EA0003: 시스템소프트웨어 실습

|  |  |  |  |
| --- | --- | --- | --- |
| - 실습 번호: lab05 | 분반: | 1 |  |
| - 실습 디렉터리: ~/lab05 |  |  |  |
| - 실습 날짜: 2021년 월 일 | 학번: | 201810912 |  |
| - 실습 제목: Subprogram |  |  |  |
| - 실습 내용: | 이름: | 김연출 |  |

1. (실습준비) 자신의 홈 디렉터리 아래에 실습 디렉터리 (~/lab05)를 만든다. 이번 실습에서 만들어지는 모든 파일들은 이 디렉터리에 두도록 한다.

텍스트이(가) 표시된 사진

자동 생성된 설명

1. 다음 두 개의 모듈로 이루어진 어셈블리 프로그램에 대해 물음에 답하시오.

main:

**…**

push dword 3

push dword 2

push dword 1

call subpr

add esp, 12

**…**

subpr:

push ebp

mov ebp, esp

sub esp, 8

;

; ①

;

mov esp, ebp

pop ebp

ret

* 1. 위 어셈블리 프로그램의 각 라인에 적절한 설명문을 붙이고, 서브프로그램 subpr 에서 call frame 이 형성된 직후 (① 부분)의 스택의 모습을 그리시오.

|  |
| --- |
| (Parameter)3 |
| (Parameter)2 |
| (Parameter)1 |
| Return address |
| Saved ebp |
| Local variable a (뭔지 아직 모름) |
| Local variable b(뭔지 아직 모름) |

main:

**…**

push dword 3 ; double word 사이즈의 파라미터 3을 push

push dword 2 ;double word 사이즈의 파라미터 2를 push

push dword 1 ;double word 사이즈의 파라미터 1을 push

call subpr ;subpr 함수 호출

add esp, 12 ;파라미터 할당 해제하기 위한 동작

**…**

subpr:

push ebp ;ebp값을 stack에 push

mov ebp, esp ;esp가 stack 의 top이니깐 ebp도 그곳을 가르킴

sub esp, 8 ; 지역변수를 위한 공간 확보

;

; ①

;

mov esp, ebp ;지역변수 할당해줬던 공간 해제

pop ebp ; ebp를pop,stack의 top(esp)은 복귀주소를 가르킴

ret ;호출된 곳의 바로 다음 명령으로 복귀

* 1. (stack.asm) ① 부분에 call frame의 내용을 출력하는 코드를 추가한 프로그램 stack.asm을 작성하시오. 이 프로그램을 어셈블, 링크하여 실행 파일 stack 을 생성하고, 이를 실행시킨 결과를 화면 캡쳐하여 첨부하시오. 또한 소스 코드를 첨부하시오. 첨부된 출력값이 2.1번의 답과 같은지 확인하시오.

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

* 1. 서브프로그램의 시작 시 call frame을 형성하는 부분과 종료 직전 call frame을 해제하는 부분은 각각 단일 명령어인 enter 와 leave 로 대체할 수 있다.
     1. 참고자료 NASM 매뉴얼의 appendix B.4.65 절과 B.4.136절을 참고하여 이 명령어들에 대한 사용 문법, 기능, 동작 과정 등을 조사하여 적으시오.

b.4.65

사용문법:enter imm, imm ( enter amount , level)

기능:Enter 명령어는 stack frame을 구성한다. Ebp를 스택에 push, ebp를 stack frame의 시작주소로 설정(esp와 ebp가 가르키는 곳이 같게끔), 지역변수를 위한 공간을 확보함. 1번째 oprend는 stack에 지역변수를 할당할 공간을 줌 두번째는 중첩 수준을 나타냄(함수 레벨을 나타냄)

동작 과정: push ebp

Mov ebp,esp

Sub esp,지역변수의 양(size\*개수)

b.4.136

사용문법: leave (오퍼랜드 없다)

기능: leave는 stack frame을 종료시킨다. Esp를 원래의 ebp값으로 복구함. 그러고 ebp를 pop

동작과정: mov esp,ebp

Pop ebp

* + 1. (stack2.asm) 위 2.2에서 작성한 프로그램 stack.asm 에서 해당 부분을 enter 와 leave 명령어로 대체한 프로그램 stack2.asm 을 작성하시오. 이 프로그램을 어셈블, 링크하여 실행 파일 stack2를 생성하고, 이를 실행시킨 결과를 화면 캡쳐, 첨부하여 동일하게 동작하는지 확인하시오. 또한 소스 코드를 첨부하시오. 또한 stack.asm 과 stack2.asm 두 프로그램 중 어느 것이 더 효율적인지 생각해보고 아래에 의견을 적으시오.

