CS&A: Lab Sessions

Exercises 2nd session: MIPS

Ruben Van den Bossche

1BA INF - 2010-2011

1 Time Schedule

Exercises are made individually. Fill in all solutions to the exercises in the file `oefeningen.html`. **Include all .asm files that contain your MIPS programs.**

Put all your files in a tgz archive, as explained on the course's website, and submit your solution to the exercises on Blackboard.

- Deadline: **August, 21 2011, 23u55**

2 Exercises

Write a MIPS program for the MARS simulator for each of the following exercises. As always, document your solution well (use #).

1. Write a MIPS program that pushes and pops integers on the stack. Execute this sequence of push and pop operations:

```
push 7
push 15
pop
push 31
pop
push 63
pop
pop
```

2. (a) Write a MIPS program that reads an integer n (using a syscall), after which it reads n integers (using syscalls), and stores them in an array. Because you don't know the size of the array in advance, you will have to allocate space for it on the heap (*Hint: use syscall 9 for* `sbrk`).

   (b) Add a (leaf) procedure that prints an array. The procedure has two parameters: the address of the first element of the array and the number of elements in the array. Call the procedure with the array on the heap.

(c) Add a (leaf) procedure that sorts an array. Call the procedure with the array on the heap. Implement the algorithm from the Oberon-procedure below.

```
PROCEDURE Sort*(a : ARRAY OF INTEGER);
    VAR
        temp : INTEGER;
        i, j, minindex : LONGINT;
BEGIN
    FOR i := 0 TO LEN(a) - 1 DO
        (* zoek kleinste in de rest van de array *)
        minindex := i;
        FOR j :=  i + 1 TO LEN(a) - 1 DO
            IF( a[j] <  a[minindex] ) THEN
                minindex := j;
            END;
        END;

        (* verwissel waarden *)
        temp := array[i];
        array[i] := array[minindex];
        array[minindex] := temp;
    END;
END Sort;
```

3. Write a MIPS program that reads an integer n (using a syscall), and calculates the fibonacci numbers from 1 to n. Use a recursive procedure! The fibonacci numbers are defined as follows:
   $F_0 = 0$
   $F_1 = 1$
   $F_i = F_{i-2} + F_{i-1}$ for $i > 1$