

2024년 1학기 시스템프로그래밍 & 시스템 프로그래밍 실습

Assignment 1-1

System Software Laboratory
College of Software and Convergence
Kwangwoon Univ.

Contents

- **Assignment1-1**
- **Report Requirements**
- **Appendix A. tar.gz compression**
- **Appendix B. Comment**
- **Appendix C. Algorithm**

Assignment1-1 (1/4)

- **To do list**

- Parameter parsing (`-a`, `-b`, `-c` [`str`]) by using `getopt()`.

- **Work Flow**

- Normally, `getopt()` is called in a loop.
- When `getopt()` returns -1, indicating no more options are present, the loop terminates.
- A switch statement is used to dispatch on the return value from `getopt()`.
- A second loop is used to process the remaining non-option arguments.

Assignment1-1 (2/4)

- Sample Code

```
#include <unistd.h>

int main (int argc, char **argv)
{
    int aflag = 0, bflag = 0;
    char *cvalue = NULL;
    int index, c;
    opterr = 0;

    while ((c = getopt (argc, argv, "abc:")) != -1)
    {
        switch (c)
        { ... /* Write your code */ }
    }
    printf("aflag = %d, bflag = %d, cvalue = %s\n", aflag, bflag, cvalue);

    /*
     * Write your code
     */

    return 0;
}
```

Assignment1-1 (3/4)

- **Result**

\$ testopt

aflag = 0, bflag = 0, cvalue = (null)

\$ testopt -a -b

aflag = 1, bflag = 1, cvalue = (null)

\$ testopt -ab

aflag = 1, bflag = 1, cvalue = (null)

\$ testopt -c foo

aflag = 0, bflag = 0, cvalue = foo

\$ testopt -cfoo

aflag = 0, bflag = 0, cvalue = foo

\$ testopt arg1

aflag = 0, bflag = 0, cvalue = (null)

Non-option argument arg1

Assignment1-1 (4/4)

- Result

\$ testopt -a arg1

aflag = 1, bflag = 0, cvalue = (null)

Non-option argument arg1

\$ testopt -c foo arg1

aflag = 0, bflag = 0, cvalue = foo

Non-option argument arg1

\$ testopt -a -

aflag = 1, bflag = 0, cvalue = (null)

Non-option argument -

\$ testopt -aa

aflag = 2, bflag = 0, cvalue = (null)

\$ testopt -d -a

aflag = 1, bflag = 0, cvalue = (null)

\$ testopt foo foo

aflag = 0, bflag = 0, cvalue = (null)

Non-option argument foo

Non-option argument foo

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 → Assignment1-1
 - **과제 내용**
 - Introduction
 - 과제 소개 - 4줄 이상(background 제외) 작성
 - Flow chart(Appendix 참고)
 - Pseudo code(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 반드시 포함**)
 - kw학번_opt.c
 - Makefile
 - **실행파일 명: kw학번_opt**
 - **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
- **Assignment1_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 Assignment1_1_A_2024123456_홍길동.pdf
 - 압축 파일 명: Assignment1_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

