

2024년 1학기 시스템프로그래밍 & 시스템 프로그래밍 실습

# Assignment3 - 3

**System Software Laboratory**  
College of Software and Convergence  
Kwangwoon Univ.

# Requirements (1/7)

FTP Command	Option	Action
NLST	X	directory에 존재하는 file/directory 출력
	-a	directory에 존재하는 모든 file/directory 출력 (hidden file까지 모두 출력)
	-l	directory에 존재하는 file/directory들의 정보들 까지 출력
	-al	directory에 존재하는 모든 file/directory 들의 (hidden file까지) 정보들을 모두 출력
LIST		NLST -al 동작과 동일
PWD		현재 동작하는 directory 출력 (Print working directory)
CWD		Directory 이동 (Change working directory)
CDUP		이전 Directory로 이동
MKD		Directory 생성 (Make Directory)
DELE		File 삭제 (Delete file)
RMD		Directory/File 삭제 (Remove directory or file)
RNFR & RNTD		Old name에서 New name으로 이름 변경 Rename From (old name) Rename to (new name)
QUIT		해당 (Child) Process 종료

# Requirements (2/7)

- **Command 구현 사항은 Assignment #1 참고**
  - Command 구현은 어떠한 상황에서도 잘 처리되어야 함.  
ex) 파일 개수가 많아도 ls로 명령어가 제대로 동작 해야 함.
- srv의 port 넘버는 인자로 받을 것.
  - ex) `./srv portnum`  
`./srv 8888`   `./srv 9999` (portnum은 임의의 수)
- using **read** system call through STANDARD INPUT
- using **write** system call though STANDARD OUTPUT

※ fnmatch 함수와 같은 패턴 매칭 함수는 사용 금지.

# Requirements (3/7)

- **Access control, User authentication 구현**
  - access.txt, passwd 파일 생성 후 활용
  - Access control을 먼저 판별하며, 로그인 기회는 최대 세 번
- **Additional Command**
  - PORT - open a data **port**
    - automatically generated command
  - RETR - **re**trieve a remote file
    - user command: **get**
  - STOR - **st**ore a file on the remote host
    - user command: **put**
  - TYPE - set transfer **type**
    - user command: **type**
    - user command: **bin** == type binary
    - user command: **ascii** == type ascii
  - USER - send "USER username" to server
  - PASS - send "PASS password" to server

# Requirements (4/7)

- **control connection과 data connection을 구현하시오.**
  - **ls, get, put** 명령어 사용시 **PORT**명령어를 이용하여 data connection을 연결하여 통신.
    - data connection은 Server가 reply code로 266과 “Complete transmission.” 메시지를 보내고 connection을 끝낸다.
  - 예)
    - client가 명령어 'ls' 입력 시, server는 처리결과를 data connection을 통해 전송.
      - data connection을 설정하기 위해, client는 임의 포트번호를 생성하고 명령어 **PORT** 를 이용해 server에게 전달
      - server는 PORT 명령어에 대한 ack 전송 후, 정해진 포트번호를 이용하여 client에 접속
      - data connection은 한 번 사용 후 close.
  - data connection port number
    - **client : 10001~60000 사이의 임의의 번호 지정**
    - **server : client가 지정한 번호로 연결 시도**
  - **ls, get, put이 제대로 동작하는지 보기 위해 cli와 srv를 다른 path에 두고 채점할 예정.**
    - 채점할 시에 cli 와 srv를 위치할 path는 채점자가 임의로 정함.
    - 뒤의 Example-get, Example-put 참고

# Requirements (5/7)

- **Type**

- Default : binary mode
- 현재 모드가 binary 모드인지 ascii 모드인지 구분할 수 있어야 한다.
- 현재 모드에 따라 reply code 150일 때, message를 달리 해야 한다.
- ex) binary mode : Opening **binary** mode data connection for "*filename.*".  
ascii mode : Opening **ascii** mode data connection for "*filename.*".

- **Handle Control Messages**

- 자세한 Control Message들에 대한 설명은 다음에 명시되어 있음.

