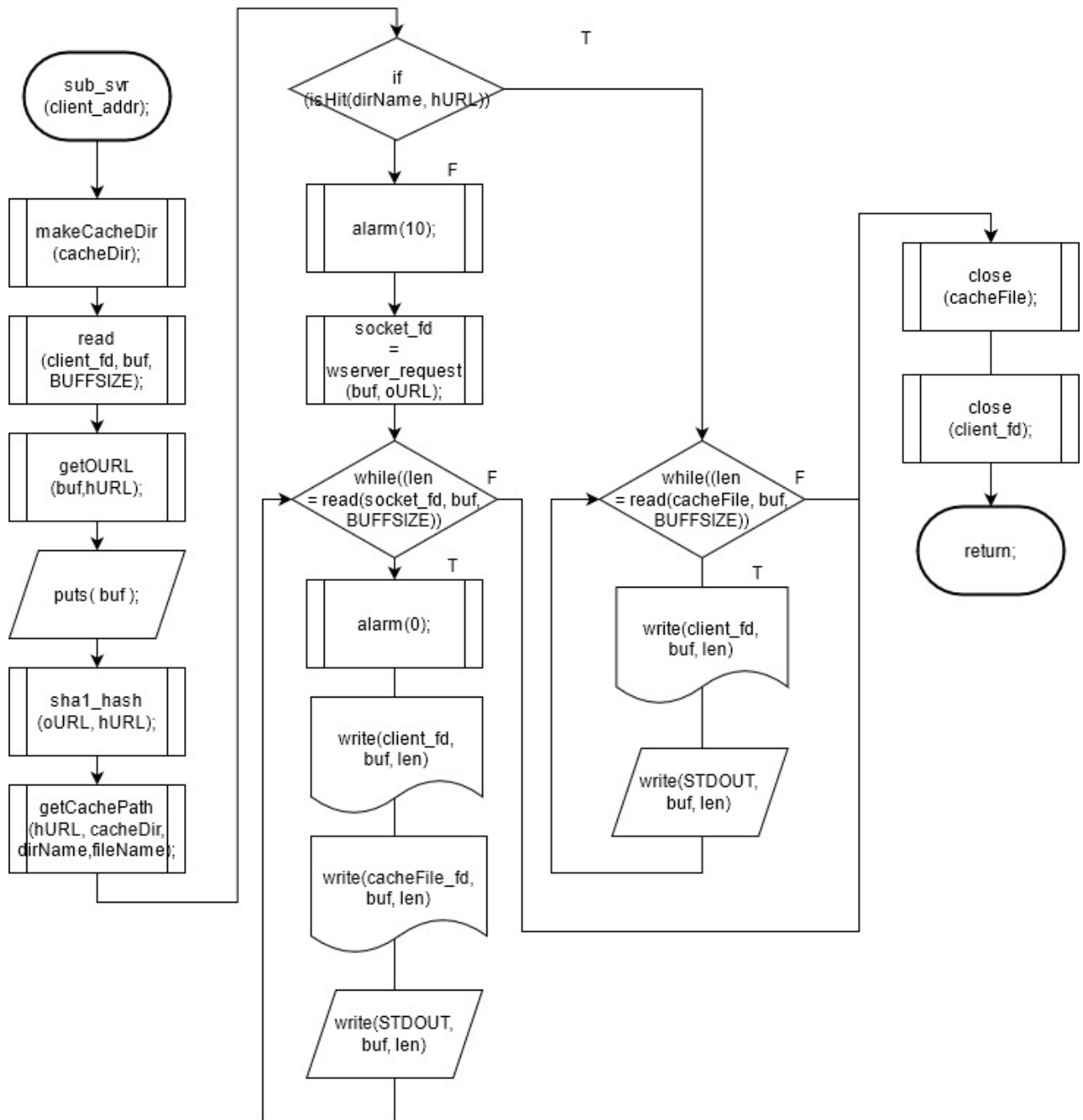
### (1) main\_svr

이전 프로젝트의 내용에서 SIGALRM의 handler를 지정하는 내용이 추가되었다.

