

2024년 1학기 시스템프로그래밍 & 시스템 프로그래밍 실습

Assignment3 - 2

System Software Laboratory
College of Software and Convergence
Kwangwoon Univ.

Assignment 3-2

- **control connection과 data connection을 구현하시오.**
 - client가 명령어 'ls' 입력 시, server는 처리결과를 data connection을 통해 전송
 - data connection을 설정하기 위해, client는 임의 포트번호를 생성하고 명령어 PORT 를 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 : 10001~30000 사이의 임의의 번호 지정
 - server : client가 지정한 번호로 연결 시도

Control Messages (1/2)

- **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 (1/2)

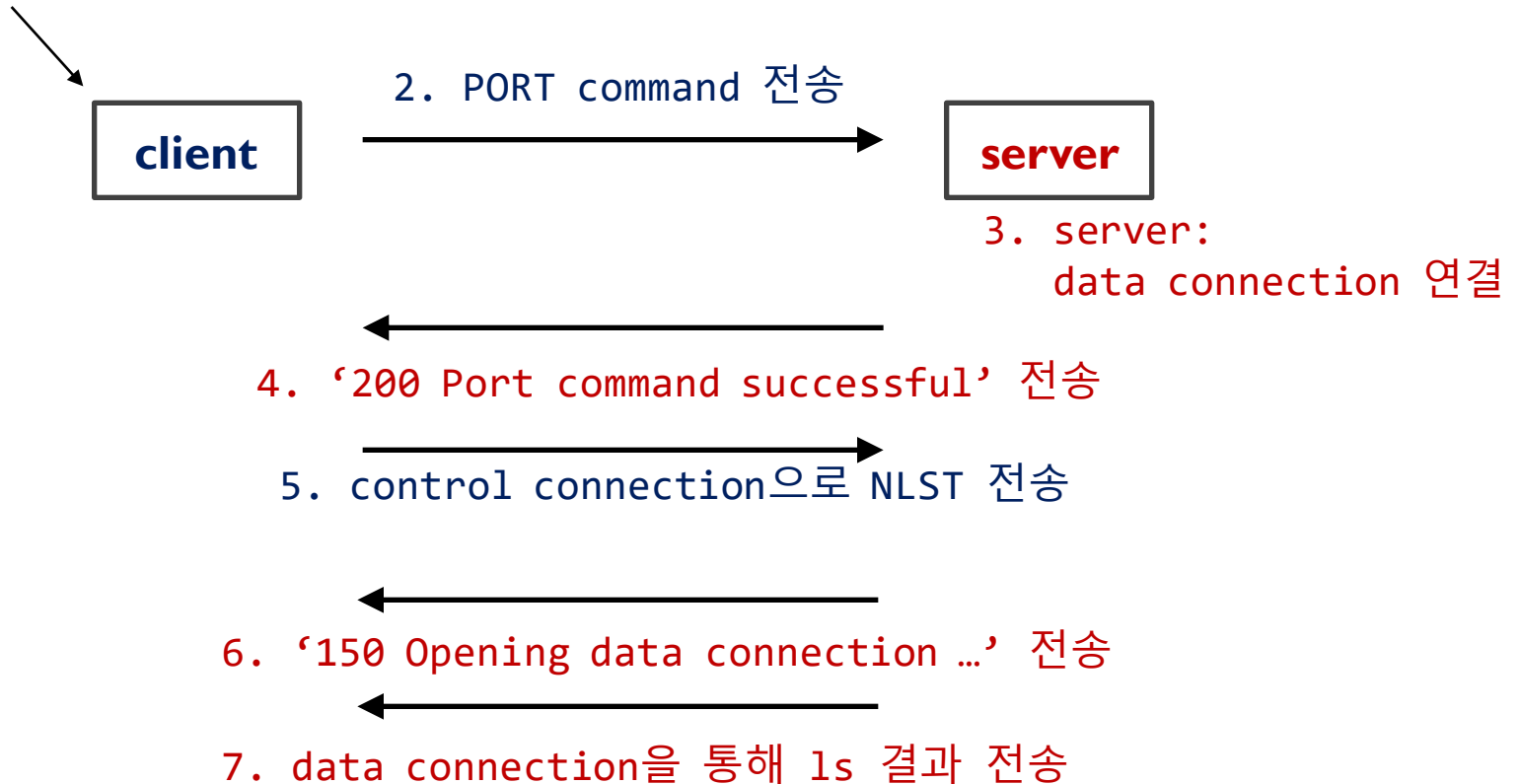
- **quit (QUIT)**
 - close the control connection.
 - client sends “QUIT” to server.
 - If it is disconnected successfully
 - Server replies code **221** with message “Goodbye”.