- **Example 결과를 참고**

# Requirements (6/7)

- **Log File Generation**
- **Process at Server**
  - Write log information on log file
    - Specify **“the date”, “client IP”, “client Port”, and “user”** in front of every log.
      - Except when server is started, and terminated.
      - If before getting *username* from user, leave the space for “user” empty.
    - When server program is started
    - When illegal user connects to server
    - When authenticated-user connects to server
    - **When server receives the FTP command, log the command.**
    - **When server sends the result of the command, log it.**
      - data connection으로 transfer한 내용은 log하지 말고 reply & message 로깅 .
      - ls/get/put 의 경우, reply code(200, 550, 150, 226)와 그에 상응하는 message만 로깅.
        - RETR(get), STOR(put)의 경우, 무슨 type인지 로깅할 것.
        - RETR(get), STOR(put)의 경우, 전송한 byte수 로깅할 것.
      - 다자 client에 대한 동기화 문제는 해결하지 않아도 된다.
    - When client disconnects to server
    - When server is terminated **(ctrl – c)**
  - **FTP3-3 강의자료의 Log File Generation 참고**
  - Log File
    - 로그 파일의 이름은 “logfile”로 한다.
    - 서버의 binary file (“srv”)과 동일한 위치에 위치한다.**(사전에 존재할 경우 이어서 작성)**

# Requirements (7/7)

- Client 에서는 Server가 보낸 reply code와 message를 STANDARD OUTPUT으로 출력.
- Server에서는 Client로 보낸 reply code와 message를 STANDARD OUTPUT으로 출력.
- get/put 예외처리
  - 사전에 동일한 이름의 파일이 존재할 경우 Failed transmission
  - Failed transmission 의 경우 파일이 생성되지 않게 처리
- Welcome message
  - “motd” 파일 사용
  - [FTP3-3](#) 강의자료 Server’s Welcome Message 참고



# Control Messages (1/11)

- **IP Access Control**

- If client's IP is in the access.txt ("ACCEPTED")
  - Server replies code **220** with welcome message.
  - If Client get reply code 220 with welcome message from server, then start User Authentication.
  - ex) **220 sswlab.kw.ac.kr FTP server (version myftp [1.0] Fri May 30 14:40:36 KST 2014) ready.**
- If client's IP is not in the access.txt ("REJECTION")
  - Server replies code **431** with message "This client can't access. Close the session."

# Control Messages (2/11)

## ■ USER Authentication

- When server replies code 220, then receive username from user, client send to server “**USER** *username*”.
- If there are username in “passwd” file
  - Server replies code **331** with message “Password is required for *username*.”
- If there are not username in “passwd” file
  - Server replies code **430** with message “Invalid username or password”.
- When receive reply code 331, then receive password from user, and send to server “**PASS** *password*”.
- If password is match
  - then server replies code **230** with message “User *username* logged in.”
- If password isn’t match
  - Server replies code **430** with message “Invalid username or password”.
- When it fail to log-in on third time (“**DISCONNECTION**”)
  - Server replies code **530** with message “Failed to log-in”

# Control Messages (3/11)

- **pwd (PWD)**

- Display the current working directory (current directory)
- Server replies message includes reply code **257** with the **current directory**.
- Prompt messages should follow the format of the **FTP**
  - 257 “directory path” is current directory
  - ex) 257 “/home/user/sp” is current directory

- **cd (CWD)**

- Change your current working directory
- If the command is successful
  - Server replies code **250** with message “CWD command succeeds.”.
- If the command is failure
  - Server replies code **550** with message “*argument*: Can’t find such file or directory”.
  - ex) cd sys-exam  
550 sys-exam: Can’t find such file or directory.

# Control Messages (4/11)

- **cd.. (CDUP)**

- Change working directory to upper directory.
- If the command is successful
  - Server replies code **250** with message “CWD command performed successfully”.
- If the command is fail
  - Server replies code **550** with message “*argument*: Can’t find such file or directory”.
  - ex) cd sys-exam  
550 sys-exam: Can’t find such file or directory.

- **delete (DELE)**

- delete file in remote side.
- If the command is successful
  - Server replies code **250** with message “DELE command performed successfully”.
- If the command is fail
  - Server replies code **550** with message “*argument*: Can’t find such file or directory”.
  - ex) delete sys-exam  
550 sys-exam: Can’t find such file or directory.

