

2024년 1학기 시스템프로그래밍 & 시스템 프로그래밍 실습

Assignment2 - 1

System Software Laboratory
College of Software and Convergence
Kwangwoon Univ.

Assignment2-1 – Basic Socket

- **To do list**

- **Client** → Implement the client module by using socket(), connect() and write()..
 - Convert the user command ("ls", "quit") to FTP command ("NLST", "QUIT ")
 - **Implement all of the "ls" options -a, -l and -al.**
 - Send FTP command to the server and receive the result
 - Display the result of processing command
- **Server** → Implement the server module by using socket(), bind(), listen().....
 - Listen the request of the client
 - Process FTP command and return the result to the client
 - Display Converted command
 - Display IP and port by using inet_ntoa(), ntohs()

Assignment2-1– Work Flow (Client)

- **Connect to server**
 - Server address is 127.0.0.1
- **Convert user command to FTP commands**
 - `ls -al` -> `NLST -al`, `quit`-> `QUIT`
- **Send ftp command to server by using write()**
- **Receive the result from server by using read()**
- **Processing about the result**
 - If the result of “`ls -al`” returned, write it to stdout by using `write()`.
 - If the result of “`quit`” returned, quit client program

Assignment2-1 - Work Flow (Server)

- **To serve connecting client**
 - Define server port number
 - Use listen() and accept()
- **When client connected, display client IP and PORT**
 - To display port : ntohs()
 - To display IP : inet_ntoa()
- **Receive FTP command from client**
 - Use read() function
- **Process FTP command**
- **Send result processed to client**
 - Use write() function

Example code – Client (1/2)

```
void main(int argc, char **argv)
{
    char buff[MAX_BUFF], cmd_buff[MAX_BUFF], rcv_buff[RCV_BUFF];
    int n;
    /* open socket and connect to server */
    for(;;)
    {
        if(conv_cmd(buff, cmd_buff) < 0) /* convert ls (including
options) to NLST (including options) */
        {
            write(STDERR_FILENO, "conv_cmd() error!!\n", sizeof());
            exit(1);
        }
        n = strlen(cmd_buff);
        if(write(sockfd, cmd_buff, n) != n)
        {
            write(STDERR_FILENO, "write() error!!\n", sizeof());
            exit(1);
        }
    }
}
```

Example code – Client (2/2)

```
if((n = read(sockfd,rcv_buff, RCV_BUFF-1)) < 0)
{
    write(STDERR_FILENO,"read() error\n", sizeof());
    exit(1);
}
rcv_buff[n] = '\0';
if(!strcmp(rcv_buff, "QUIT"))
{
    write(STDOUT_FILENO,"Program quit!!\n", sizeof());
    exit(1);
}

process_result(rcv_buff); /*display ls (including options) command
result */

}

}
```

Example code – Server (1/2)

```
int main(int argc, char **argv)
{
    char buff[MAX_BUFF], result_buff[SEND_BUFF]
    int n;
    /* open socket and listen */
    for(;;)
    {
        connfd = accept(listenfd, (SA *) &cliaddr, &clilen);
        if(client_info(&cliaddr) < 0) /* display client ip and port */
            write(STDERR_FILENO, "client_info() err!!\n", sizeof());
        while(1)
        {
            n=read(connfd, buff, MAX_BUFF);
            buff[n] = '\0';
```

Example code – Server (2/2)

```
if(cmd_process(buff, result_buff) < 0)
/* command execute and result */
{
    write(STDERR_FILENO, "cmd_process() err!!\n");
    break;
}
write(connfd, result_buff, strlen(result_buff));
if(!strcmp(result_buff, "QUIT"))
{
    write(STDOUT_FILENO, "QUIT\n", sizeof());
    close(connfd);
    break;
}
}
close(serverfd);
}
```


Construct Function

`conv_cmd(buff, cmd_buff)`

들어온 명령어를 FTP명령어로 변환 뒤 cmd_buff에 저장.