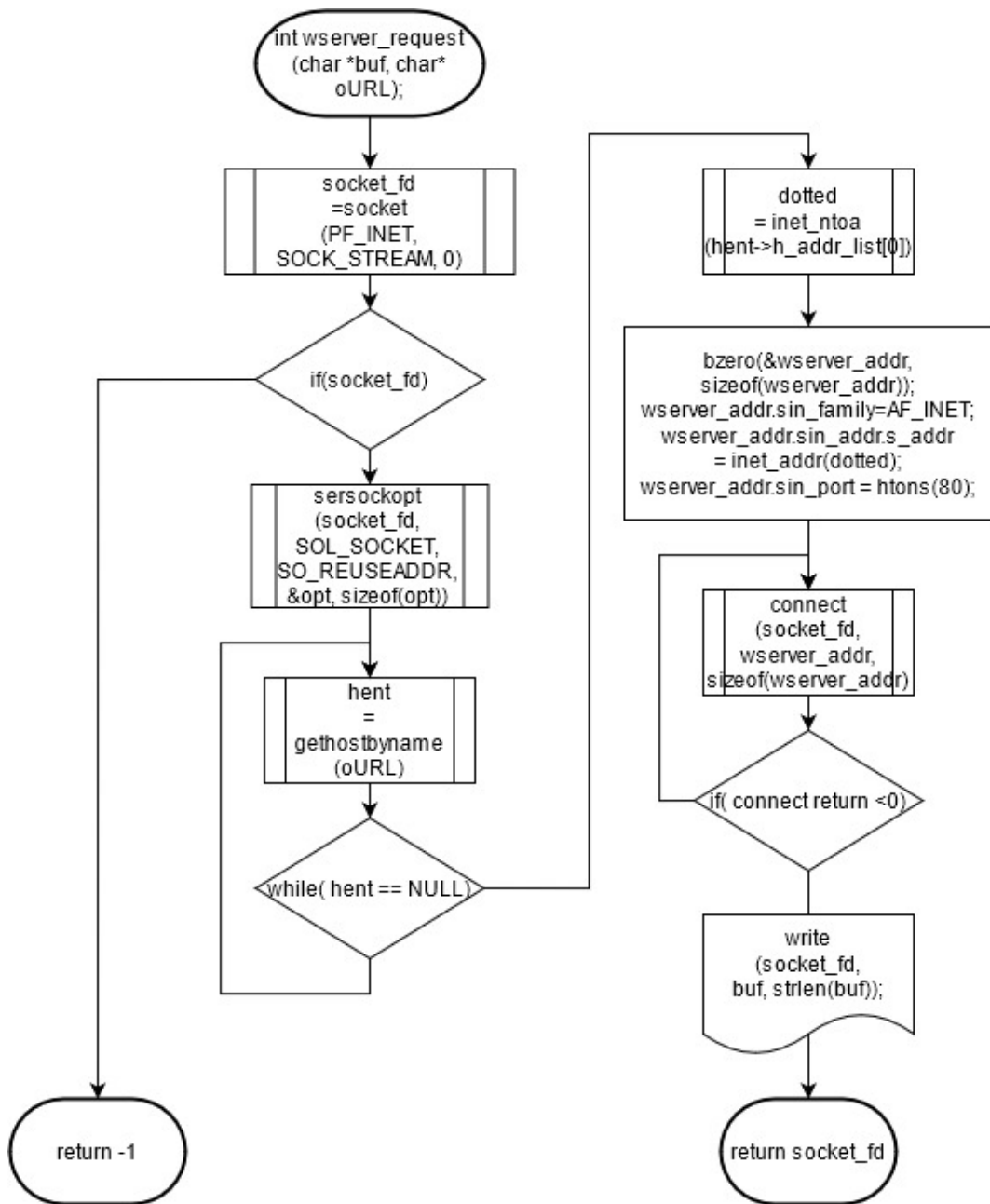


### (2) sub\_svr

client\_fd가 전역변수로 선언되어 인자에서 제외되었고, 상태 출력을 위해 인자로 client\_addr를 전달받도록 변경되었다.



