

컴퓨터구조 (2018년 2학기)

과제 #1 – MIPS Procedure Call 구현

0. 과제 제출 및 유의 사항

- A. 수업 홈페이지에 링크된 과제 제출 사이트에 제출 기한에 맞춰서 온라인으로 제출한다. 제출 사이트에는 다른 과목들의 과제들도 같이 제출할 수 있도록 되어 있으므로 해당 과목과 과제를 정확하게 선택하도록 주의한다.
- B. 소스 코드와 MS 워드로 작성된 보고서의 pdf 파일들을 하나의 zip 파일로 묶고 파일 이름은 "이름_hw1.zip"으로 한다. 보고서에 코드의 실행 방법을 자세히 명시해야 한다. 사용 요령을 명확하게 명시하지 않아 실행이 되지 않은 코드의 경우 과제를 0 점 처리한다.
- C. 제출 마감 시간을 기준으로 24 시간씩 지날 때마다 만점에서 25%씩 감점처리한다.

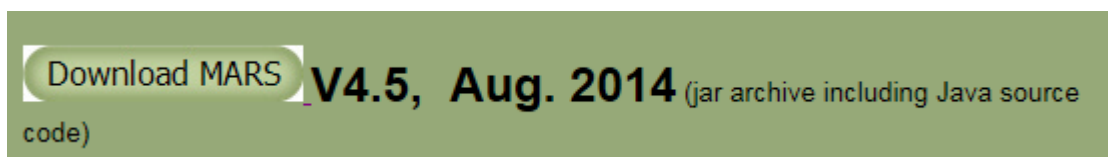
1. Mars설치

MIPS 프로그래밍을 하기 위하여 MIPS 시뮬레이터를 다운받아야 한다. 시뮬레이터를 다운받을 수 있는 사이트의 주소는 아래와 같다.

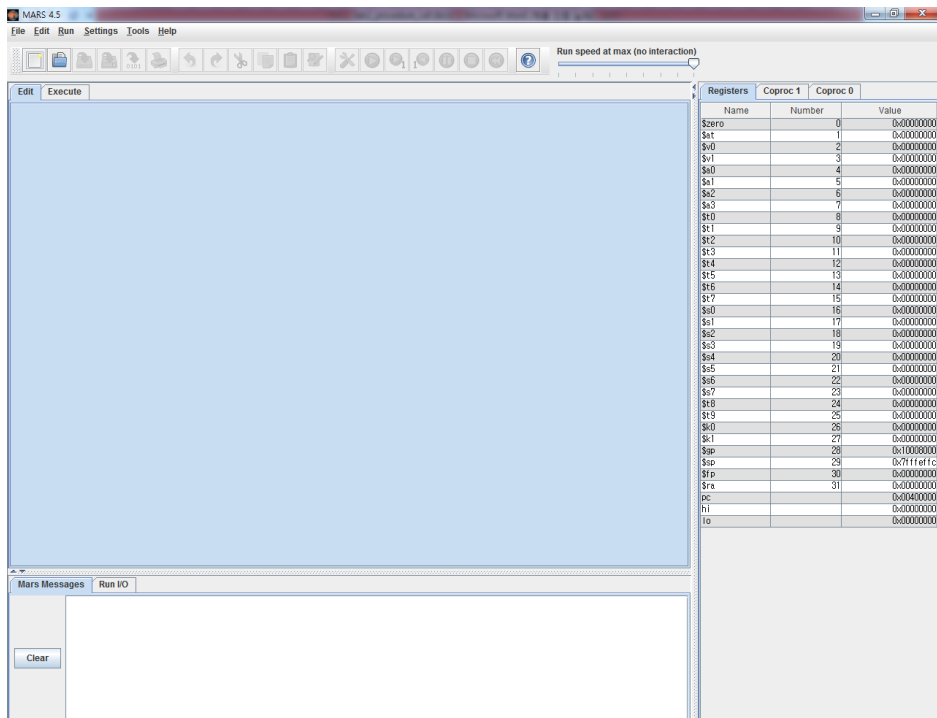
<http://courses.missouristate.edu/KenVollmar/MARS/>



위와 같이 사이트가 나오면 Download를 클릭하여 Mars를 다운로드한다.



