CSE 378: Machine Organization and Assembly Language

Assignment #4: Anagrams

(Assigned October 20, 2000; Due October 27, 2000 by 5:00)

1 Anagram

Anagrams are words or phrases created by rearranging the letters in another word or phrase. For example, satin is an anagram for stain. For this assignment you will write a program in MIPS assembly that produces all possible anagrams for a given word. For example, if you were given the word 'cat', the program would produce the following output: cat, cta, act, atc, tca, tac.
The algorithm used to solve this problem is fairly complicated, so we have provided you with a C program that does the required operation. If you are up to a challenge, I suggest that you attempt to solve the problem by yourself before you look at the sample code.
The source code consists of three functions. The 'anagram' function takes a string and calls 'anagramRecursive' which recursively prints all of the possible combinations of the letters. The third function, 'almoststrcpy', is a small variant of the standard C function 'strcpy' except that it doesn't copy the i'th letter. You should take some time to understand how this program works before you start coding it in assembly (run through it on paper with something simple like 'cat').

2 Requirements

You should turn in a file called 'hw4.s' that contains three functions: 'anagram', 'anagramRecursive' and 'almoststrcpy'. These functions should take *exactly* the same arguments as their counterparts in the C program. In other words the functions should have the following parameters:

<u>anagram:</u>
$a0 – address of a string to be anagrammed
<u>anagramRecursive:</u>
$a0 – address of a string that contains the letters not yet used in the anagram string
$a1 – address of the current anagram string
$a2 – pointer into the anagram string where the next character is to be inserted
<u>almoststrcpy:</u>
$a0 – address of string to copy characters into
$a1 – address of string to copy characters from
$a2 – index of character that is *not* copied from one string to the other

As usual you should have a comment at the top of the file with your name, student number and section. You should also be careful to comment your code well. Note that this code is very recursive, so you have to be very, very careful about your call stack setup when you are creating anagramRecursive so that you don't overwrite important registers.