

CS312 : MIPS Assembly Programming

Assignment 7: Functions

Instructor: Dr. Sukarn Agarwal,
Assistant Professor,
Department of CSE,
IIT (BHU) Varanasi

March 21, 2021

Functions

Functions are required to utilize the frequently accessed code, make a program more modular and readable and easier for debugging. Execution of a function change the control flow of the program two times: one at the time of calling the function and other at the time of returning from the function.

```
void main()
{
    int y, z;
    y = sum(42, 7);
    z = sum(10, -8);
    ...
}

int sum(int a, int b)
{
    return (a + b);
}
```

Figure 1: Function Code Example in C

In the above example, the `main` function invokes the function `sum` twice and the function `sum` return two times, but at the different control point in the `main`. Note that each time the `sum` invoke, the control flow has to remember the appropriate return address.

MIPS uses the jump and link instruction `jal` (format details and example are already given in Assignment-5) to invoke the function

- The `jal` store the return address (which is the address of the next instruction in the control flow of `main`) into the dedicated register `$ra`, before changing the control flow to called function.
- It is the only instruction that can access the value in the program counter. Hence it can easily store the return address `PC + 4` of the caller function in `$ra`.

To transfer control back to caller function, the callee function has to jump to address provided by the `$ra` using the following instruction: `jr $ra`.

Function accept some number arguments and operate upon them and produce return values. For example, in the above code snippet, the values 42 and 7 in the function `sum` are the actual argument and the variables `a` and `b` are the formal argument. The function return the sum of `a` and `b` as a return value.

MIPS uses the following rules for the function arguments and the return values:

- With MIPS, upto four arguments can be passed by using only argument register `a0 – a3` before invoking the function with `jal` command.
- A callee function can return upto two values using the register `v0 – v1`, before returning via `jr`.

Note that these above conventions are not enforced by the assembler or hardware, but it will be easy for different programmers to interface with the written code.

A Note about data types of the arguments passed in the function

- MIPS assembly language is untyped, means there is no distinction between integer, float, characters or pointers passed through argument.
- It is the assembly programmer job to type check different variable argument passed in the function. In other words, programmer need to make sure that argument value(s) and return value(s) are consistent to the program.

problem 1: Write a MIPS assembly program that implements different ways (using a temporary variable, using bit-wise logical operator, using multiplication and division, using addition and subtraction) to swap two integer numbers. Each swapping way is written separately in a different function.

Problem 2: Write a MIPS assembly program that lets the user to enter the two strings of a maximum length of 8 characters and a character. The MIPS program implements three functions: (a) `compute_length`: compute the length of the input string and print it (b) `search_character`: search the character and determines whether the character is present or not in the string (c) `concat_string`: concatenate the two input string into the third-string and return the concatenated string into caller function and print it. Note that the first two functions are called for both the strings, and the third function is called for two times: the first time, the first input string is passed as the first input argument, and the second input string is passed as the second argument. Whereas for the second time, the first input string is passed as the second input argument, and the second input string is passed as the first argument. The output format for the input strings "Hi", "World" and the input character 'H' is as follows:

```
Length of first string: 2
Length of second string: 5
Input character is present in the first string
Input character is not present in the second string
First concatenated string: HiWorld
Second concatenated string: WorldHi
```

Problem 3: Write a MIPS assembly program that implements three functions. The first function calculates the factorial for a given number. The second function finds the number of different permutations of n distinct objects taken r at a time. Whereas the third function finds the number of different combinations of n distinct objects taken r at a time. Note that the user provides the values of n and r .

Problem 4: Write MIPS assembly program that implements the bubble sort, quick sort (consider the pivot element to be the middle element of the array always), and selection sort functions that sort an integer array of size 10. The sorting algorithm print the sorted array along with the number of comparison and swap operations. Further, based on the comparison operation, print the best sorting algorithm. Similarly, based on the swap operation, print the best sorting algorithm. The user provides the input value of the integers for the array. The program prompts the following output on the screen:

```
Before Bubble Sort: 2, 3, 1, 0, 5, 4, 7, 6, 9, 8
After Bubble Sort: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
Total Number of Comparisons: 12 (incorrect number)
Total Number of Swaps: 13 (incorrect number)
Before Quick Sort: 2, 3, 1, 0, 5, 4, 7, 6, 9, 8
After Quick Sort: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
Total Number of Comparisons: 20 (incorrect number)
Total Number of Swaps: 19 (incorrect number)
```

Before Selection Sort: 2, 3, 1, 0, 5, 4, 7, 6, 9, 8
After Selection Sort: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
Total Number of Comparisons: 9 (incorrect number)
Total Number of Swaps: 15 (incorrect number)
Selection sort is best in terms of comparison
Bubble sort is best in terms of swap

Note: Submit all of your source code and final screen shot of the console and register panels (both integer and floating point) to the google classroom portal on the end of the day of 31st March 2021 (Indian Standard Time). Further any copy case between the assignments results into the zero marks. In case of any doubt(s) regarding the assignment, you can contact TA: Nirbhay and Deepika.