Assignment 3-2

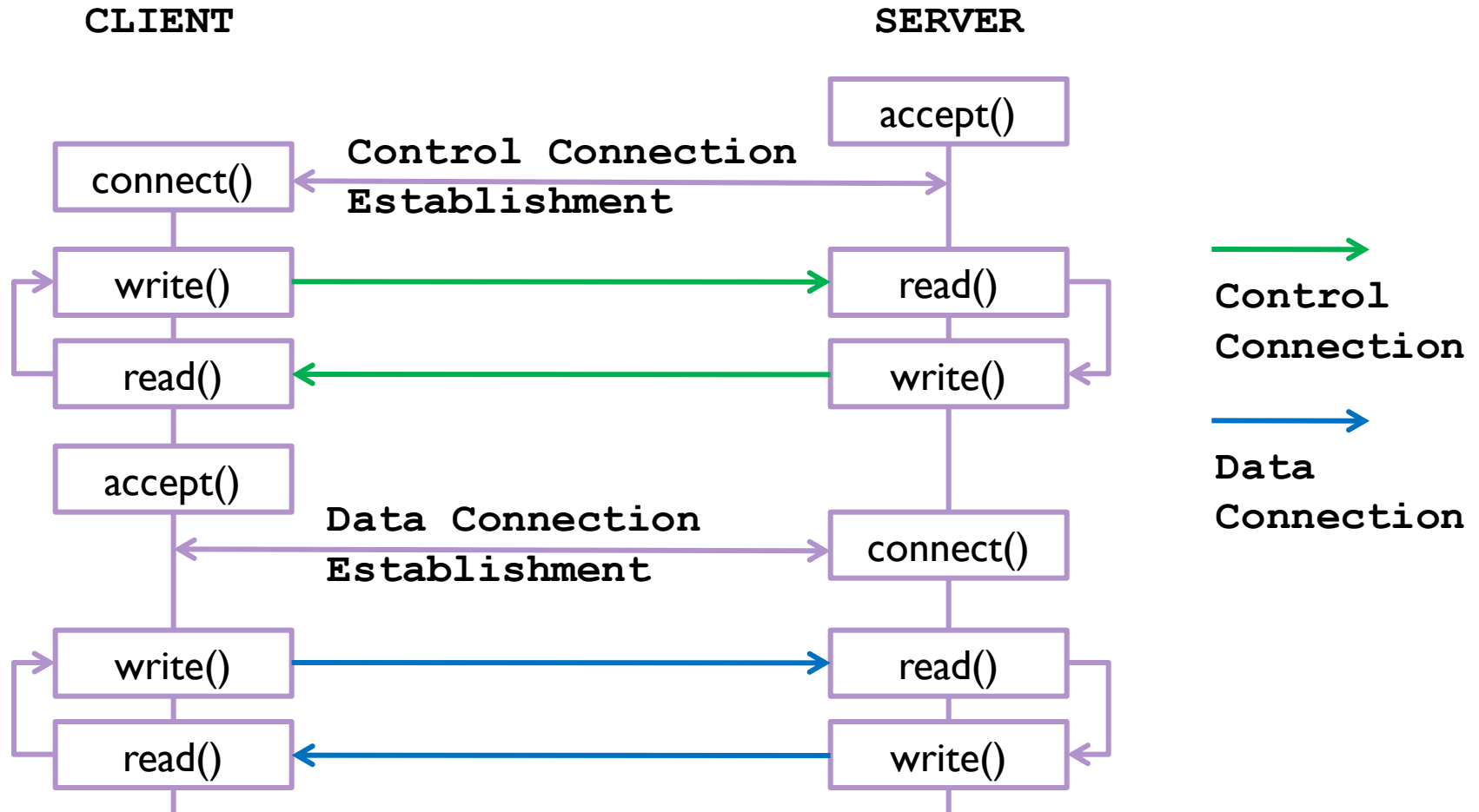
■ 동작 순서

※ 콘솔 출력은 example result 참고

1. 사용자, ls 입력 (control connection은 연결 상태)



Assignment 3-2



Assignment 3-2 – cli.c

```
void main(int argc, char **argv)
{
    char *hostport;
    struct sockaddr_in temp;
    /* your own codes */
    hostport = convert_addr_to_str(temp.sin_addr.s_addr, temp.sin_port);
    printf("converting to %s\n", hostport);
    /* make control connection and data connection */
    /* your own codes */
}

char* convert_addr_to_str(unsigned long ip_addr, unsigned int port)
{
    //자신의 IP주소와 임의의 포트번호를 PORT명령어에 붙는 형태로 변경
    char *addr;
    /* your converting algorithm */
    return addr;
}
```

