postlab2

Wednesday, February 2, 2022   2:34 PM

Post-lab Assignment 2

Due: Fri, 28 Jan 2022 14:20:00 (approximately 5 days ago)
[0 points possible] [200 penalties possible] [100 penalties graded so far]
Submit your materials for lab 2 here.

Academic Honesty Statement [0 ... -100 points]

By typing my name, below, I hereby certify that the work on this lab is my own and that I have not copied the work of any other student (past or present) while completing it. I understand that if I fail to honor this agreement, I will receive a score of zero for the lab, a one letter drop in my final course grade, and be subject to possible disciplinary action.
I also understand that I am to use only my own hardware for this and every other lab and not share it with any other student. I have used only my own hardware for this lab (except when explicitly permitted otherwise by the course staff). If a discovery is made otherwise, I will receive a zero for this lab.

Tzu Yu Chen

Instructions

Create a new project in SystemWorkbench called "lab2" in a similar way in which you created lab0 and lab1. Remove the main.c file that is automatically created. In its place, create a "main.s" file to hold the assembly language subroutines you write for this lab experiment.

Two C subroutines, named **`intsub`** and **`charsub`** are described below. Each one refers to a global array and maybe an global variable. Implement each one in ARM Cortex-M0 assembly language with exactly the subroutine names and variable names specified in the provided C code.

You should test your code with an assembly language main subroutine that appears as follows:

```
.text
.global login
login: .string "xyz" // Make sure you put your login here.
.balign 2
.global main
main:
    //bl autotest
    bl intsub
    bl charsub
    bkpt
```

When you are satisfied with the operation of your subroutines, you can insert a **`bl autotest`** to check your subroutines, report on their correctness, and generate a completion code for the lab experiment. The autotest.o compiled object file is linked below. Incorporate it into your project has you did before for labs 0 and 1.

When you construct the subroutines, you should take care to modify only the R0 - R3 registers, in keeping with the ARM Cortex-M0 Application Binary Interface (ABI). This is important because the subroutines will be called multiple times by the autotest subroutine. If you must use more registers than R0 - R3, you should use PUSH and POP to properly save the extra registers on subroutine entry and restore them on subroutine exit.

The C problem specification:

```
// The global variables used for this program.
// autotest may change any of these data before invoking
// your subroutines on them.
int arr[]={ 23, 11, 23, 27, 19, 14, 13, 13, 13, 12, 17, 24, 21, 16, 19, 22 };
int value = 0;
//char str[] = "Hidden+ , messages - 01234 in . this % string!";
char str[] = "jKfFgP+ , oGuUcIgU - 01234 Kp . VjKu % UvTkPi!";

void intsub()
{
    for(int i = 0; i < 15; i++) {
        if ((value >> 4) < 5) { // That's an ASRS instruction.
          value += 2 * arr[i] + arr[i+1];
        } else {
          value -= 3 * arr[i];
        }
        arr[i] = 2 * arr[i] + arr[i+1];
    }
}

void charsub()
{
    for(int x=0; str[x] != '\0'; x=x+1) {
        if (str[x] >= 'a' && str[x] <= 'z') // 'a' == 0x61, 'z' == 0x7a
            str[x] = str[x] & ~0x20;
        if (str[x] >= 'A' && str[x] <= 'Z') // 'A' == 0x41, 'Z' == 0x5a
            str[x] = str[x] - 2;
    }
}

const char login[] = "xyz";
void main()
{
  //autotest();
  intsub();
  charsub();
}
```

The results for the data given to you should be:

```
iteration 0: value = 57, arr[0] = 57
iteration 1: value = 102, arr[1] = 45
iteration 2: value = 33, arr[2] = 73
iteration 3: value = 106, arr[3] = 73
iteration 4: value = 49, arr[4] = 52
iteration 5: value = 90, arr[5] = 41
iteration 6: value = 51, arr[6] = 39
iteration 7: value = 90, arr[7] = 39
iteration 8: value = 51, arr[8] = 38
iteration 9: value = 92, arr[9] = 41
iteration 10: value = 41, arr[10] = 58
iteration 11: value = 110, arr[11] = 69
iteration 12: value = 47, arr[12] = 58
iteration 13: value = 98, arr[13] = 51
iteration 14: value = 41, arr[14] = 60
str is 'HIDDEN+ , MESSAGES - 01234 IN . THIS % STRING!'
strlen(str) = '46'
```

You should build a main.s file that looks like:

```
.cpu cortex-m0
.thumb
.syntax unified
.fpu softvfp

.data
.balign 4
// Your global variables go here


.text
.global intsub
intsub:
    // Your code for intsub goes here

    // You must terminate your subroutine with bx lr
    // unless you know how to use PUSH and POP.
    bx lr



.global charsub
charsub:
    // Your code for charsub goes here

    // You must terminate your subroutine with bx lr
    // unless you know how to use PUSH and POP.
    bx lr


.global login
login: .string "xyz" // Make sure you put your login here.
.balign 2
.global main
main:
    //bl autotest // uncomment AFTER you debug your subroutines
    bl intsub
    bl charsub
    bkpt
```

---

download autotest.o

Completion code

Enter the upper and lower lines of the completion code displayed on the OLED LCD display:

Upper line: NgaxH6XGag4gD9K@
Lower line: gQ#NGB@Qv7Qp=%NF

Last Saved Results:

```
1: intsub base case:     10
1: intsub random:        20
2: charsub base case:     0
2: charsub random:        0
Total   :                30
```