(3) wserver\_request



3. Pseudo code //알고리즘

```

void main_svr(void) { // proxy2-2와 유사하다
    signal(SIGCHLD, handler);
    signal(SIGALRM, timeOut);

    socket(PF_INET, SOCK_STREAM);
    if (socket error ) return; // can't open socket
    setsockopt(socket_fd, SOL_SOCKET, SO_REUSEADDR, &opt, sizeof(opt));
  
```

```

server_addr = 0;
server_addr.sin_family = AF_INET;
server_addr.sin_addr.s_addr = htonl(INADDR_ANY);
server_addr.sin_port = htons(PORTNO);

bind(socket_fd, &server_addr, sizeof(server_addr))
if( bind error ) return;    // can't bind address

listen(socket_fd, 5);

while(true){
    client_addr = 0;
    len = sizeof(client_addr);
    client_fd = accept(socket_fd, &server_addr, &len)
    if(client_fd < 0) return;    // accept err
    write ("[client addr] client was connected/n");
    pid = fork();
    if (fork err) {
        close(client_fd);
        close (socket_fd);
        continue;
    }
    if (sub process) {
        sub_svr(client_addr);
        write("[client addr] client was disconnected\n");
        close(client_fd);
        return;
    }
    close(client_fd);    //main process
}
close(socket_fd);
return;
}

void sub_svr(struct sockaddr_in client_addr) {
    makeCacheDir(cacheDir);

    read(client_fd, buf, BUFFSIZE);
    strcpy(tmp, buf);
    getOURL(tmp, oURL);

    sha1_hash(oURL, hURL);
    getCachePath(hURL, cacheDir, dirName, fileName);

    if(isHIT) {
        cacheFile = open(fileName, "r");
        while((read(cacheFile, buf, BUFFSIZE))>0) {
            write(client_fd, buf, len);
            write(STDOUT_FILENO, buf, len);
        }
        close(cacheFile);
    }
    else {
        alarm(10);
    }
}