Assignment 3-2 – srv.c

```
void main(int argc, char **argv)
{
    char *host_ip;
    char temp[25];
    unsigned int port_num;
    /* your own codes */
    /* make control connection */
    host_ip = convert_str_to_addr(temp, (unsigned int *) &port_num);
    /* your own codes */
    /* make data connection */
    /* your own codes */
}

char* convert_str_to_addr(char *str, unsigned int *port)
{ //client로부터 받은 PORT명령어에 붙은 IP주소와 포트번호를 변경
  char *addr;
  /* your converting algorithm */
  return addr;
}
```


Example result

client

```
user@ubuntu:~/sp$ ./cli 127.0.0.1 10000
> ls
converting to 127,0,0,1,48,57
200 Port command performed successful
150 Opening data connection for directory list.
cli
cli.c
makefile
srv
srv.c
226 Complete transmission.
OK. xxxx bytes is received.
>quit
221 Goodbye.
user@ubuntu:~/sp$
```

server

```
user@ubuntu:~/sp$ ./srv 10000
PORT 127,0,0,1,48,57 ※ 48,57 => 12345
200 Port command performed successful
NLST
150 Opening data connection for directory list
226 Complete transmission.
QUIT
221 Goodbye.
^C
user@ubuntu:~/sp$
```

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 → Assignment3-2
 - **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_2_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_2_A_2024123456.pdf
 - 압축 파일 명: Assignment3_2_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