`client_info(&cliaddr)`

Client의 정보를 출력하기 위한 함수.

Client의 주소와 포트 번호를 다음과 같은 형식으로 출력.

```
=====Client info=====
client IP: 127.0.0.1 ← When client connect to Server

client port: 13428 ← Changeable (not fixed)
=====
```

`cmd_process(buff, result_buff)`

변환 된 명령어를 읽고 (buff) 프로세싱을 하고 결과에 저장 (result_buff).

Example Result (Client)

```
$ ./cli 127.0.0.1 1000
```

```
> ls
```

← user input

```
cli
```

```
cli.c
```

```
srv
```

```
srv.c
```

```
> ls -l
```

← user input

```
-rwxr-xr-x 1 splab SSLab 24644 4월 28일 18:01 cli
-rw-rw-r-- 1 splab SSLab      14 4월 28일 17:35 cli.c
-rwxr-xr-x 1 splab SSLab       7 4월 28일 18:01 srv
-rw-rw-r-- 1 splab SSLab      22 4월 28일 18:45 srv.c
```

```
> quit
```

← user input

```
Program quit!!
```

```
$
```

"ls -a", "ls -al" also have been implemented

Example Result (Server)

```
./srv 1000
```

```
=====Client info=====
```

```
client IP: 127.0.0.1 ← When client connect to Server
```

```
client port: 13428 ← Changeable (not fixed)
```

```
=====
```

```
NLST -al
```

```
QUIT
```

```
$
```

Report Requirements

- **Ubuntu 20.04.6 Desktop 64bits 환경에서 채점**
- **Copy 발견 시 0점 처리**
- **보고서 구성**
 - **보고서 표지**
 - 수업 명, 과제 이름, 담당 교수님, 학번, 이름 필히 명시
 - 과제 이름 → Assignment2-1
 - **과제 내용**
 - Introduction
 - 과제 소개 - 4줄 이상(background 제외) 작성
 - Flow chart(4주차 강의자료 appendix 참고)
 - Pseudo code(4주차 강의자료 appendix 참고)
 - 결과화면
 - 수행한 내용을 캡처 및 설명
 - 고찰
 - 과제를 수행하면서 느낀점 작성
 - Reference
 - 과제를 수행하면서 참고한 내용을 구체적으로 기록
 - 강의자료만 이용한 경우 생략 가능

Report Requirements

▪ Softcopy Upload

- 제출 파일
 - 보고서 + 소스파일 하나의 압축 파일로 압축하여 제출(tar.gz)
 - 보고서(.pdf. 파일 변환)
 - 소스코드(Comment 작성)
 - cli.c, srv.c
 - Makefile
 - 실행파일명: cli, srv
 - 소스 코드, 실행파일명 다르게 작성 시 감점
- Tar 압축 및 해제 방법
 - 압축 시 → tar -zcvf [압축 파일명].tar.gz[폴더 명]
 - 해제 시 → tar -zxvf 파일명.tar.gz
- 보고서 및 압축 파일 명 양식
- Assignment2_1 _수강분류코드_학번_이름 으로 작성

수강요일	이론1 월5수6	이론2 목4	실습1 금12	실습2 금56
수강분류 코드	A	B	C	D

- 예시-이론 월5 수6 수강하는 학생인 경우
 - 보고서 Assignment2_1_A_2024123456_홍길동.pdf
 - 압축 파일 명: Assignment2_1_A_2024123456_홍길동.tar.gz

Report Requirements

- 실습 수업을 수강하는 학생인 경우

- 실습 과목에 과제를 제출(.tar.gz)
- 이론 과목에 간단한 .txt 파일로 제출

📄 실습수업때제출했습니다.