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명enter 와 leave 명령으로 stack frame을 구성하기 위한 코드를 간소화할 수 있기 때문에 stack2가 사용자 편리성이 좋다고 생각합니다

1. (sum.asm) 다음 조건을 만족하는 어셈블리 프로그램 sum.asm을 작성하시오. 이 프로그램을 어셈블, 링크하여 실행 파일 sum을 생성하고, 이를 실행시킨 결과를 화면 캡쳐하여 첨부하시오. 또한 소스코드도 캡쳐하여 첨부하시오.

* 프로그램 sum 은 1 ~ n 의 정수들의 합을 계산하여 출력하는 프로그램이다. Sum 프로그램은 main 과 calc\_sum 두 개의 모듈로 구성된다.
* main 모듈은 (1) read\_int 를 이용해서 사용자로부터 숫자 n 을 입력, (2) 서브프로그램 calc\_sum 호출, (3) 결과값 출력 의 순서로 수행된다.
* calc\_sum 은 1 ~ n 의 덧셈을 계산하는 모듈이다. calc\_sum 은 main 에 의해 호출되며, 이 때 n 값과 합이 저장될 변수의 주소를 파라미터로 넘겨받는다. calc\_sum 은 수행 중에 두 개의 지역 변수를 사용하도록 한다 (아래 C 함수 참조)

void calc\_sum( int n, int \* sump)  
{  
    int i, sum= 0;

    for (i=1; i<=n; i++){  
        sum = sum + i;

\*sump = sum;  
}

* calc\_sum 호출 시의 calling convention은 반드시 수업 시간에 다룬 내용을 따라야 한다.

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

1. (sum2.asm) 위 2번의 “하나의” 소스 파일 sum.asm 에 정의된 두 개의 모듈 main 과 calc\_sum 을 각각 다른 소스 파일 main.asm 과 sub.asm 에 나누어 작성하시오. 두 어셈블리 소스 파일을 각각 어셈블하여 오브젝트 파일 main.o 와 sub.o 를 생성한 후, 이들을 링크하여 하나의 실행 파일 sum2 를 생성하시오.
   1. 링크 과정에서 실행 파일을 생성하기 위해 사용한 명령줄을 아래에 적으시오.

Gcc -o sum2 main.o sub.o asm\_io.o

* 1. 프로그램 sum2 를 실행시킨 결과를 화면 캡쳐하여 첨부하고, 2번과 동일하게 동작하는지 확인하시오. 또한 소스코드도 캡쳐하여 첨부하시오.

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

* 1. 두 개의 별도의 소스 파일로 나누는 경우 각 소스 코드에 추가되어야 할 어셈블리 지시어가 무엇인지 아래에 적으시오 (지난 실습 Lab2 의 3.2번에서 조사한 내용 참고).

Global : 해당 함수가 정의되어있는 곳에서는 global 지시어를 사용하여 외부에서 사용할 경우 가능하도록 해야한다.

Extern : 해당함수를 사용할 곳에서는 extern 지시어를 사용하여 외부에 있는 함수를 참조한다는 것을 컴파일러한테 알려줘야 한다.

1. (sum3.asm) 위 3번의 프로그램 sum 을 수정하여, n을 프로그램 실행 후 사용자로부터 입력 받는 대신 명령줄에 명시하는 형태로 바꾼 프로그램 sum3 을 작성하고, 이를 실행시킨 결과를 화면 캡쳐하여 첨부하시오. 또한 소스코드도 캡쳐하여 첨부하시오 (즉, 프로그램 sum3은 명령줄에서 “%sum3 10” 의 형태로 실행되어야 한다).

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

끝.