OKB

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

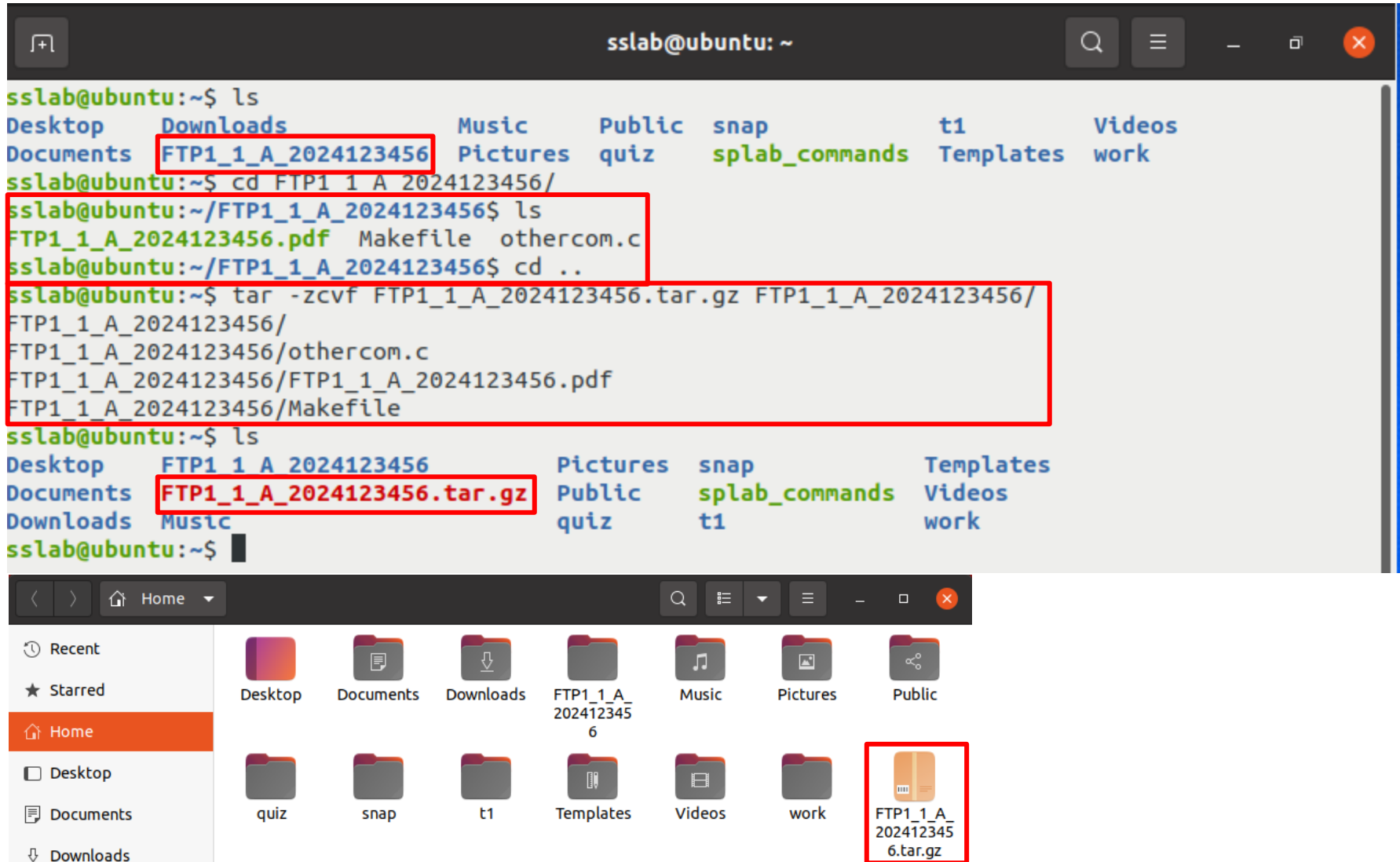
- 과제 제출

- KLAS – 강의 과제 제출
- 2024년 4월 4일 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

System Software Laboratory
College of Software and Convergence
Kwangwoon Univ.

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)

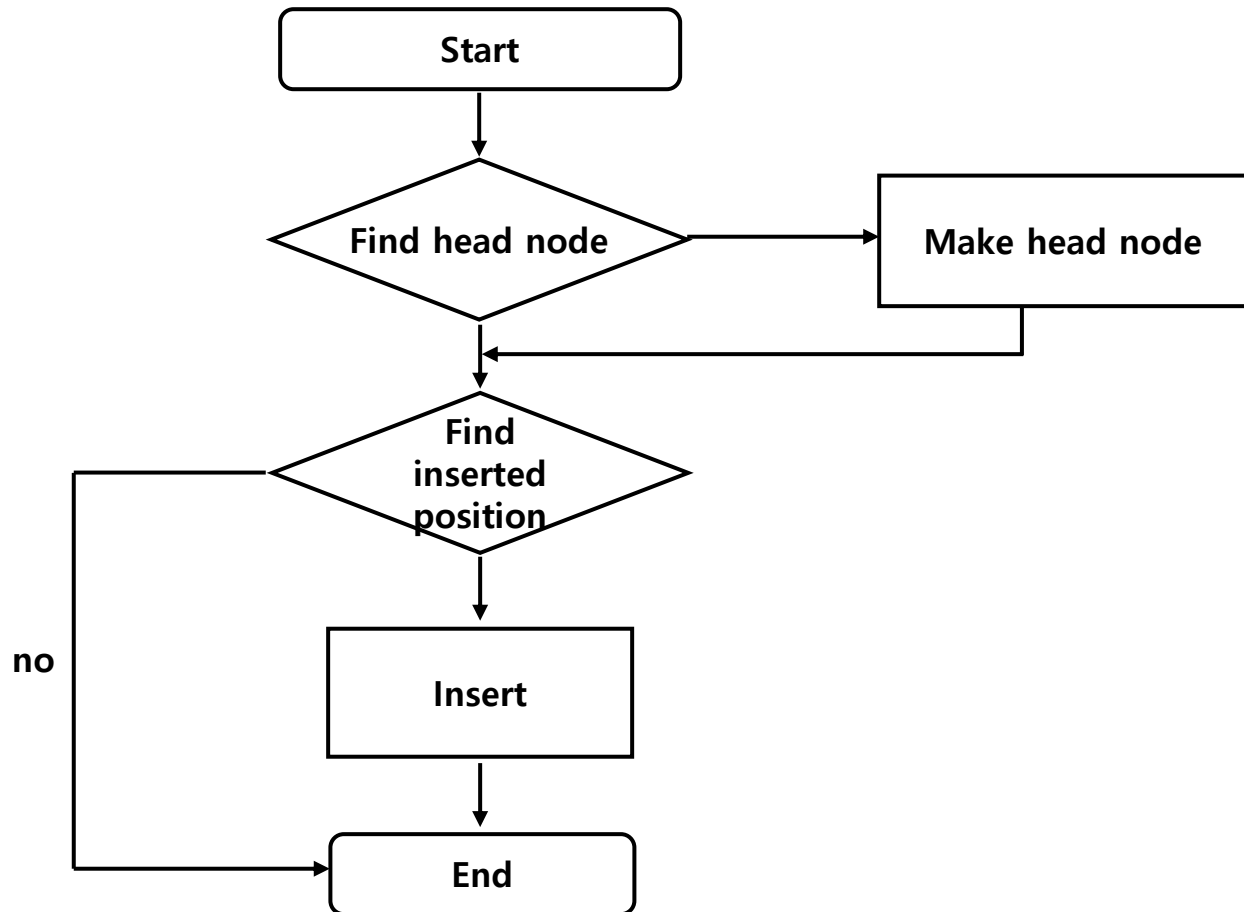
- In-line Comment

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