상명대학교 컴퓨터과학과 "EA0003: 시스템소프트웨어" 실습

- 실습 번호: lab03

- 실습 디렉터리: ~/sys\_sw/lab03 이름:

- 실습 날짜: 2016년 5월 일

- 실습 제목: Loop 학번:

- 실습 내용:

1. (과제준비) 자신의 홈 디렉터리 아래에 과제 디렉터리 (~/sys\_sw/lab03)를 만든다. 이번 과제에서 만들어지는 모든 파일들은 이 디렉터리에 두도록 한다.
2. (read\_string.asm) ASM IO 라이브러리에는 문자열을 입력 받는 루틴은 포함되어 있지 않다. 문자열을 입력 받는 프로그램 read\_string.asm 을 작성하시오. 이 프로그램은 (1) 엔터 키가 눌릴 때까지 read\_char 루틴을 반복 호출하여 문자열을 입력 받고, (2) 이를 개행 문자를 제외한 null-terminated 문자열로 저장한 후, (3) 문자열이 제대로 입력, 저장되었는지 확인하기 위해 print\_string 루틴을 사용하여 입력 받은 문자열을 출력해야 한다. 이 프로그램을 어셈블, 링크하여 실행 파일 read\_string 을 생성하고, 이를 임의의 문자열 입력을 주어 실행시킨 결과를 화면 캡쳐 하여 첨부하시오. 단, 아래의 코드를 활용해도 좋으나 입력 문자열의 길이가 저장 공간의 크기를 넘어가지 않도록 수정하여 사용하시오.

**…**

segment .bss

string resb 32 ; (마지막 null 문자를 포함한) 입력 문자열의

; 최대 길이

segment .text

**…**

mov edx, 0

repeat: call read\_char

mov byte [string+edx], al

inc edx

cmp al, 0x0a ; 엔터가 눌릴 때까지

jne repeat ; 반복 입력

mov byte [string+edx-1], 0 ; 마지막 개행문자는

; null 문자로 대체

**…**

1. (atoi.asm) 사용자로부터 십진수 문자만으로 이루어진 문자열(‘0’ ~ ‘9’로만 이루어진 문자열)을 입력 받고, 이를 해당 정수로 변환하는 프로그램 atoi.asm 을 작성하시오. 이 프로그램을 어셈블, 링크하여 실행 파일 atoi 를 생성하고, 이를 실행시킨 결과를 화면 캡쳐 하여 첨부하시오.

* ASM IO 라이브러리의 read\_int 루틴은 사용할 수 없다; 문제 1번과 같이 반드시 read\_char 를 반복 호출하는 형태이어야 함.
* 십진수 1234 의 경우, 1234 = 1x103 + 2x102 + 3x101 + 4x100 = (((((0+1) x 10) + 2) x 10) + 3) x 10 +4 임을 활용한다.
* 문자 ‘0’의 ASCII 코드 값은 0x30, 문자 ‘1’의 값은 0x31, …, 문자 ‘9’의 값은 0x39 이다.
* 숫자 변환이 제대로 되었는지 확인하기 위해 print\_int 루틴을 사용하여 계산된 숫자값을 출력해 보도록 한다.

1. (bin2dec.asm) 문제 2번의 atoi.asm 을 수정하여, 하나의 이진수 문자열을 입력 받은 후 이에 해당하는 십진수 값을 출력하는 프로그램 bin2dec.asm 을 작성하시오. 이 프로그램을 어셈블, 링크하여 실행 파일 bin2dec 을 생성하고, 이를 실행시킨 결과를 화면 캡쳐 하여 첨부하시오.

* 문제 2번의 변환 과정 중 10을 곱하는 대신 2를 곱하면 됨
* 숫자 변환이 제대로 되었는지 확인하기 위해 print\_int 루틴을 사용하여 계산된 숫자값을 출력해 보도록 한다.

1. (mult.asm) 두 수를 입력 받아, 덧셈을 반복하는 방식으로 두 수의 곱을 계산하여 결과를 출력하는 프로그램 mult.asm 을 작성하시오 (즉 2 x 4 = 2 + 2 + 2 + 2 형태로 계산). 이 프로그램을 어셈블, 링크하여 실행 파일 mult 을 생성하고, 이를 실행시킨 결과를 화면 캡쳐 하여 첨부하시오.

* 곱하는 두 수의 입력은 read\_int 루틴을 사용한다. 입력 받은 두 수는 일단 .bss segment 에 저장 후 계산에 사용하도록 한다.
* 결과값의 출력은 print\_int 루틴을 사용
* 루프는 반드시 loop 명령을 사용하여 구현

끝.