다운로드한 'Mars4_5.jar' 를 double click 하면 MARS가 실행된다.



2. 개요

첫 번째 실습은 수업 시간에 배웠던 MIPS assembly를 이용하여 procedure call을 작성해 보는 것이다. 수업 홈페이지에 링크된 lab1_1_template.asm은 아래와 같다.

```
1      .data
2      n: .word 5
3      .text
4      lb      $a0, n
5      jal      sum
6      add      $v1, $v1, $v0
7
8      exit:
9      li      $v0, 10
10     syscall
11
12     #put your code here
```

위의 코드에서 라인 1-2는 data선언부를 나타낸다. 그리하여 n에 5의 값을 입력한 변수를 선언한다. 라인 3은 코드의 실행부분임을 나타낸다. 라인 4는 n의 값을 \$a0에 저장하는 것을 의미한다. 5번째라인은 sum함수를 호출하고 \$ra = pc+4를 하는 것을 의미한다. 6번째 라인은 종료를 위해 \$v0의 값을 \$v1에 옮기는 것을 의미한다. 라인8-10은 프로그램을 종료하는 부분이다.

12번째 라인부터 우리가 작성해야 할 코드 부분에 해당한다.

```
1      .data
2      s1: .asciiz "hello"
3      s2: .asciiz "hello"
4      .text
5
6      la      $a0, s1
7      la      $a1, s2
8      jal      streq
9      add      $v1, $v1, $v0
10
11     exit:
12     li      $v0, 10
13     syscall
14
15     #put your code here
```

위의 코드에서 라인 1-4는 data선언부를 나타낸다. 라인 6-7은 \$a0과 \$a1에 s1과 s2의 주소를 옮긴다. 라인 9또한 종료를 위하여 \$v0의 값을 \$v1에 옮긴다. 라인 11-13은 프로그램을 종료하는 부분이다.

15번째 라인부터 우리가 작성해야 할 코드 부분에 해당한다.

3. 실습 내용

- A. 다음의 C 코드를 MIPS 어셈블리로 변환하여라. (lab1_1_template.asm 참고)

```
int sum(int n) {  
    return n ? n + sum(n - 1) : 0;  
}
```

- B. 아래의 함수 streq()는 두 개의 char 타입 포인터 인자(s1, s2)가 동일한 문자열을 가리키고 있으면 1을 아니면 0을 돌려주는 함수이다. (lab1_2_template.asm 참고)

```
int streq(char *s1, char *s2) {  
    do {  
        if(*s1 != *s2) {  
            return 0;  
        }  
    } while(*s1++ && *s2++);  
    return 1;  
}
```

함수 streq()를 MIPS 어셈블리 코드로 변환하여라.

4. 프로그램 compile/build 및 실행

Lab1_1_templat.asm 을 프로그래밍 한 후 이를 수행하기 위해서는 우선 파일을 저장한 후, F3 키를 눌러 Assemble과정을 거쳐야 한다. Assemble이 완료된 후에는 F5키를 눌러 프로그램을 실행한 후, Register의 값을 확인한다.

(Lab1_2_templat.asm 또한 동일하다.)

5. 보고서에 작성할 항목들

- A. recursion, do-while, stack을 이용한 register backup 등의 구현 세부 사항들에 대해 간결하고 명확하게 설명함.
- B. 프로그램의 수행 결과를 확인하기 위해 프로그램 수행 후 레지스터의 내용을 보여주는 screen capture를 보고서에 포함해야 함
- C. screen capture를 사용할 때에는 내용을 명확하게 파악할 수 있어야 함