

Unix 프로그래밍 및 실습

1장. 유닉스 시스템 프로그래밍 개요

강의 내용

- ▶ 강의 내용 (1 ~ 18/27)

- ▶ <http://lily.mmu.ac.kr/lecture/13u2/ch01.pdf>

- ▶ 참고자료 1 (make)

- ▶ http://lily.mmu.ac.kr/lecture/12u2/UNIX_mk.pdf

과제

▶ C 프로그래밍 정리

1. 매개변수로 두 개의 양수를 받아들여 그 사이에 있는 수들의 합을 구하는 함수 작성 (sum_of_nos.c)
 - ▶ `int sum_of_nos(int no1, int no2);`
 - ▶ 예를 들면, 3과 7가 넘어가면 $4+5+6$ 이 결과로 나와야 함
 2. 함수의 기능을 확인하는 간단한 `main()` 작성 (test_ex1_1.c)
 - ▶ 실행 파일 test_ex1_1
 3. 사용자로부터 2개의 수를 받아들여 step 1에서 작성한 함수를 호출하여 결과를 출력하는 `main()` 작성 (ex1_1.c)
 - ▶ 잘못된 입력이 들어오는 오류 처리 (음수가 들어오거나, 정수가 아닌 입력이 들어올 때)
 - ▶ 실행파일 이름 ex1_1
 4. 참고자료1을 참조하여 Makefile 작성
 - ▶ make 이용
- ▶ 프로그램 파일(주석), 실행화면 캡처 포함하여 보고서 작성 후 pdf로 저장
 - ▶ 과제 개요, 수행 과정에 대한 설명 필수
- ▶ 제출기한 : 9월 9일 자정 (<http://cms.mmu.ac.kr/bear>)

Check List

1. Xshell 등 개발환경을 구축했는가?
 1. Xshell 설치
 2. 계정 발급
2. UNIX/LINUX 기본 명령을 자유롭게 활용할 수 있는가?
 1. mkdir, cd, rmdir
 2. cp, mv, rm
3. vi를 이용하여 문서를 자유롭게 작성할 수 있는가?
 1. 기본 명령(모드 전환, 화면 이동 등)
 2. 삭제, 복사 (글자, 줄, 블록 단위)
4. C 언어로 프로그램을 작성할 수 있는가?
 1. if, for, while
 2. printf, scanf
 3. function
5. gcc를 이용하여 컴파일 할 수 있는가?
6. man을 이용하여 주요 명령, 라이브러리, 시스템콜에 대한 정보를 얻을 수 있는가?
7. make를 이용하여 분할 컴파일할 수 있는가?

과제 참고자료 (1)

▶ Makefile

```
1 CC = gcc
2 CFLAGS = -g
3
4 test_ex1_1: ex1_1.h sum_of_nos.o test_sum_of_nos.o
5             $(CC) $(CFLAGS) -o test_ex1_1 sum_of_nos.c test_sum_of_nos.o
6
7 ex1_1: ex1_1.h sum_of_nos.o ex1_1.o
8         $(CC) $(CFLAGS) -o ex1_1 sum_of_nos.c ex1_1.o
9
10 all:
11     make test_ex1_1
12     make ex1_1
```

과제 참고자료 (2)

▶ Ex1_1.h

```
1 #ifndef EX_1_1_H
2 #define EX_1_1_H
3
4 int sum_of_nos(int no1, int no2);
5
6 #endif
```

과제 참고자료 (3)

▶ test_sum_of_nos.c

```
1 #include <stdio.h>
2 #include "ex1_1.h"
3
4 main()
5 {
6     printf("Result(4< ~ <7): %d\n", sum_of_nos(4, 7));
7 }
```

과제 참고자료 (4)

▶ ex1_1.c

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include "ex1_1.h"
4
5 #define BUFFER_SIZE    255
6
7 int main()
8 {
9     int end_of_input = 0;
10    int scan_rt;    /* return value of scanf */
11    int num1, num2;
12    char invalid_str[BUFFER_SIZE];
13
14    printf("Enter two positive numbers: ");
15
16    while (!end_of_input) {
17        scan_rt = scanf("%d %d", &num1, &num2);
18        if (scan_rt == 2)
19            end_of_input = 1;
20        else if (scan_rt == EOF) {
21            perror ("scanf");
22            exit(1);
23        }
24        else {
25            scanf("%s", invalid_str);
26            printf("Enter two positive numbers: ");
27        }
28    }
29
30    printf("Result(%d ~ %d): %d\n", num1, num2, sum_of_nos(num1, num2));
31
32 }
```


강의 내용

- ▶ 강의 내용 (19 ~ 27)

- ▶ <http://lily.mmu.ac.kr/lecture/13u2/ch01.pdf>

- ▶ 참고자료 2 (gdb)

- ▶ http://lily.mmu.ac.kr/lecture/12u2/UNIX_gdb.pdf

과제

- ▶ 본문 예제 수행
 - ▶ [예제 1-4]
 - ▶ [예제 1-5]
 - ▶ [예제 1-6]
 - ▶ [예제 1-7]
 - ▶ 프로그램 파일(주석), 실행화면 캡처 포함하여 보고서 작성 후 pdf로 저장
 - ▶ 과제 개요, 수행 과정에 대한 설명 필수
- ▶ man으로 perror, strerror, getopt 찾아 요약 (man 화면 캡처)
- ▶ gdb로 지난 프로그래밍 과제 결과 디버깅 (주요 화면 캡처)
- ▶ 제출기한 : 9월 23일 자정 (<http://cms.mmu.ac.kr/bear>)