

과제 #1

M1522.006700 확장형 고성능 컴퓨팅 (001)

M3239.005400 데이터사이언스를 위한 컴퓨팅 2 (001)

Due: 2024년 9월 29일(일) 23:59:59

1 (40점) Compilation Process

`sqrt.c` 는 실수 하나를 입력받아 제곱근을 출력하는 간단한 C 코드이다. 해당 코드는 실습 서버 로그인 노드의 `/shpc/skeleton/hw1/sqrt` 디렉토리에 제공된다. 다음 질문들에 답하라.

1.1 Preprocessing

- (a) (5점) `sqrt.c` 는 두 개의 헤더 파일(`stdio.h` 와 `math.h`)을 `include`한다. 하지만 우리는 이 헤더 파일들을 작성한 적이 없다. 이 두 파일은 어디에 있는가? 파일을 찾아서 열어보고, 정확한 파일 경로와 파일의 라인 수를 답하라.

Hint 1: `cpp -v /dev/null`로 C preprocessor인 `cpp`의 설정들을 확인할 수 있다. Hint 2: Linux 의 `find` 명령을 사용해보기.

- (b) (5점) Preprocess까지만 진행하는 `gcc` 옵션을 찾아서 `sqrt.c` 를 preprocess 하라. Preprocess 된 결과에서 `scanf`, `printf`, `sqrt` 를 각각 찾아서 해당 부분을 보고서에 붙여넣어라.

Hint1: `man gcc`로 `gcc`의 옵션들을 확인할 수 있다. preprocessing으로 검색하면 쉽게 찾을 수 있다.

Hint2: `gcc`는 preprocessing 결과를 `stdout`, 즉 화면에 직접 출력한다. 이를 파일에 저장하고 싶다면 `-o` 옵션이나 리다이렉션(`>`)을 사용하자.

- (c) (5점) Preprocess 된 결과에 `scanf`, `printf`, `sqrt` 의 실제 구현이 들어있는가? 다시 말해, `sqrt` 코드에 실제로 제곱근을 구하는 코드가 포함되어 있는가? 포함되어 있다면 코드를 간략하게 설명하고, 포함되어 있지 않다면 포함되어 있지 않은 이유를 답하라.

1.2 Compilation

- (a) (5점) Object file을 출력하는 `gcc` 옵션을 찾아서 `sqrt.o`를 생성하라. 생성하는데 사용한 명령을 답하라.
- (b) (5점) Preprocessing의 결과는 C 코드가 담긴 단순한 텍스트 파일이었다. `sqrt.o`의 파일 포맷은 무엇인가? 파일 포맷과 파일 포맷을 알아낸 방법을 답하라.

1.3 Linking

- (a) (10점) `sqrt.o` 를 아무런 옵션 없이 컴파일을 시도하면 다음과 같은 에러가 발생한다.

```
$ gcc sqrt.o
sqrt.o: In function 'main':
sqrt.c:(.text+0x3d): undefined reference to 'sqrt'
collect2: error: ld returned 1 exit status
```

왜 이러한 에러가 발생하는가? 이유와 함께 올바르게 컴파일하여 최종 실행파일 `sqrt`를 생성하는 명령을 답하라. (이유 5점, 컴파일 명령어 5점)

- (b) (5점) `sqrt`를 실행하여 임의의 수를 입력해보고, 결과 화면을 캡처하여 보고서에 첨부하라.

2 (40점) C Programming

2.1 Shift (10점)

다음 코드를 보고 출력되는 32-bit binary number를 예상하고, arithmetic shift와 logical shift의 차이를 설명하시오.

```
// Function to print the 32-bit binary number
void printBits(unsigned int n) {
    for (int i = 31; i >= 0; i--) {
        printf("%d", (n >> i) & 1);
    }
    printf("\n");
}

int main() {
    int a = -16; // 32-bit signed integer
    unsigned int ua = (unsigned int)a; // Cast 'a' to an unsigned integer

    printf("a : ");
    printBits(a); // Print the 32-bit binary number of 'a'

    printf("Arithmetic Shift (a >> 2) : ");
    printBits(a >> 2); // Print the 32-bit binary number of 'a' after arithmetic right shift

    printf("Logical Shift (ua >> 2) : ");
    printBits(ua >> 2); // Print the 32-bit binary number of 'ua' after logical right shift

    return 0;
}
```

- (a) (2.5점) a의 초기 값인 -16을 32비트 2의 보수표현으로 나타낸 binary number는 무엇인가?
- (b) (2.5점) $a \gg 2$ 의 32-bit binary number는 무엇인가?
- (c) (2.5점) $ua \gg 2$ 의 32-bit binary number는 무엇인가?
- (d) (2.5점) Arithmetic Shift와 Logical Shift의 차이는 무엇인가? 이를 bit 변화를 기준으로 설명하시오.

