Spring 2016: CSI3102-02 : Architecture of Computers

Project 1

Due Date: Monday, March 28, 2016, 9:55 AM

Objective: Learn to program in MIPS assembly language

In this project, you will be required to write assembly code to implement the following four problems. In all the programs, you are required to strictly adhere to the register conventions, as stipulated by the MIPS architecture. Please refer to the class slides, or the ISA reference manual, for the same. Please read the Submission Instructions section carefully.

Submission Instructions

There should be **no name or student number** of the team members in any part of the file or program of the project submission. If a team violates this policy, the score for the project will be **ZERO**.

1) Save your program files with extension .asm.

2) Zip all files into one file and submit it to the blackboard. Name the zip file Project1_YourId_YourTeamMemberId.zip.

3) Submit only one project zip per team

4) Insert proper comments in your code

Problem 1: String handling [10 points]

Declare a string in the data section:

```
.data

string:  .asciiz " THE FLASH THAT CUTS THROUGH DARKNESS, THE LIGHT THAT BREAKS THE NIGHT"
```

Write a program that converts the upper-case strings to lower-case characters. To convert a character to lower case character, add **0x20** to the character in the string.

Problem 2: Arithmetic expressions [20 points]

Write a program to evaluate the following function in u and v:

$$3u^2 + 6uv + 9v^2 - 5$$

Here, the variables u and v, are user inputs, and the program should receive the same from the user as input. [Refer to the sample MIPS example programs, for help with this implementation].

<u>You are required to create two subroutines for this program:</u>

Pay attention to the registers used for passing arguments to *subroutine*, and also the registers used for returning the output from a *subroutine*.

1) int Square (a): Return a^2.

2) int Multiply (a, b): Return a * b.

Problem 3: Pointers [30 points]

Write a program in MIPS assembly language that will compute the sum of all the elements in an array. Write this program using a function “PSum,” that takes two parameters; a pointer to the running sum, and a pointer to the current element.

The “C” program looks like this:

```
int sum = 0; int *sump = &sum;

int a[10];

void PSum(int *s, int *e)

{

        *s += *e;

}

int main()

{

        for (int i=0; i<10; i++)

        {

                a[i] = 3(i+1);

        }

        for (int i=0; i<10; i++)

        {

                PSum(sump, a+i);

        }

        printf("sum = %d\n", sum);

}
```

Problem 4: Recursion [40 points]

Write a program in MIPS Assembly Language to find fix(i, x), where fix(i, x) is defined recursively as:

```
int fix (int i, int x) // assume i > 0, x > 0

{

        if (x>0)

                return fix(i,x-1)+1;

        else if (i>0)

                return fix(i-1, i-1)+3;

        else

                return 1;

}
```

Note: The values for i, and x are user inputs to the program, and provision has to be made to receive the same from the user. You may set your upper-bounds for the values input for these variables.