The UNIVERSITY *of* NORTH CAROLINA *at* CHAPEL HILL

Comp 411 Computer Organization
Spring 2011

Lab #6: String Processing and Recursion
Issued Fri. 2/18/11; Due Fri. 2/25/11 (beginning of lab)

This assignment consists of three parts. For the first two parts, you will be writing functions to be called by other assembly programs. You ***must*** manage the stack appropriately by following all the conventions discussed in class regarding saved and temporary registers, passing arguments and values, etc. Your final step in this assignment is to combine these functions into one program.

Exercise 1. ASCII string to binary number (atoi.asm)
Write a function `atoi` that converts a string of up to 8 ASCII digits into a 32-bit integer. The function will receive as an argument the starting address of the string and must return a 32-bit integer containing the integer value of the string. Assume that the string is an ASCIIZ string, i.e., ends with the null character (ASCII code 0). You don't need to check for errors in the string, i.e., you may assume the string contains only characters '0' through '9', and will not represent a negative number or too large a number. For example, `atoi` called with the argument "12345" will return the integer 12345.

Exercise 2. Computing the factorial of a number (factorial.asm)
Write a ***recursive*** function `factorial` that computes the factorial of N, i.e., N! The recursive definition of factorial is: factorial(0)=1, factorial(N)=N*factorial(N-1). This function takes in a single unsigned 32-bit integer and will return an unsigned 32-bit integer result. You can assume that the function will be called only with an argument small enough so that the result does not overflow, i.e., fits within 32 bits (unsigned). A non-recursive implementation of the function will receive *zero credit*.

Exercise 3. Put it all together (complete.asm)
Once you have created complete and correct functions for Exercises 1 and 2, incorporate them into the provided assembly file for part 3. This main procedure should call `atoi` on the provided string in order to convert the string value to an integer, then call your `factorial` function to compute the corresponding factorial. Finally, the main procedure will print out the result.

Submission: Starter assembly programs are provided for you on the class website. Fill in your name and onyen in the comments at the top. Add your code where indicated; do not change anything else in the file! When you have completed your assignment, submit copy of the code (all three files) and a picture of the program's output (for all three parts).

Grading: Exercise 3 will be graded first, and if it is correct, you will receive full credit for this lab. If incorrect, Exercises 1 and 2 will be graded separately (40 points each). Points will be deducted for not strictly following the conventions for writing procedures (i.e., managing activation records properly on the stack, and following register usage conventions).