

2024년 1학기 시스템프로그래밍 & 시스템 프로그래밍 실습

# Assignment2 - 2

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

# Assignment2-2 Requirements (1/2)

- Client

- p.4~p.5의 소스코드를 그대로 이용.
- 실행 시, 인자로 서버의 주소와 포트번호를 각각 입력 받음.
  - E.g.) `./cli 127.0.0.1 10000`
- 서버와의 연결이 성공하면, 아래와 같은 행위를 반복함. 단, 아래 동작 중 하나라도 실패하면 서버와의 연결 종료 후 해당 프로세스 종료.
  - (1) 사용자로부터 문자열 입력 받음.
  - (2) 서버로 해당 문자열 보냄.
  - (3) 서버로부터 문자열 받아서 그대로 출력.

# Assignment2-2 Requirements (2/2)

- **Server**

- p.6 ~ p.8을 그대로 이용하여 작성할 것.
  - 단, p.7의 주석(`/* ... */`)들이 있는 공간에 아래 조건에 맞게 스스로 채울 것.
- 실행 시, 인자로 포트번호를 입력 받음.
  - E.g.) `./srv 1000`
- Client와 연결된 후, 새로운 프로세스를 생성하여 다음과 같은 작업을 각각 수행.
  - **Parent Process** → 연결된 Client의 정보 출력 (IP, Port) 및 다음 client와 연결 준비.
  - **Child Process** → Client에게 문자열을 받고, 해당 문자열을 그대로 다시 전송. 이를 반복.
    - 단, 해당 문자열이 **"QUIT"** 일 경우,
      - 해당 Client와 연결 종료.
      - **SIGALRM Signal**을 호출하여 1초 뒤에 해당 프로세스도 종료.
  - Process가 새로 생성될 때 마다 해당 Process의 PID 출력

## [Example code] cli.c – with fork () (1/2)

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <arpa/inet.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <sys/wait.h>
#include <signal.h>
|
#define BUF_SIZE 256

int main(int argc, char **argv)
{
    char buff[BUF_SIZE];
    int n;
    int sockfd;
    struct sockaddr_in serv_addr;

    sockfd = socket(AF_INET, SOCK_STREAM, 0);

    memset(&serv_addr, 0, sizeof(serv_addr));
    serv_addr.sin_family=AF_INET;
    serv_addr.sin_addr.s_addr=inet_addr(argv[1]);
    serv_addr.sin_port=htons(atoi(argv[2]));
    connect(sockfd, (struct sockaddr*)&serv_addr, sizeof(serv_addr));
```

## [Example code] cli.c – with fork () (2/2)

```
while(1) {
    write(STDOUT_FILENO, "> ", 2);
    read(STDIN_FILENO, buff, BUF_SIZE);

    if(write(sockfd, buff, BUF_SIZE) > 0){
        if(read(sockfd, buff, BUF_SIZE) > 0)
            printf("from server:%s",buff);
        else
            break;
    }else
        break;
}
close(sockfd);
return 0;
}
```

## [Example code] srv.c – with fork () (1/3)

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <arpa/inet.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <sys/wait.h>
#include <signal.h>

#define BUF_SIZE 256

void sh_chld(int); // signal handler for SIGCHLD
void sh_alrm(int); // signal handler for SIGALRM

int main(int argc, char **argv)
{
    char buff[BUF_SIZE];
    int n;
    struct sockaddr_in server_addr, client_addr;
    int server_fd, client_fd;
    int len;
    int port;
```

## [Example code] srv.c – with fork () (2/3)

```
/* Applying signal handler(sh_alrm) for SIGALRM */
/* Applying signal handler(sh_chld) for SIGCHLD */

server_fd = socket(PF_INET, SOCK_STREAM, 0);

memset(&server_addr, 0, sizeof(server_addr));
server_addr.sin_family=AF_INET;
server_addr.sin_addr.s_addr=htonl(INADDR_ANY);
server_addr.sin_port=htons(atoi(argv[1]));

bind(server_fd, (struct sockaddr *)&server_addr, sizeof(server_addr));

listen(server_fd, 5);

while(1)
{
    pid_t pid;
    len = sizeof(client_addr);
    client_fd = accept(server_fd, (struct sockaddr*)&client_addr, &len);

    /* 필요한 소스 삽입 (fork() 이용) */

    close(client_fd);
}
return 0;
}
```

## [Example code] srv.c – with fork () (3/3)

```
void sh_chld(int signum) {  
    printf("Status of Child process was changed.\n");  
    wait(NULL);  
}  
  
void sh_alrm(int signum) {  
    printf("Child Process(PID : %d) will be terminated.\n", getpid());  
    exit(1);  
}
```



# Example result

## ① server program 실행

```
kks@sslab-Ubuntu:~/SPLab/practice/2-2$ ./srv 10000
=====Client info=====
client IP: 127.0.0.1
client port: 43257
Child Process ID : 24593
Child Process(PID : 24593) will be terminated.
Status of Child process was changed.
```