# Control Messages (5/11)

- **rename (RNFR ... RNTD ...)**
  - Renames a file in the remote-directory
  - Rename has **two** FTP commands : RNFR old-name RNTD new-name
- **RNFR:** original remote file name
  - If the command is successful
    - Server replies code **350** with message “File exists, ready to rename”.
  - If the command is fail
    - Server replies code **550** with message “*argument*: Can’t find such file or directory”.
    - ex) rename sys-exam sys-eee  
550 sys-exam: Can’t find such file or directory.
- **RNTD:** file name to change
  - If the command is successful
    - Server replies code **250** with message “RNTD command succeeds”.
  - If the command is fail
    - Server replies code **550** with message “*argument*: can’t be renamed”.
    - ex) rename sys-exam sys-eee  
550 sys-exam: can’t be renamed.

# Control Messages (6/11)

- **Is (NLST)**

- list files & directories.
- open Data connection using PORT command.
- If it is connected successfully
  - Server replies code **200** with message “PORT command performed successfully.”
- If it is fail to connect
  - Server replies code **550** with message “Failed to access”.
- After successful connection using PORT command, send converted command to server.
  - Server replies code **150** with message “Opening data connection for directory list.”.
- Client receives result of “Is” through Data connection.
- Server replies code **226** with message “Complete transmission.”, and close the data connection.
- Server replies code **550** with message “Failed transmission.”, and close the data connection.
- Client displays how many bytes it received from server.
  - ex) “OK. 2308 bytes is received.”

# Control Messages (7/11)

- **get (RETR) – get, put할 때 주고받을 파일을 다른 경로에 두고 확인**
  - open Data connection using PORT command.
  - If it is connected successfully
    - Server replies code **200** with message “PORT command performed successfully.”
  - If it is fail to connect
    - Server replies code **550** with message “Failed to access”.
  - After successful connection using PORT command, send converted command to server.
    - Server replies code **150** with message “Opening **binary/ascii** mode data connection for *filename*.”.
  - Client **gets the file** named *filename* **from server**.
  - Server replies code **226** with message “Complete transmission.”, and close the data connection.
  - Server replies code **550** with message “Failed transmission.”, and close the data connection.
  - Client displays how many bytes it got from server.
    - ex) “OK. 2308 bytes is received.”

## Example - get

- cli 와 srv를 다른 경로에 위치시킴.

```
shy@shy-virtual-machine:~/test1$ ls
```

```
cli
```

```
shy@shy-virtual-machine:~/test2$ ls
```

```
cli.c exam1 srv srv.c
```

- Client 에서 get 명령어로 exam1을 Server로부터 copy해온다.

```
shy@shy-virtual-machine:~/test1$ ./cli 127.0.0.1 8888
```

```
> ls
```

```
.  
cli.c  
srv.c  
exam1  
srv  
..
```

```
> get exam1
```

- cli가 있는 쪽에 exam1이 있는 것을 확인한다.

```
shy@shy-virtual-machine:~/test1$ ls
```

```
cli exam1
```



# Control Messages (8/11)

- **put (STOR)**
  - open Data connection using PORT command.
  - If it is connected successfully
    - Server replies code **200** with message “PORT command performed successfully.”
  - If it is fail to connect
    - Server replies code **550** with message “Failed to access”.
  - After successful connection using PORT command, send converted command to server.
    - Server replies code **150** with message “Opening **binary/ascii** mode data connection for *filename*.”.
  - Client **puts the file** named *filename* **to server**.
  - Server replies code **226** with message “Complete transmission.”, and close the data connection.
  - Server replies code **550** with message “Failed transmission.”, and close the data connection.
  - Client displays how many bytes it put to server.
    - ex) “OK. 2308 bytes is sent.”

## Example - put

- cli 와 srv를 다른 경로에 위치시킴.

```
shy@shy-virtual-machine:~/test1$ ls
```

```
cli vhd1
```

```
shy@shy-virtual-machine:~/test2$ ls
```

```
cli.c srv srv.c
```

- Client 에서 put 명령어로 vhd1를 Server로 copy한다.

```
shy@shy-virtual-machine:~/test1$ ./cli 127.0.0.1 8888
```

```
> ls
```

```
.
```

```
cli.c
```

```
srv.c
```

```
srv
```

```
..
```

```
> put vhd1
```

- srv가 있는 쪽에 vhd1이 있는 것을 확인한다.

```
shy@shy-virtual-machine:~/test2$ ls
```

```
cli.c srv srv.c vhd1
```

# Control Messages (9/11)

- **mkdir (MKD)**

- make a directory in the remote-directory
- If the command is successful
  - Server replies code **250** with message “MKD command performed successfully”.
- If the command is fail
  - Server replies code **550** with message “*argument*: can’t create directory”.
  - ex) mkdir sys-exam  
550 sys-exam: can’t create directory.