2022-08-29 오후 3:58

텍스트 문서

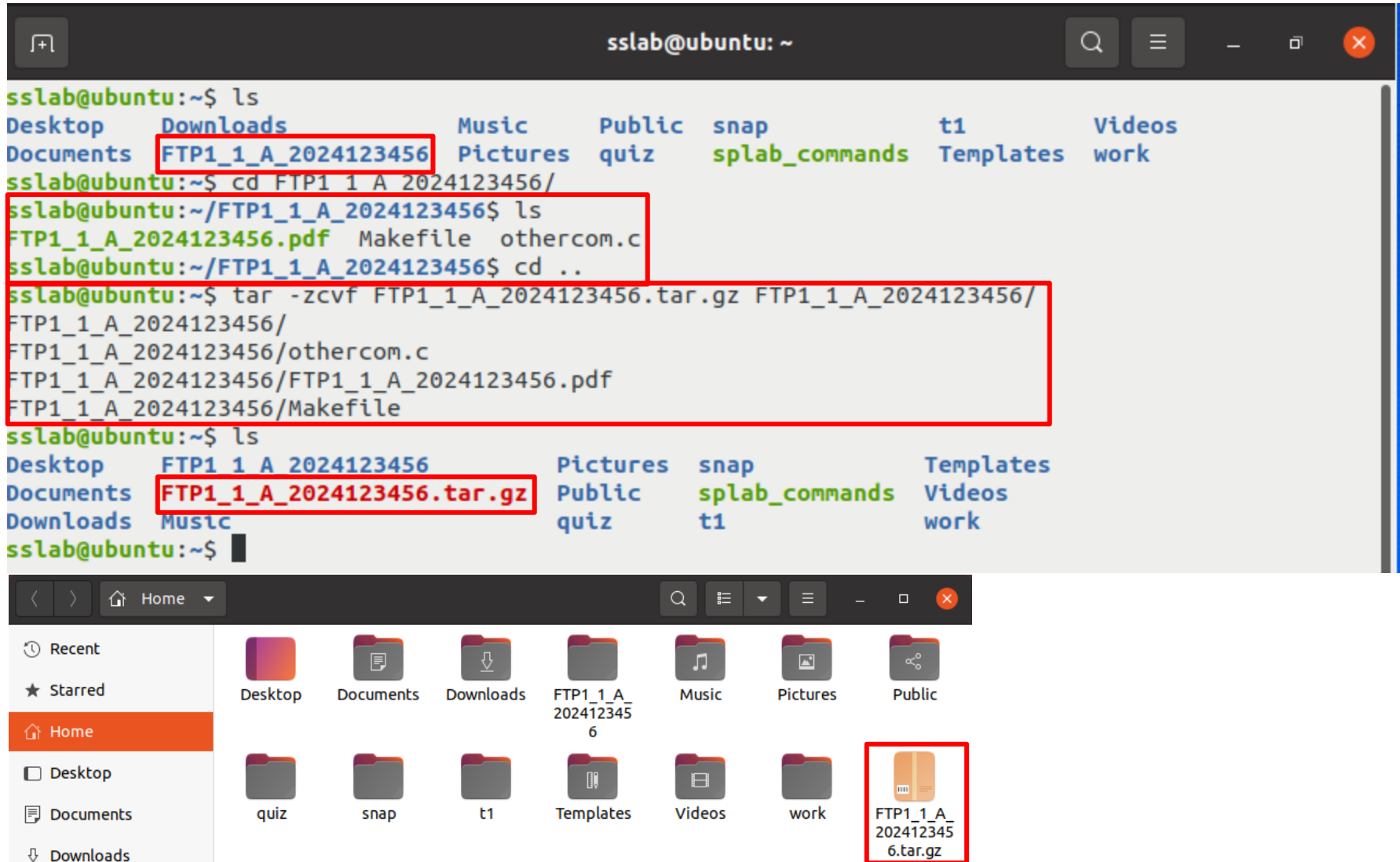
0KB

- 이론 과목에 .txt 파일 미 제출 시 감점
- .tar.gz 파일로 제출 하지 않을 시 감점

- 과제 제출

- KLAS – 강의 과제 제출
- 2024년 5월 02일 목요일 23:59까지 제출
- 딜레이 받지 않음
 - 제출 마감 시간 내 미제출시 해당 과제 0점 처리
 - 교내 서버 문제 발생 시, 메일로 과제 제출 허용

Appendix A. tar.gz compression



The image shows a terminal window and a file manager window illustrating the process of creating a tar.gz archive.

