CDA 4205 Computer Architecture

Assignment 3: MIPS Assembly Procedure

1. (10 pts) Implement the following C code in MIPS assembly. Show the contents of the stack after the function call to the function “compare” is made. Assume that the stack pointer is originally at address 0x7FFFFFFc.

**int compare(int a, int b) {**

**if (sub(a, b) >= 0) return 1;**

**else return 0;**

**}**

**int sub(int a, int b) {**

**return a – b;**

**}**

1. (10 pts) Implement the following C code in the table in MIPS assembly. Show the contents of the stack after the function call to the function “fib\_iter” is made. Assume that the stack pointer is originally at address 0x7ffffffc.

**int fib\_iter(int a, int b, int n) {**

**if (n == 0) return b;**

**else return fib\_iter(a+b, a, n-1);**

**}**

1. (15 points) The following problems refer to a function f that calls another function func. The function declaration for func is “int func(int a, int b);”. The code for function f is as follows:

**int f(int a, int b, int c) {**

**return func(func(a, b), c);**

**}**

1. Translate function f into MIPS assembly code, using the MIPS calling convention. If you need to use register $t0 through $t7, use the lower-numbered registers first.
2. Right before your function f of Problem 4 returns, what do you know about contents of registers $ra, and $sp? Keep in mind that we know what the entire function f looks like, but for function func we only know its declaration.
3. (15 points) The following problems refer to a function f that calls another function func. The function declaration for func is “int func(int a, int b);”. The code for function f is as follows:

**int f(int a, int b, int c) {**

**return func(a, b) + func(b, c);**

**}**

1. Translate function f into MIPS assembly code, using the MIPS calling convention. If you need to use register $t0 through $t7, use the lower-numbered registers first.
2. Right before your function f of Problem 4 returns, what do you know about contents of registers $ra, and $sp? Keep in mind that we know what the entire function f looks like, but for function func we only know its declaration.
3. (20 points) Write a program in MIPS assembly language to **convert a positive integer decimal string to an integer**. Your program should expect register $a0 to hold the address of a null‐terminated string containing some combination of the digits 0 though 9. Your program should compute the integer value equivalent to this string of digits, then place the number in register $v0. If a nondigit character appears anywhere in the string, your program should stop with the value ‐1 in register $v0.
4. (20 points) Repeat problem 5 with **convert a string of hexadecimal digits to an integer**.

* **Submission Requirements**
* Your solutions must be in a single file with a file name yourname-hw1.
* If scanned from hand-written copies, then the writing must be legible, or loss of credits may occur.
* Only submissions via the link on Canvas where this description is downloaded are graded. Submissions to any other locations on Canvas will be ignored.
* Late submissions are accepted for a maximum of 3 late days with 20% assignment credit off as late penalization. Assignments submitted after 3 late days will not be accepted.