- **rmdir (RMD)**

- remove a directory in the remote-directory.
- If the command is successful
  - Server replies code **250** with message “RMD command performed successfully”.
- If the command is fail
  - Server replies code **550** with message “*argument*: can’t remove directory”.
  - ex) rmdir sys-exam  
550 sys-exam: can’t remove directory.

# Control Messages (10/11)

- **bin / type binary**
  - convert to “**TYPE I**”.
  - If the command succeed
    - Server replies code **201** with message “Type set to I.”.
  - If the command fail
    - Server replies code **502** with message “Type doesn’t set.”
- **ascii / type ascii**
  - convert to “**TYPE A**”.
  - If the command succeed
    - Server replies code **201** with message “Type set to A.”.
  - If the command fail
    - Server replies code **502** with message “Type doesn’t set.”

# Control Messages (11/11)

- **quit (QUIT)**
  - close the control connection.
  - client sends “QUIT” to server.
  - If it is disconnected successfully
    - Server replies code **221** with message “Goodbye”.

## Example (1/5)

---> ~~~ ➔ Data sent to Server

```
#saurus:/home1/grad/saurus> ./cli 127.0.0.1 10000
```

```
Connected to sswlab.kw.ac.kr.
```

```
220 sswlab.kw.ac.kr FTP server (version myftp [1.0] Fri May 30 14:40:36  
KST 2014) ready.
```

```
Name : sslab
```

```
---> USER sslab
```

```
331 Password required for sslab.
```

```
Password:
```


```
---> PASS *****
```

```
230 User sslab logged in.
```

```
ftp> pwd
```

```
---> PWD
```

```
257 "/home/sslalab" is current directory.
```



**'motd'** file에 기록된  
내용을 server에서 읽  
어온 후 client에 전달

## Example (2/5)

---> ~~~ → Data sent to Server

```
ftp> ls -al
```

```
---> PORT 127,0,0,1,150,33
```

```
200 PORT command performed successfully.
```

```
---> NLST -al
```

```
150 Opening data connection for directory list.
```

```
drwxr-xr-x  11  sslab  sslab      2048 Sep 14 21:38 .
```

```
.....
```

```
drwxr--r--   1  sslab  sslab      2048 Jan 16  2013 sys_exam
```

```
-rwxr--r--   1  sslab  sslab    109056 Jan 16  2013 unwise.exe
```

```
-rwxr--r--   1  sslab  sslab    204858 May 18  2013 kernel
```

```
226 Complete Transmission.
```

```
OK. 2308 bytes is received
```

## Example (3/5)

---> ~~~ ➔ Data sent to Server

```
ftp> cd sys-exam
```

```
---> CWD sys-exam
```

```
550 sys-exam: Can't find such file or directory.
```

```
ftp> cd sys_exam
```

```
---> CWD sys_exam
```

```
250 CWD command performed successfully.
```

```
ftp> pwd
```

```
---> PWD
```

```
257 "/home/sslabs/sys_exam" is current directory.
```

```
ftp> cd ..
```

```
---> CDUP
```

```
250 CWD command performed successfully.
```



## Example (4/5)

---> ~~~ ➔ Data sent to Server

```
ftp> bin
---> TYPE I
201 Type set to I.
ftp> ascii
---> TYPE A
201 Type set to A.
ftp> type binary
---> TYPE I
201 Type set to I.
ftp> type ascii
---> TYPE A
201 Type set to A.
ftp> delete vhd1
---> DELE vhd1
250 DELE command performed successfully.
```

```
ftp> rename imsi temp
---> RNFR imsi
350 File exists, ready to rename.
---> RNT0 temp
250 RNT0 command performed successfully.
ftp> mkdir imsi
---> MKD imsi
250 MKD command performed successfully.
ftp> rmdir imsi
---> RMD imsi
250 RMD command performed successfully.
```