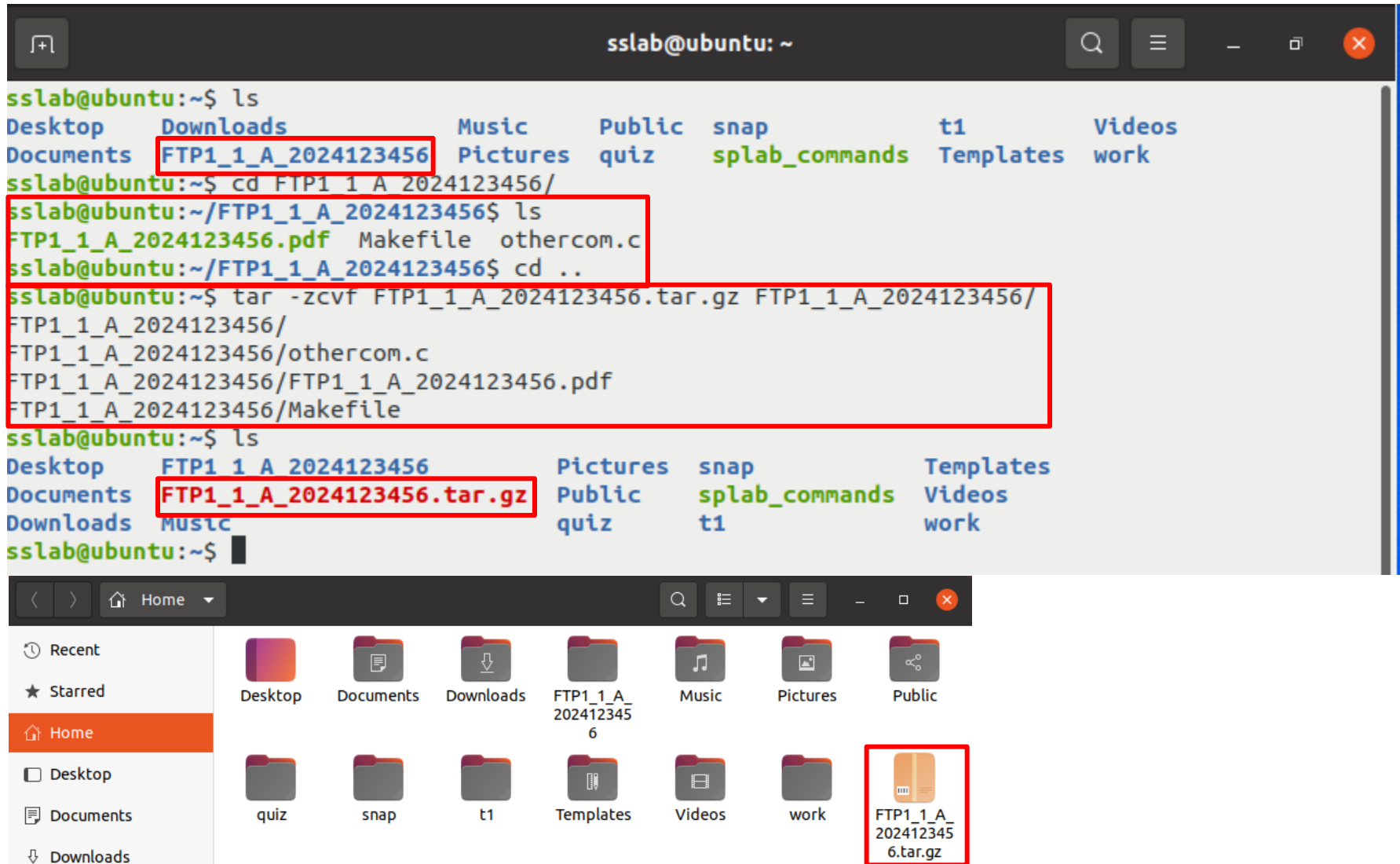
0KB

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

- 과제 제출

- KLAS – 강의 과제 제출
- 2024년 5월 30일 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)

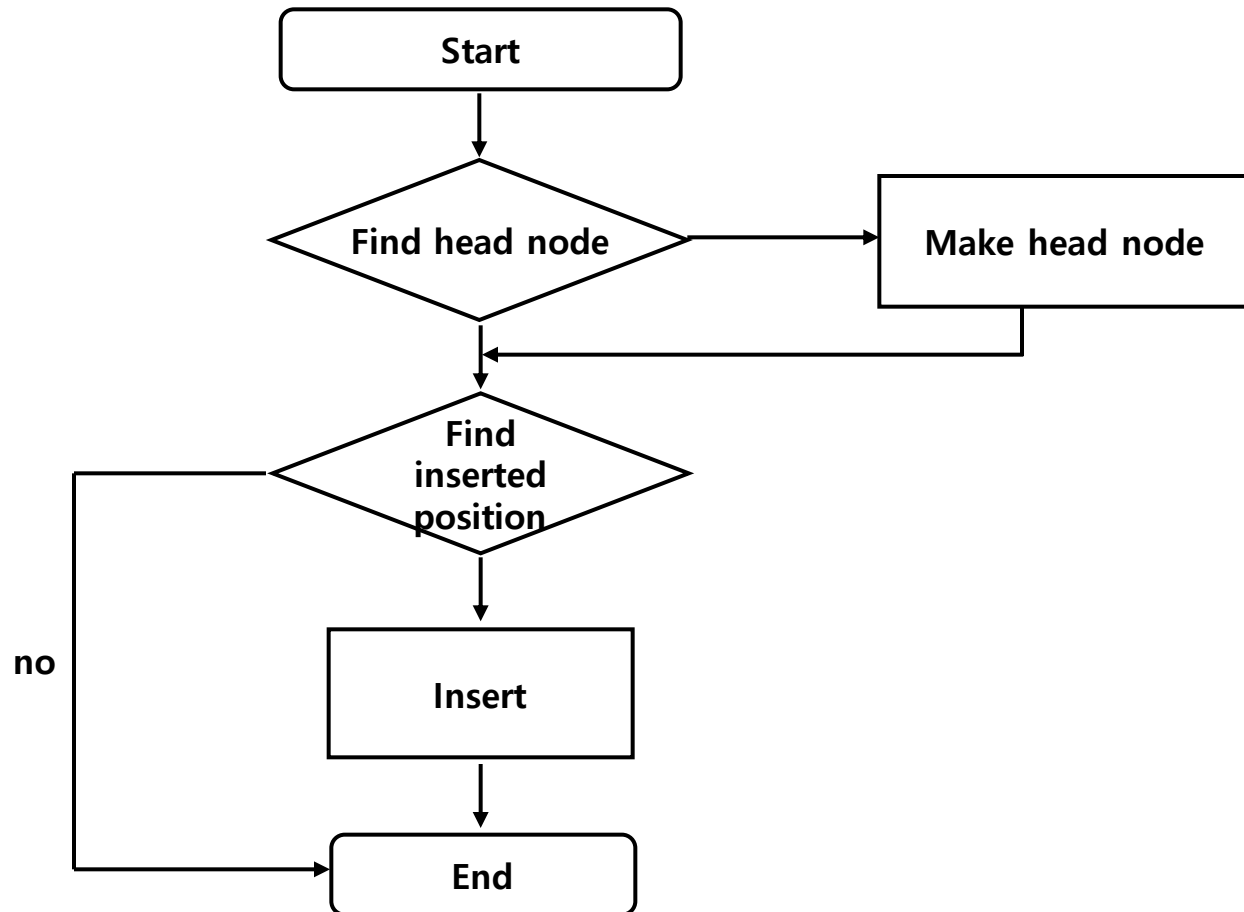
- In-line Comment

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