"EA0003: 시스템소프트웨어" 실습

|  |  |  |  |
| --- | --- | --- | --- |
| - 실습 번호: lab02 | 분반: | 1 |  |
| - 실습 디렉터리: ~/lab02 |  |  |  |
| - 실습 날짜: 2021년 11 월 8 일 | 학번: | 201810912 |  |
| - 실습 제목: ASM IO 라이브러리 |  |  |  |
| - 실습 내용: | 이름: | 김연출 |  |

1. (실습준비) 자신의 홈 디렉터리 아래에 과제 디렉터리 (~/lab02)를 만든다. 이번 과제에서 만들어지는 모든 파일들은 이 디렉터리에 두도록 한다.



1. (ASM IO 라이브러리) ASM IO 라이브러리의 기능을 확인해보자
   1. ASM IO 라이브러리가 담긴 압축 파일 asm\_io.tar.gz를 실습 디렉터리로 복사한 후, 탐색기에서 이 파일을 더블클릭하면 압축을 해제할 수 있다. 압축이 해제되면I/O 라이브러리 소스 코드 파일과 관련 헤더 파일이 생성된다 (이 파일들은 이후 실습에서도 계속 필요하니 삭제하지 말 것). 생성된 파일들에 대해 파일 이름, 크기, 간단한 설명을 아래에 적으시오.

Asm\_io.asm / 9.0kb / io 라이브러리 소스코드

Ams\_io.inc / 563byte / 디버깅 매크로 라이브러리 소스코드

* 1. (2-2.asm) 압축 파일 asm-io-sample-programs.zip 를 파일에는 입출력 라이브러리를 사용하는 예제 프로그램이 들어있다. 탐색기에서 이 파일을 더블클릭하여 압축을 해제한 후, 생성된 skeleton.asm 파일을 바탕으로 소스 파일 2-2.asm 을 생성하시오. 이 프로그램의. data 섹션에 out\_string 이라는 레이블을 붙여 임의의 문자열을 하나 정의하고, print\_string 루틴을 사용하여 이 문자열을 출력하는 프로그램을 작성하시오. 이 프로그램을 어셈블, 링크하여 실행 파일 2-2을 생성한 후 이를 실행시키는 과정을 담은 화면을 캡쳐하여 아래에 첨부하시오.

텍스트이(가) 표시된 사진

자동 생성된 설명

* 1. (2-3.asm) skeleton.asm 파일을 이용하여, 다음과 같은 프로그램 2-3.asm 을 생성하시오. 이 프로그램은 먼저 read\_int 루틴을 사용하여 정수 하나를 입력 받은 후, 이를 print\_int 루틴을 사용해 출력해야 한다. 이 프로그램을 어셈블, 링크하여 실행 파일 2-3를 생성한 후 이를 실행시키는 과정을 담은 화면을 캡쳐하여 아래에 첨부하시오.

텍스트이(가) 표시된 사진

자동 생성된 설명

1. (산술 연산) 간단한 숫자 계산을 하는 어셈블리 프로그램을 작성해 보자
   1. (3-1.asm) 두 개의 정수를 입력 받아, 두 수의 합을 계산하여 출력하는 어셈블리 프로그램 3-1.asm 을 작성하시오. 단, 이 프로그램은 “반드시” 아래와 같이 동작해야 한다 (예에서 붉은색 글자는 사용자 입력에 해당함). 이 프로그램을 어셈블, 링크하여 실행 파일 3-1을 생성한 후 이를 실행시키는 과정을 담은 화면을 캡쳐하여 아래에 첨부하시오.

% 3-1

Enter a number (A) : 456

Enter a number (B) : 123

A + B = 579

%

텍스트이(가) 표시된 사진

자동 생성된 설명

* 1. (3-2.asm) 위의 3-1.asm 에 두 수의 차를 출력하는 기능을 추가한 어셈블리 프로그램 3-2.asm 을 작성하시오. 단, 이 프로그램은 “반드시” 아래와 같이 동작해야 한다. 이 프로그램을 어셈블, 링크하여 실행 파일 3-2를 생성한 후 이를 실행시키는 과정을 담은 화면을 캡쳐하여 아래에 첨부하시오.