## Example (5/5)

---> ~~~ → Data sent to Server

```
ftp> get exam1
---> PORT 128,134,54,60,159,153
200 PORT command performed
    successfully.
---> RETR exam1
150 Opening BINARY mode data
connection for exam1.
226 Complete transmission.
OK. 24644 bytes is received
```

```
ftp> put vhd1
---> PORT 128,134,54,60,159,155
200 PORT command performed
successfully.
---> STOR vhd1
150 Opening BINARY mode data
connection for vhd1.
226 Complete transmission.
OK. 204858 bytes is sent
ftp> quit
---> QUIT
221 Goodbye.
```

# 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 → Assignment3-3
  - **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
    - 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
- **Assignment3\_3\_수강분류코드\_학번**으로 작성

수강요일	이론1 월5수6	이론2 목4	실습1 금12	실습2 금56	실습3 금78
수강분류 코드	A	B	C	D	E

- 예시-이론 월5 수6 수강하는 학생인 경우
  - 보고서 Assignment3\_3\_A\_2024123456.pdf
  - 압축 파일 명: Assignment3\_3\_A\_2024123456.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

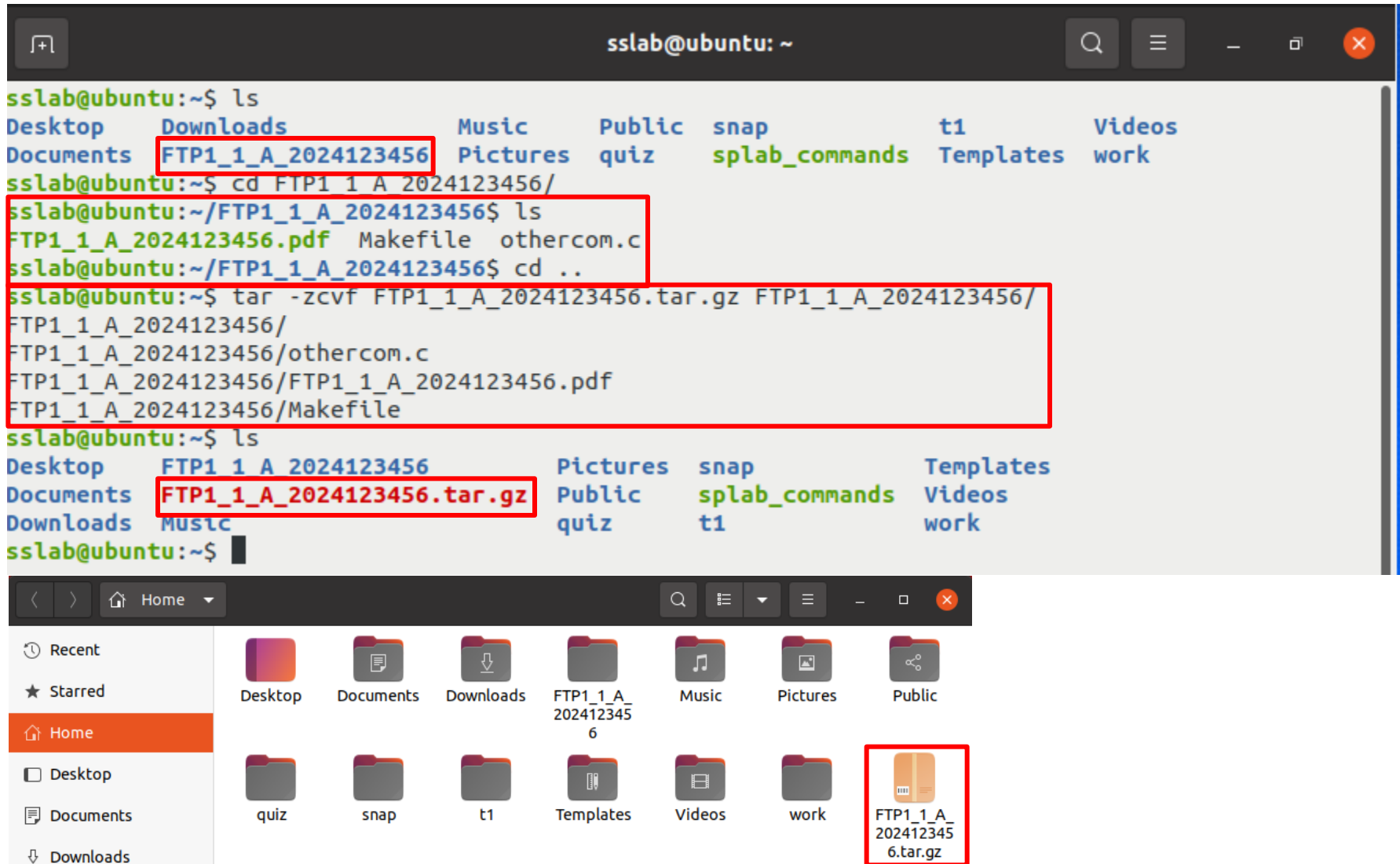
0KB

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

- 과제 제출

- KLAS – 강의 과제 제출
- 2024년 6월 06일 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



