CS220: Introduction to Computer Organization

Assembly Programming

17 Oct 2011

SPIM/MIPS code due on: Nov 6^{th} 2011, 11:59PM

Write the following programs in SPIM/MIPS assembly language. Use SPIM simulator (`http://pages.cs.wisc.edu/~larus/spim.html`) to test your programs. In each case the input should be read from screen and output should be printed on screen.

1. `primes.asm`: Reads an integer n and prints all prime numbers from 2 to n. Define a function `isPrime` to test if an integer is prime, and use it.

2. `fibonacci.asm`: Reads an integer n and prints n^{th} fibonacci number Fib(n). Use iteration to compute fibonicci numbers, and store all intermediate fibonacci values in a global array. (NOTE: Use **sbrk** system call to allocate memory for global array.)

3. `octal.asm`: Write a function that reads a string and checks if it is a valid octal number or not. The value in the string is a valid octal value if it contains characters from '0' to '7' only. If the string contains a character outside the range '0' to '7', the string will not be considered a valid octal value.

   The return value of the function is 0 if the string is **not** a valid octal number. It is 1 if the string is a valid octal number.

4. `substr.asm`: Write a program that takes as input two strings. Lets refer to the first string as A and the second string as B. Then search whether B is present in string A. If B is present in A, delete all occurrence of B from A. Do this repeatedly till there is no occurence of B in the result. Show the resulting string at the output.
   Examples:

   - A = 10111101 and B = 011 then result = 11101
   - A = 01010010 and B = 010 then result = 10
     Note that B is detected in A from left to right and the first occurrence of B is deleted first. Hence A = **010** 10 **010** $\Rightarrow$ result= 10 is the correct result. On the other hand, A = 01 **010 010** $\Rightarrow$ result = 01 is wrong.
   - A = 10010101 and B = 010 then result=11
     A = 10 **010** 101 $\Rightarrow$ 1 **010** 1 $\Rightarrow$ 11

5. `hanoi.asm`: Write a program to solve the problem of *Tower of Hanoi*[1].

   You have to assume that the three rods are named **A, B,** and **C**. The program reads an integer n, and prints the movements to move all n disks from rod A to rod C, using B as the place-holder.

   The movements have to be printed in the following form (this is for $n = 3$):

```
Move from A to C.
Move from A to B.
Move from C to B.
Move from A to C.
Move from B to A.
Move from B to C.
Move from A to C.
```

6. `recur.asm`: Consider the following recursive function:

$$\begin{aligned} A(m,n) &= n+1 \text{ if } m == 0 \\ A(m,n) &= A(m-1,1) \text{ if } m > 0 \;\&\&\; n == 0 \\ A(m,n) &= A(m-1, A(m,n-1)) \text{ if } m > 0 \;\&\&\; n > 0 \end{aligned}$$

   Write a program that takes as input integers m and n. If either of m or n is negative, it prints an error. Otherwise, it computes $r = A(m,n)$. Further, it counts the number of (recursive) calls c to the function A. The output of the program is the value r and the count c.

[1] If you do not know what *tower of hanoi* is, refer to wikipedia article at `http://en.wikipedia.org/wiki/Tower_of_Hanoi` for the description and the rules of puzzle.