% 3-2

Enter a number (A) : 456

Enter a number (B) : 123

A + B = 579

A - B = 333

%

텍스트, 장치, 측정기, 게이지이(가) 표시된 사진

자동 생성된 설명

## (3-3.asm) 위의 3-2.asm 에 두 수의 곱을 출력하는 기능을 추가한 어셈블리 프로그램 3-3.asm 을 작성하시오. 단, 이 프로그램은 “반드시” 아래와 같이 동작해야 한다. 이 프로그램을 어셈블, 링크하여 실행 파일 3-3을 생성한 후 이를 실행시키는 과정을 담은 화면을 캡쳐하여 아래에 첨부하시오.

% 3-3

Enter a number (A) : 456

Enter a number (B) : 123

A + B = 579

A - B = 333

A \* B = 56088

%

텍스트이(가) 표시된 사진

자동 생성된 설명

* 1. (3-4.asm) 위의 3-3.asm 에 두 수를 나눈 몫과 나머지를 출력하는 기능을 추가한 어셈블리 프로그램 3-4.asm 을 작성하시오. 단, 이 프로그램은 “반드시” 아래와 같이 동작해야 한다. 이 프로그램을 어셈블, 링크하여 실행 파일 3-4 를 생성하고, 이를 실행시킨 결과를 화면 캡쳐하여 첨부하시오.

% 3-4

Enter a number (A) : 456

Enter a number (B) : 123

A + B = 579

A - B = 333

A \* B = 56088

A / B = 3, 87

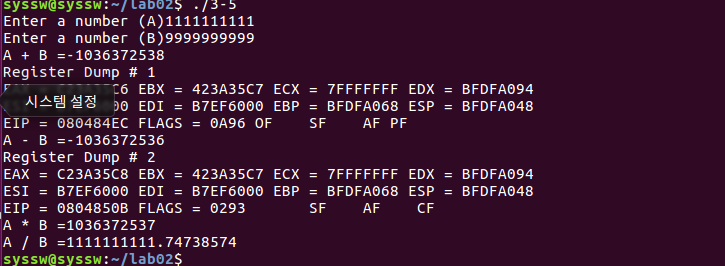
%

텍스트이(가) 표시된 사진

자동 생성된 설명

* 1. (3-5.asm) 위의 3-2.asm 프로그램 중, 각 add / sub 명령 수행 직후마다 dump\_regs 매크로를 사용하여 현재의 레지스터 값들을 출력하는 프로그램 3-5.asm 을 작성하고, 이 프로그램을 어셈블, 링크하여 실행 파일 3-5 를 생성하시오. 프로그램 3-5 를 다양한 입력값을 주어 덧셈과 뺄셈을 수행해 보고, 이 과정에서 CF 비트가 1 이 되는 수행에 대해 화면 캡쳐하여 첨부하고, 왜 CF 플래그가 1 이 되었는지 이유를 설명하시오. (OF 와 SF 가 1 이 되는 경우도 각각 첨부하고 이유를 설명하시오)

1. CF 가 1 이 되는 경우



부호 없는 연산의 결과가 너무 커서 자리올림이 발생할 때 설정된다.

1. SF 가 1 이 되는 경우

텍스트이(가) 표시된 사진

자동 생성된 설명

최상위 비트의 값이 1이 되는 경우입니다 . 결과가 음수일 때

1. OF 가 1 이 되는 경우

텍스트이(가) 표시된 사진

자동 생성된 설명

부호 있는 연산 결과가 너무 많은 비트들을 요구하여 피연산자에 저장되지 못할 때 설정된다.

끝.