# 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)

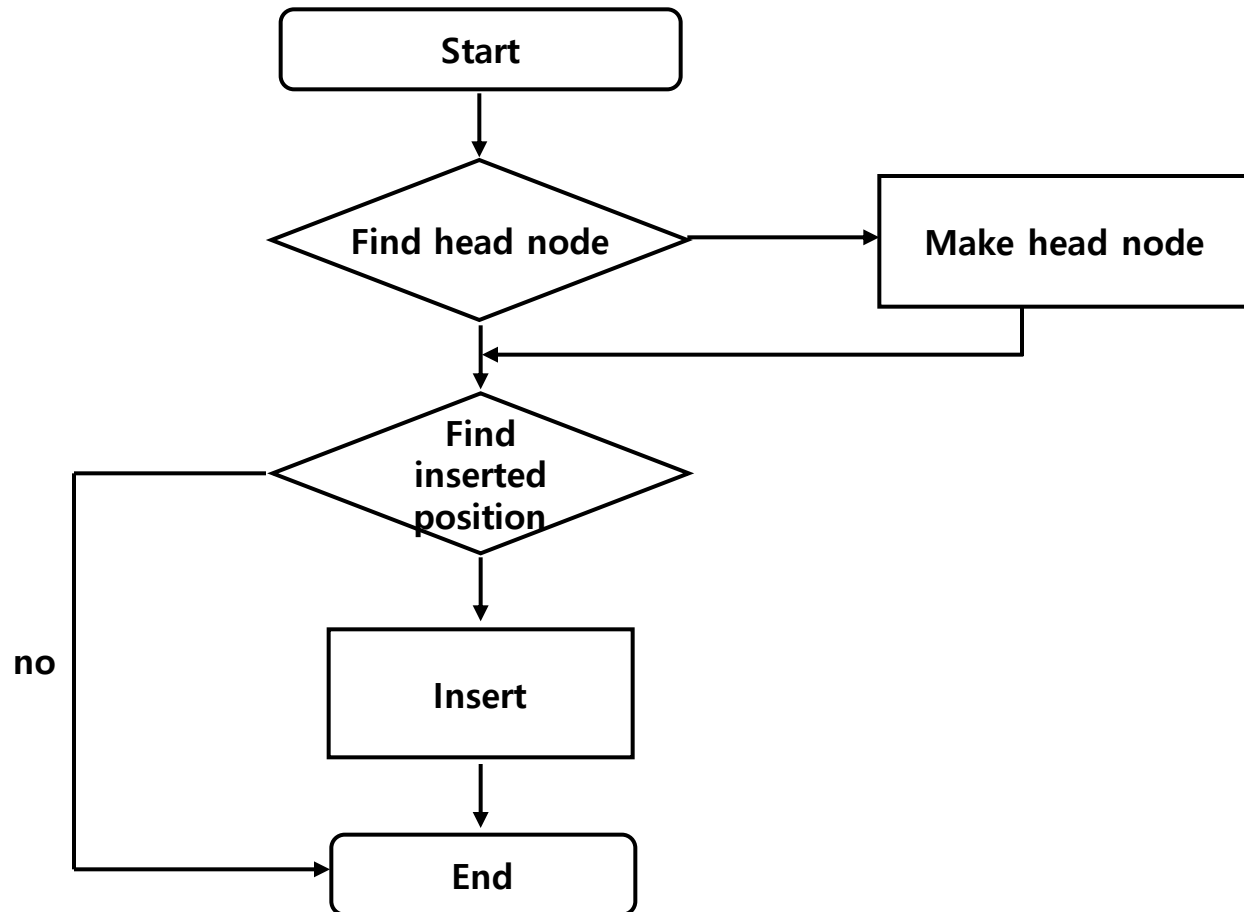
- In-line Comment

```
//////////////////////////////////// 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;
    }
}
```