```

```

        socket_fd = wserver_requset(buf, oURL);
        cacheFile = open(fileName, "w");
        while((len = read(socket_fd, buf, sizeof(buf)))>0){
            alarm(0);
            write(client_fd, buf, len);
            write(cacheFile, buf, len);
            write(STDOUT_FILENO, buf, len);
            buf = 0;
        }
        close(socket_fd);
        close(cacheFile);
    }
    close(client_fd);
    return;
}

int wserver_requset(char* buf, char* oURL){
    socket_fd = socket(PF_INET, SOCK_STREAM);
    if (socket error ) return;    // can't open socket
    setsockopt(socket_fd, SOL_SOCKET, SO_REUSEADDR, &opt, sizeof(opt));

    hent = gethostbyname(oURL);
    while (gethostbyname error) retry gethostbyname(oURL) after 1 sec
    dotted = inet_ntoa(hent->h_addr_list[0]);

    wserver_addr = 0;
    wserver_addr.sin_family = AF_INET;
    wsrever_addr.sin_addr.s_addr = dotted
    wsrever_addr.sin_port = htons(80);

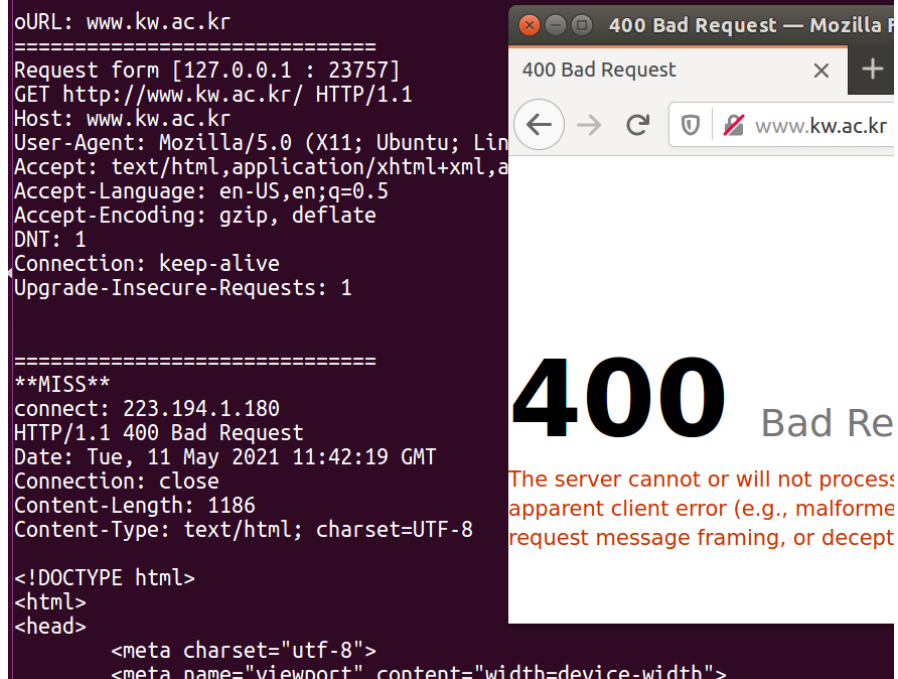
    connect(socket_fd, &wserver_addr, sizeof(wserver_addr));
    while(connect error) retry connect after 1 sec

    write(socket_fd, buf, strlen(buf));
    return socket_fd;
}

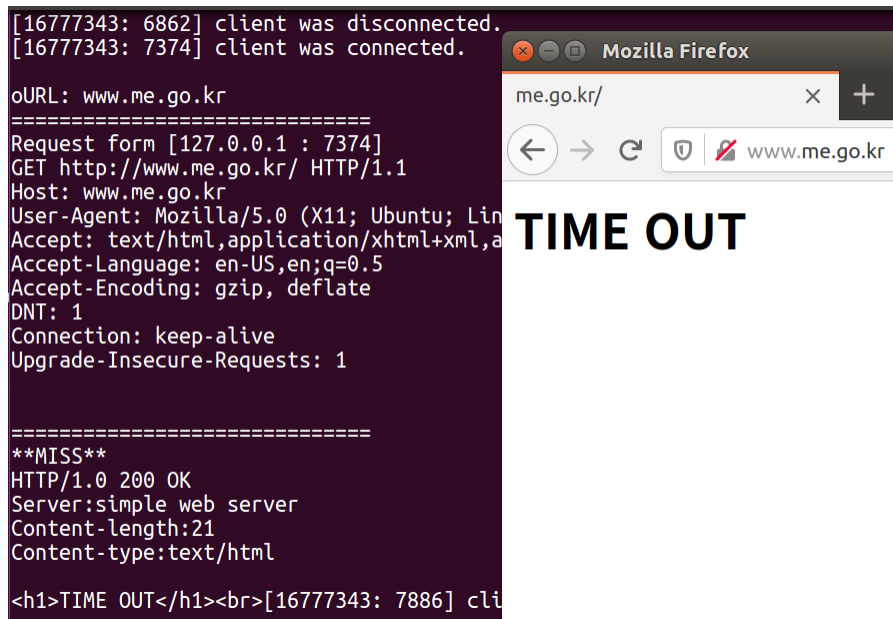
```