Terminal Window:

```
sslab@ubuntu: ~  
sslab@ubuntu:~$ ls  
Desktop Downloads Music Public snap t1 Videos  
Documents FTP1_1_A_2024123456 Pictures quiz splab_commands Templates work  
sslab@ubuntu:~$ cd FTP1_1_A_2024123456/  
sslab@ubuntu:~/FTP1_1_A_2024123456$ ls  
FTP1_1_A_2024123456.pdf Makefile othercom.c  
sslab@ubuntu:~/FTP1_1_A_2024123456$ cd ..  
sslab@ubuntu:~$ tar -zcvf FTP1_1_A_2024123456.tar.gz FTP1_1_A_2024123456/  
FTP1_1_A_2024123456/  
FTP1_1_A_2024123456/othercom.c  
FTP1_1_A_2024123456/FTP1_1_A_2024123456.pdf  
FTP1_1_A_2024123456/Makefile  
sslab@ubuntu:~$ ls  
Desktop FTP1_1_A_2024123456 Pictures snap Templates  
Documents FTP1_1_A_2024123456.tar.gz Public splab_commands Videos  
Downloads Music quiz t1 work  
sslab@ubuntu:~$
```

File Manager Window:

The file manager shows the home directory with a sidebar on the left containing 'Recent', 'Starred', 'Home', 'Desktop', 'Documents', and 'Downloads'. The main pane displays a grid of folders and files. A red box highlights the file `FTP1_1_A_2024123456.tar.gz` in the bottom right corner of the grid.

Appendix B. Comment 작성 요령 (1/3)

- File Head Comment

```
////////////////////////////////////  
// File Name      : Main.c                               //  
// Date           : 2024/03/01                           //  
// OS              : Ubuntu 20.04.6 LTS 64bits           //  
//               //  
// Author          : Hong Gil Dong                       //  
// Student ID      : 2024123456                           //  
// ----- //  
// Title : System Programming Assignment #1-1 ( ftp server ) //  
// Description : ...                                       //  
////////////////////////////////////
```


Appendix B. Comment 작성 요령 (2/3)

- Function Head Comment

```
////////////////////////////////////  
// InsertNode                                                    //  
// =====                                                    //  
// Input: Node* -> Insert Node,                                  //  
//           Node* -> Column node before insert node           //  
//           Node* -> Row node before insert node               //  
//           (Input parameter Description)                        //  
// Output: int  - 1 success                                       //  
//           0 fail                                              //  
//           (Out parameter Description)                          //  
// Purpose: Inserting node                                       //  
////////////////////////////////////
```

Appendix B. Comment 작성 요령 (3/3)

