



2-2

금3,4

과목명	시스템프로그래밍
담당교수	김태석 교수님
학과	컴퓨터정보공학부
학년	3학년
학번	2019202009
이름	서여지
제출일	21.05.04 (화)

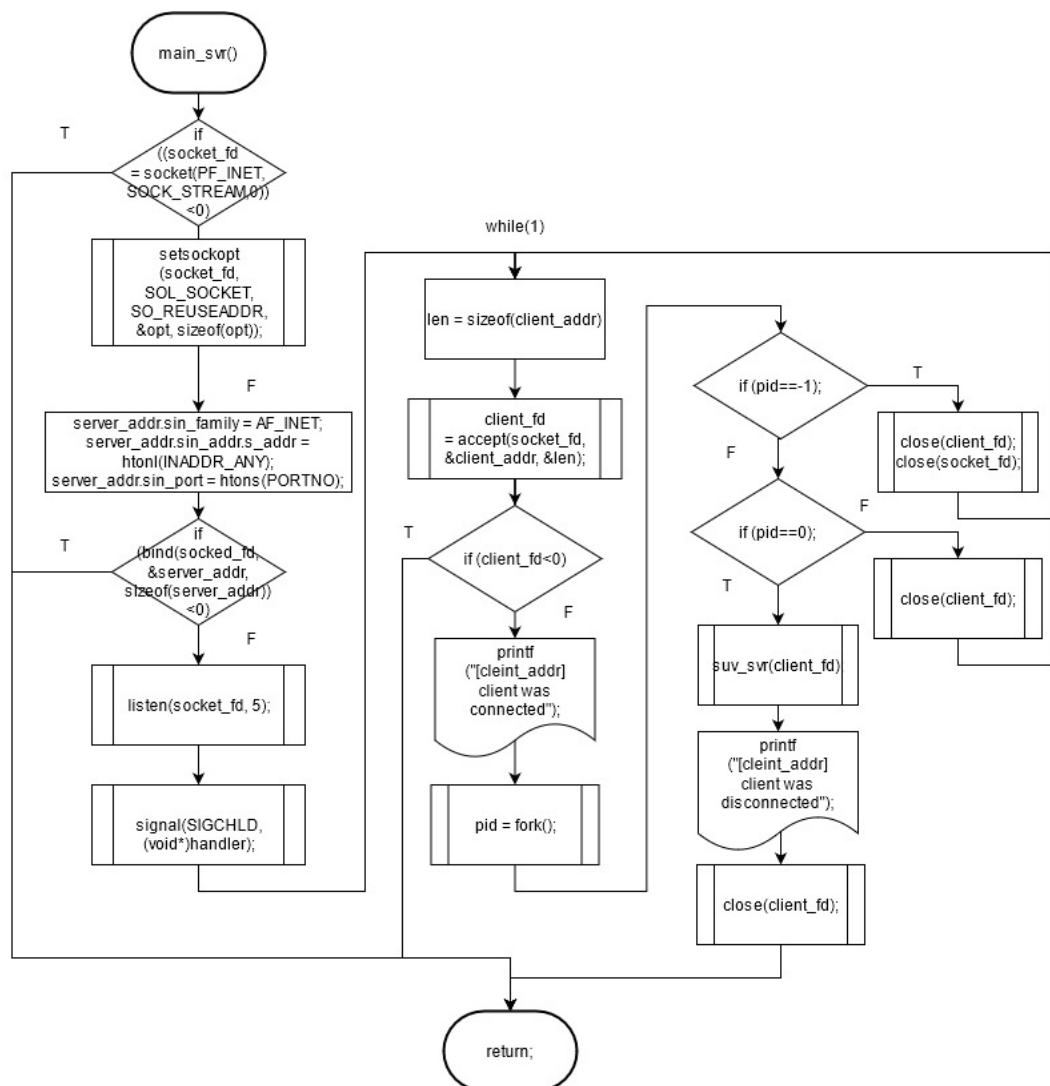
1. Introduction //과제 소개 - 5줄 내외(background 제외)

지금까지 작성한 프로그램 중 사용자에게 stdin을 통해 URL을 입력받는 client를 인터넷 브라우저로 변경한다. 이때 브라우저가 server에 전송하는 것은 HTML request message이고, server는 message에서 URL을 찾아 hash한 이후 cache 파일이 존재하는지 확인한다. 그 결과를 인터넷 브라우저에 HTML response message로 전송한다.