#### 4. 결과화면 //수행한 내용을 캡처 및 설명

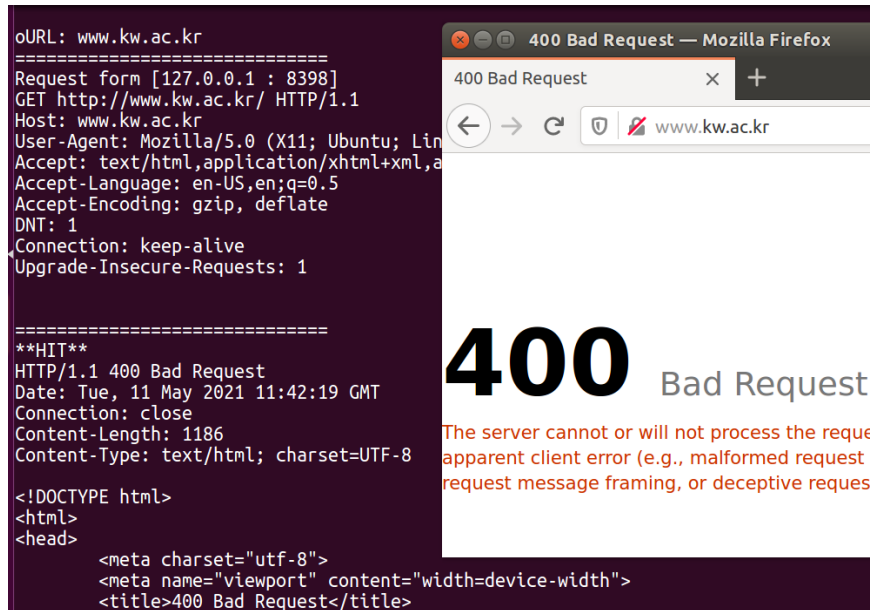
인터넷에 연결된 상태로 firefox 브라우저에 url을 입력하여 웹서버로부터 정보를 받아오는 기능을 확인하고, 인터넷을 해제한 뒤, 또 다른 URL을 입력하여 time out을 확인한다. 계속하여 인터넷이 해제된 상태에서 처음 입력했던 url을 다시 입력하여 cache를 불러오는 기능을 확인하는 것으로 프로그램 동작을 확인하였다. https로 연결되는 것을 피하기 위해 http를 이용하는 환경부 홈페이지 [www.me.go.kr](http://www.me.go.kr) 을 이용하였다.



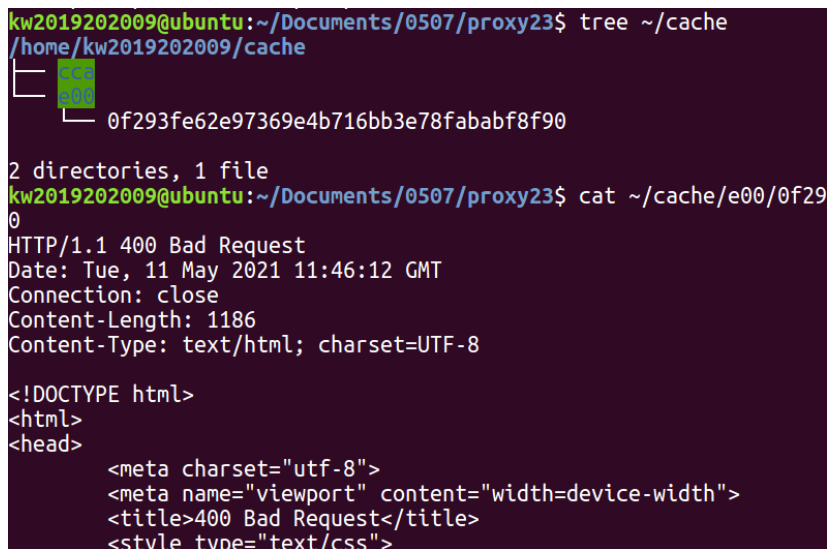
<처음 [www.kw.ac.kr](http://www.kw.ac.kr)을 입력한 결과>



<인터넷 연결을 해제한 뒤, [www.me.go.kr](http://www.me.go.kr)을 입력한 결과>



<연결이 해제된 상태에서 [www.kw.ac.kr](http://www.kw.ac.kr)을 다시 입력한 결과>



<실행 완료 후 cache파일의 내용>

## 5. 결론 및 고찰

작성한 proxy server와 웹서버를 연결하는 내용을 추가로 작성하고, alarm signal을 이용하여 timeout 기능을 구현하는 과제이다. 과제를 진행하며 몇 가지 기능을 구현하는데 어려움을 겪었다. signal함수와 alarm(10)을 이용하여 timeOut함수가 10초 뒤에 실행되도록 작성하였는데, 인터넷을 연결하지 않은 상태에서는 gethostbyname함수가 부적절한 값을 반환하고, connect함수는 socket연결이 바로 실패하기 때문에 10초를 기다리지 않고 즉시 종료되었다. 이 문제를 해결하기 위해 while반복문과 sleep(1)을 이용하여 gethostbyname과 connect를 1초마다 재시도 하도록 설계하였다.