2.2 Convert (30점)

수 하나를 scanf로 입력받아 이진 표현으로 변환 후 출력하는 프로그램 convert.c 를 작성하라. 뼈대 코드와 Makefile 이 실습 서버 로그인 노드의 /shpc/skeleton/hw1/convert 디렉토리에 제공된다. 뼈대 코드를 이해한 뒤, 주석으로 표시된 부분을 작성하면 된다. 주석으로 표시되지 않은 부분의 코드를 수정하는 것은 불가능하다.

아래는 완성된 프로그램의 실행 예시이다.

[illegible]

때때 코드와 같은 위치에 프로그램의 예시 입/출력이 제공된다. 본인이 작성한 프로그램이 정확한 형식으로 이진 표현을 출력하는지 확인해 보자. 프로그램 채점 시 diff 명령을 이용해 **출력이 정확히 같은지** 검사할 것이다. 따라서 제출 전에 출력 형식을 정확히 지켰는지 확인하고, 디버깅 코드 및 불필요한 출력을 모두 제거했는지 확인하자.

Hint 1: 정수 자료형에 저장된 값을 이진 표현으로 출력하기 위해서는 두 가지 방법이 가능하다. (1) 수업 시간에 다룬 변환 알고리즘을 직접 구현하는 것, 또는 (2) 정수 자료형으로 저장된 값을 형변환을 통해 이진수로 출력하는 법. 각자 편한 방식으로 구현할 것. 채점 시에는 프로그램의 출력만 확인함.

Hint 2: 정수 자료형과 다르게, 실수 자료형에 저장된 값을 이진 표현으로 출력하기 위해 수업 시간에 다른 변환 알고리즘을 **정확히** 구현하는 것은 어렵다. 실수 자료형으로 저장된 값을 형변환을 통해 이진 표현으로 출력하는 방법을 고민해볼 것.

3 (20점) 클러스터 사용 연습

본 수업에서 사용하는 서버 중 실습에 사용되는 서버는 총 13개의 노드 (로그인 노드 1개, 계산 노드 12개) 로 구성되어 있는 클러스터 시스템이다. 홈 디렉토리(/home/s0/shpcXXX)는 모든 노드에 공유된다.

SSH 를 통해 서버로 접속하면 로그인 노드에 접속하게 된다. 계산 노드로의 직접 접속은 불가능하고 slurm 작업 스케줄러를 이용해 실행하고 싶은 작업을 제출해야 한다.

- (a) (5점) 로그인 노드에서 `sinfo` 명령을 실행한 결과를 보고서에 붙여넣고, 해당 명령어와 출력이 어떤 의미를 갖는지 설명하라.
- (b) (5점) 로그인 노드에서 `squeue` 명령을 실행한 결과를 보고서에 붙여넣고, 해당 명령어와 출력이 어떤 의미를 갖는지 설명하라.
- (c) (5점) 로그인 노드에서 `srun -N 2 hostname` 명령을 실행한 결과를 보고서에 붙여넣고, 해당 명령어와 출력이 어떤 의미를 갖는지 설명하라.
- (d) (5점) 로그인 노드에서 `lscpu` 명령과 `srun -N 1 lscpu` 명령을 각각 실행한 결과를 보고서에 붙여넣고, 해당 명령어와 출력이 어떤 의미를 갖는지 설명하라. 두 명령의 출력이 다른 이유는 무엇인가?

4 제출 방법

- 과제 제출은 실습 서버에서 이루어진다.
- 보고서는 pdf 형식으로 만들어 `report.pdf` 이름으로 제출한다. 제출할 `report.pdf` 파일이 위치한 디렉토리에서 `shpc-submit submit hw1 report.pdf` 명령을 실행한다.
- 제출할 `convert.c` 파일이 위치한 디렉토리에서 `shpc-submit submit hw1 convert.c` 명령을 실행한다.
- 파일들이 잘 제출되었는지 확인을 위해 `shpc-submit status` 명령을 실행한다.
- 과제 마감 기한이 지난 뒤 다시 제출 명령을 실행하면 마지막 제출시간이 변경되므로 주의할 것.
- 과제 마감 기한이 지난 뒤 파일이 수정된 경우 `grace day` 를 사용한 것으로 간주한다.

5 주의 사항

- 뼈대 코드를 각자의 홈 디렉토리로 복사해 가 작업하도록 한다.
- 실습용 서버에서 과제를 수행하도록 한다. 소스 코드를 제출하는 과제의 경우 실습용 서버에서 작동하지 않으면 점수를 받을 수 없다.
- 보고서는 간략하게 필요한 내용만 적는다.