

Computer Architecture
CPSC 32, Fall Semester 2003
Lab Assignment #2

Due: One week after your lab session – complete by yourself.

1 Assignment

[35 points] Write a MIPS assembly procedure `strtoi` that has two input parameters: `$a0`, containing the beginning of a NUL-terminated ASCII string; and `$a1`, an integer containing the base. The procedure returns in `$v0` the string converted to an integer using the base given in `$a1`. You can assume that the integer can be represented by a 32bit signed integer, and that the string is well-formatted, that is, does not contain any letters outside the range. For example, if the base is 16, then the string will contain letters from `[0-9a-f]`, possibly preceded by a minus sign.

Embed this procedure into a program that prompts the user to input a string and a base, and prints the converted integer.

[35 points] Write a procedure `substr` that takes three arguments `$a0`, `$a1`, and `$a2`. The register `$a0` contains the address of a NUL-terminated string, the register `$a1` containing an integer defining the beginning of a substring, and `$a2` a register containing the length of the substring. The procedure should return in `$v0` an address to a NUL-terminated string, consisting of the substring defined by these arguments.

For example, suppose that `$a0` contains the beginning address of the string “abcdef”. If `$a1` contains 1, and `$a2` contains 3, then the procedure should return the address of the beginning of a string containing “bcd”.

Embed this procedure into a program that prompts the user to input a string, and the two integers specifying the beginning and length of the substring; the program should print the substring.

Hint: Study the system call `sbrk`. You give just one argument containing the number of bytes you want, and the procedure `sbrk` will return an address of memory of a newly allocated memory field.

[30 points] Write a procedure `mss` that takes two arguments, `$a0` containing the address of an array `A[0..n-1]` of integers. The length n of the array is given in `$a1`. The procedure returns in `$v0` an integer describing the maximum sum of any contiguous subsequence of `A`, that is, it should return

the number

$$\text{mss}(A) = \max \left\{ \sum_{k=i}^{j-1} A[k] \mid 0 \leq i \leq j \leq n \right\}.$$

Note that you can find $\text{mss}(A)$ in linear time. The following program sketches a solution in pseudocode:

```
maxsum = 0;
maxsuf = 0;
for  $i = 0$  to  $n - 1$  do
    maxsuf = max{0, maxsuf+A[i]};
    maxsum = max{maxsum, maxsuf};
od;
return maxsum;
```

You can specify the array and its length in the data segment; there is no need for any user interface. The main procedure should call `mss` after loading the beginning address of the array into the `$a0` and the length into register `$a1`.

Explain in the comments why this procedure calculates the correct result.

Important. Ensure that you document in all programs the usage of registers, explain the functionality by an equivalent program in C or C++, and have detailed documentation of nearly every line of code.

2 Dishonesty

Make sure that you complete the assignment by yourself. Do not copy the code from others, nor provide others with your code. Refrain from copying and modifying the code from other sources.