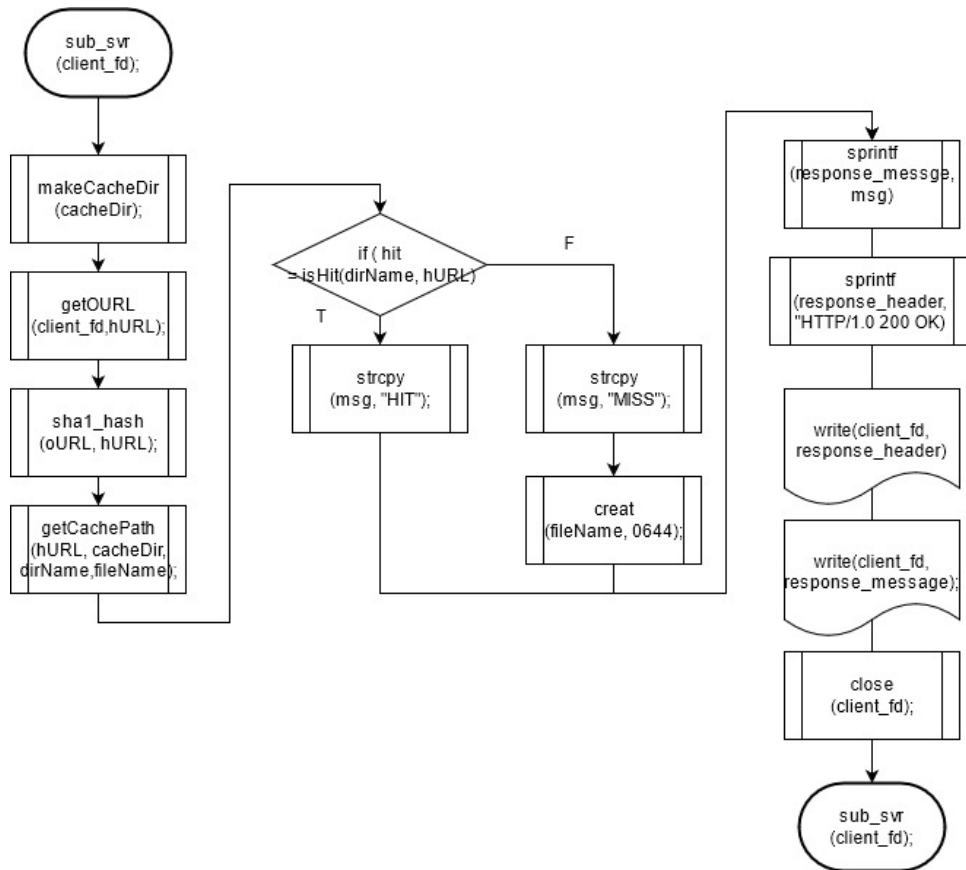
2. Flow Chart //코드 작성 순서도

(1) main_svr

proxy2-1과 유사하지만 setsockopt가 추가되었다.



(2) sub_svr



3. Pseudo code //알고리즘

```

void main_svr(void) { // proxy2-1과 유사하다
    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);
    signal(SIGCHLD, (void*)handler);

    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_fd);
            write("[client addr] client was disconnected\n");
            close(client_Fd);
            return;
        }
        close(client_fd);    //main process
    }
    close(socket_fd);
    return;
}

void sub_svr(int client_fd) {
    makeCacheDir(cacheDir);

    getOURL(client_fd, oURL);
    sha1_hash(oURL, hURL);

    getCachePath(hURL, cacheDir, dirName, fileName);

    if(isHIT) msg="HIT";
    else {
        msg = "MISS";
        creat(fileName, 0644);
    }
    sprintf(response_message, msg);
    sprintf(response_header, "HTTP/1.0 200 OK");

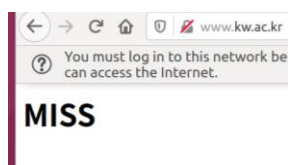
    write(client_fd, response_header);
    write(client_fd, response_message);

    close(client_fd);
    return;
}

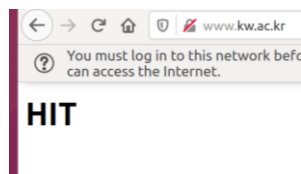
```

4. 결과화면 //수행한 내용을 캡처 및 설명

작성한 프로그램인 proxy_cache를 실행한 뒤, firefox 웹 브라우저에 동일한 URL을 전달하여 결과를 확인하였다.



처음 www.kw.ac.kr에 접속했을 때



F5를 눌러 동일한 페이지를 불러왔을 때

```
kw2019202009@ubuntu:~/Documents/0430/proxy22$ tree ~/cache
~/cache
├── 3ae58926a18c1b6815988948cb3ebd894d702
├── 9a3ee5ecb4b0d3255bfef95601890afd80709
└── 0f293fe62e97369e4b716bb3e78fababf8f90
3 directories, 3 files
```

실행 결과 cache directory의 내용

5. 결론 및 고찰

웹 브라우저가 보내는 HTML request message를 읽어서 cache를 확인할 URL을 찾아 동작하는 프로그램을 작성하였다. 실습을 진행하는데 큰 어려움은 없었으나 URL을 읽어오는 getOURL함수에서 request의 “GET” 이후 http이후 “//”과 디렉토리를 구분하는 “/” 사이의 내용을 URL로 추출하도록 작성하였다. “/favicon.ico”와 같은 request message를 예외처리하기 위함이다. 특별히 제시된 요구사항이 없었기 때문에 cache를 확인하는 동작이 정상적으로 전달되는 것만을 확인하였다. 그러나 이 경우 <https://www.kw.ac.kr/en/index.jsp> 와 <https://www.kw.ac.kr/ko/>이 모두 www.kw.ac.kr로 인식된다. 이것을 적절히 처리하는 방법이 필요할 것이다.

코드를 작성한 이후 하나의 .h파일과 두 개의 .c파일로 나누었는데, 이것의 Makefile을 작성하는 부분에서 문제가 발생하였다. -c 옵션을 이용하여 .o file을 두 개 생성하고, .o file에 대해 다시 한 번 gcc 명령어를 사용하려고 했는데 실패하였다. 이 문제는 작년 데이터구조시간에 사용했던 Makefile을 참고하여 해결하였다. *.c, *.h 를 이용하여 한 번에 모든 파일을 포함하여 컴파일하도록 하였다. 이 방식을 이용하면 같은 디렉토리의 모든 c파일과 h파일을 이용하여 컴파일 하므로 주의해야한다.

6. reference //과제를 수행하면서 참고한 내용을 구체적으로 기록

이형근교수님, 데이터구조실습 강의자료, 광운대학교 컴퓨터정보공학과, 2020