## ③ client 연결 확인

- ⑥ SIGALRM (from Child Process)  
→ Child Process의 종료
- SIGCHLD (from Parent Process)

## ② client program 실행

```
kks@sslab-Ubuntu:~/SPLab/practice/2-2$ ./cli 127.0.0.1 10000
> this is test
from server:this is test
> QUIT
kks@sslab-Ubuntu:~/SPLab/practice/2-2$
```

## ④ text 전송 및 수신 확인

## ⑤ 서버에 QUIT 메시지 전달

- ⑦ Server Process 종료 때문에  
read 실패 후 Child process 종료

# Example result

## server

```
kks@sslab-Ubuntu:~/SPLab/practice/2-2$ ./srv 10000
=====Client info=====
client IP: 127.0.0.1

client port: 43586
=====
Child Process ID : 3220
Child Process(PID : 3220) will be terminated.
Status of Child process was changed.
=====Client info=====
client IP: 127.0.0.1

client port: 43600
=====
Child Process ID : 3222
Child Process(PID : 3222) will be terminated.
Status of Child process was changed.
^C
```

## client

```
kks@sslab-Ubuntu:~/SPLab/practice/2-2$ ./cli 127.0.0.1 10000
> this is test1
from server:this is test1
> QUIT
```

```
kks@sslab-Ubuntu:~/SPLab/practice/2-2$ ./cli 127.0.0.1 10000
> This is test2
from server:This is test2
> 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 → Assignment2-2
  - **과제 내용**
    - Introduction
      - 과제 소개 - 4줄 이상(background 제외) 작성
    - Flow chart(강의자료 appendix 참고)
    - Pseudo code(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
- **Assignment2\_2\_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\_2\_A\_2024123456\_홍길동.pdf
  - 압축 파일 명: Assignment2\_2\_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

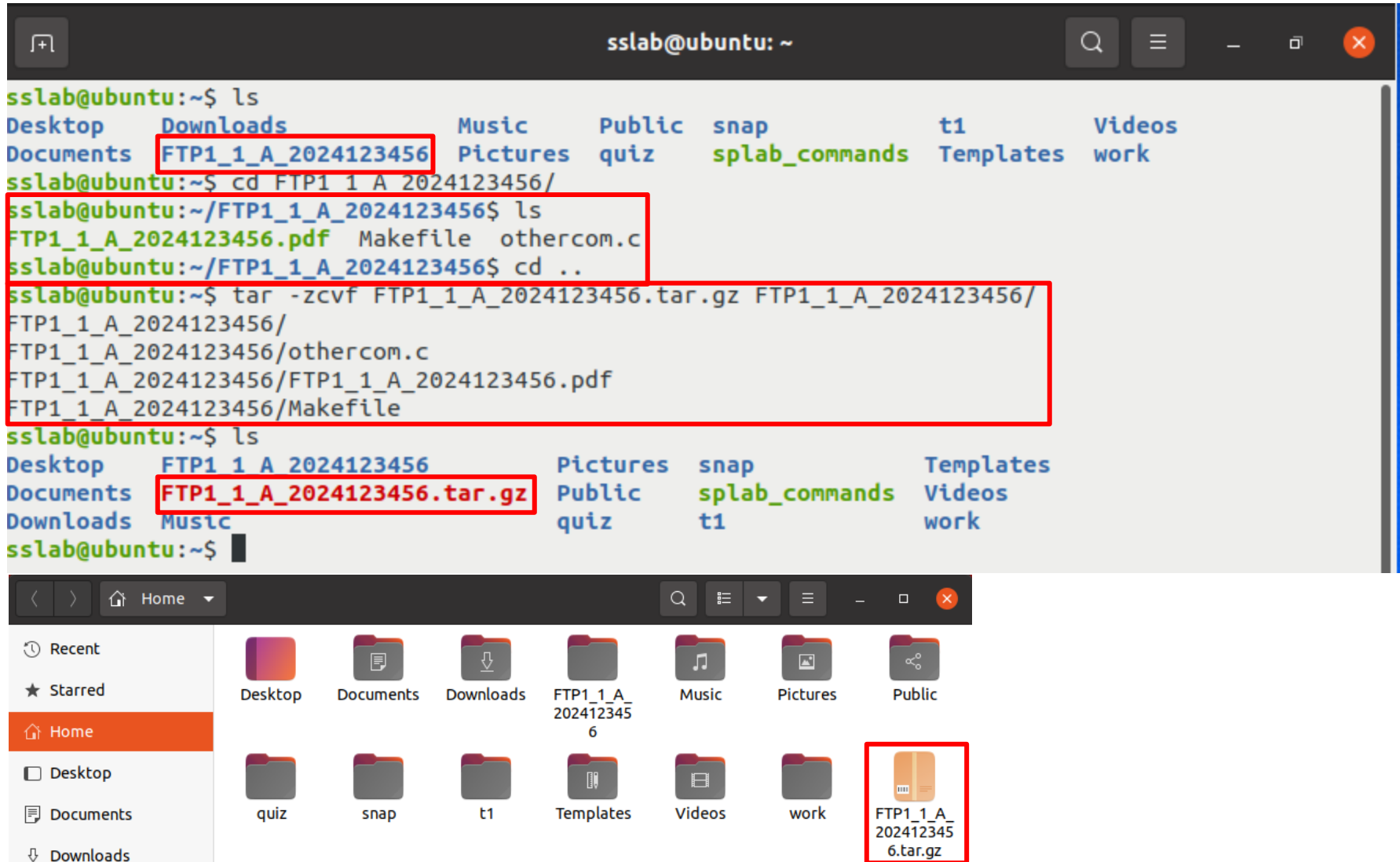
0KB

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

- 과제 제출

- KLAS – 강의 과제 제출
- 2024년 5월 09일 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)

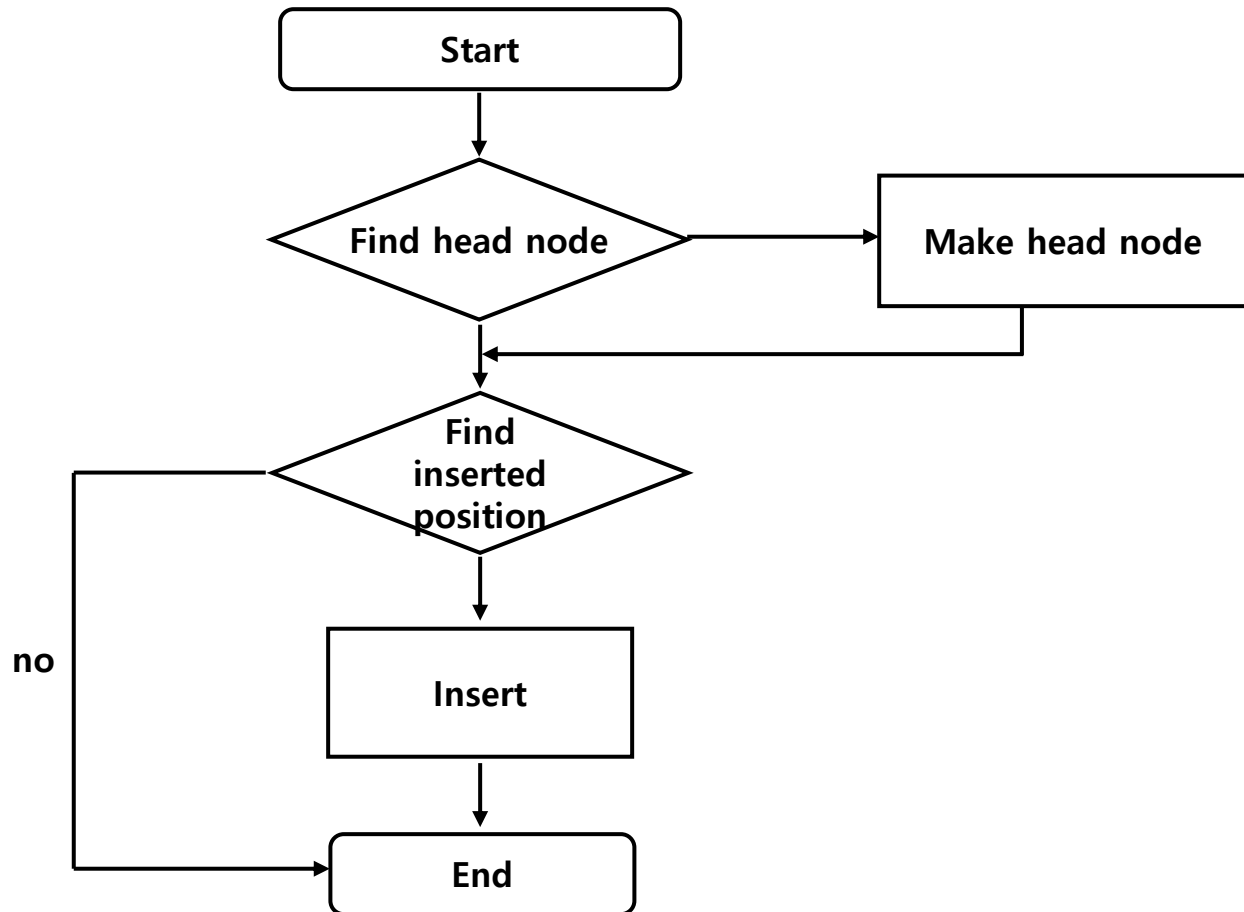
- In-line Comment

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