Lab Exercise 2

Due Friday May 11

Part 1.

Now that you are familiar with QTSPIM and have successfully modified code, you will create a small program with 5 options.

The program should prompt the user indicating what the 5 options are and ask which option the user wishes to perform. The user can enter the option via a number.

NOTE: if one of the options available is not entered the program should state that the input of the user is invalid! For example if I enter option 9…Then the program should state that this is not a valid option and prompt the user again for an input showing all the options!

The options should be as follows: NOTE: All options should be written in a procedure!

1. Add two numbers.

2. Multiply four numbers.

3. Read a string store it in the memory, compute and print the frequency of each character and then print the reversed string.

4. Write a recursive procedure that prints the Fibonacci sequence up to the number input by the user. NOTE: If for example I enter 10 then you should print: 0, 1, 1, 2, 3, 5, 8 The next number would be 13 and it is larger then 10 thus it would not be printed out and it should end the Fibonacci procedure.

5. Exit the program… exit the program with the proper syscall code

When the user enters an option let’s say option 1. The program should acknowledge the users request by stating what it is about to do. Thus it should print something such as “Option 1 adding two numbers” or something along those lines. Next the program should request two numbers from the user. The program must state what it is doing at all times. Hence print for example” Please enter the first number to be added:” after the second number is input the program must state what the output is such as” The sum is: ” Do not just output the result.

Finally when the user specific option has completed the program should again prompt the user for the next option to be completed.

Part 2.

The program you created was really a cover program that allows you to be identified by the U.S.A. in case of your capture. Thus there is a secret option “264”. If this option is input the program should print out the following: ‘input 264?” If the user enters the proper password “Davarpanah” then you will continue to part 3. Otherwise you will print out that the option was invalid and prompt the user with the options from step 1.

Part 3.

If the proper password was input the program should request “number of bytes to allocate?” The number input by the user will be the number of bytes dynamically allocated in memory. After the memory has been dynamically allocated in the heap it should print out the first address returned by the syscall and store your first and last name over and over again until the dynamically allocated memory is full.

After this the program should print out “Identity in HEAP” and terminate the whole program.

Example:

The program will ask…

*number of bytes to allocate?*  50

The program will print out…

*The memory was allocated at address 0x30000000*

At this point you should fill the 50bytes of data with your first and last name over and over again until it is full. In my case it will have starting @ address 0x30000000 nimadavarpanahnimadavarpanah... until the memory is full. In your case it should be your first and last name looped over and over again.

The program will print out…

*Identity in HEAP*

The program should terminate