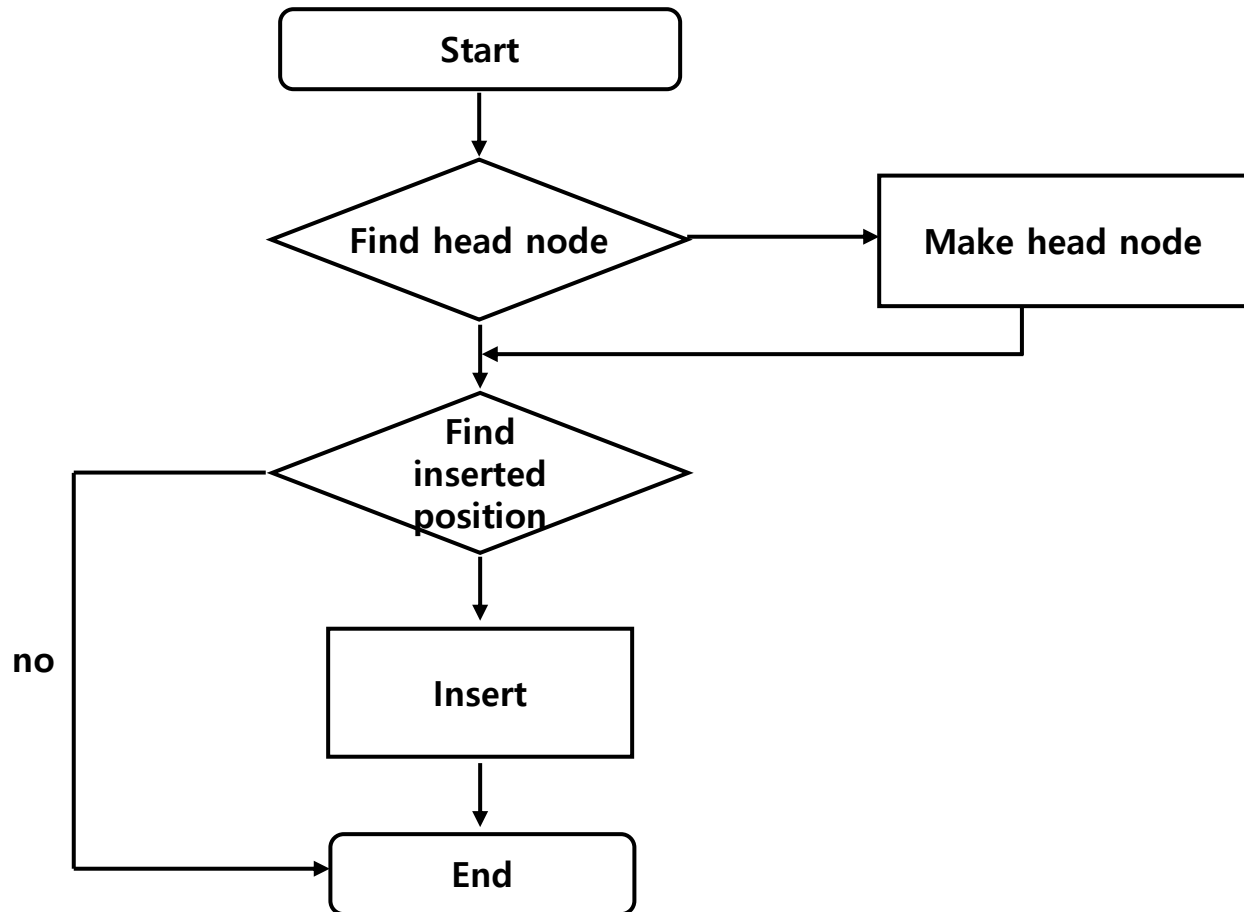
- In-line Comment

```
//////////////////////////////////// Row insert //////////////////////////////////////
if( pRowPos->pNextRow != pRowPos ) {
    pTemp->pNextRow = pRowPos->pNextRow;          // pTemp set next row
    if( !( pRowPos->pNextRow->bHead ) ){
        pRowPos->pNextRow->NodeItem.pPrevRow = pTemp;
    } // end of if
} // end of if
else {
    pTemp->pNextRow = pRowPos;                    // pTemp set next row
} // end of else
pTemp->NodeItem.pPrevRow = pRowPos;              // pTemp set previous row
pRowPos->pNextRow = pTemp;
//////////////////////////////////// End of row insert //////////////////////////////////////
```

Appendix C. 보고서 작성 요령 (1/2)

- Algorithm – Flow Chart (Each function)

- E.g.



Appendix C. 보고서 작성 요령 (2/2)

- Algorithm – Pseudo Code

```
FixHeap(Node *root, Key k)
{
    Node vacant, largerChild;
    vacant = root;
    while( vacant is not leaf ) {
        largerChild = the child of vacant with the larger key;
        if( k < largerChild's Key ) {
            copy largerChild's key to vacant;
            vacant = largerChild;
        }
        else exit loop;
    }
}
```