Indian Institute of Technology Kharagpur

AUTUMN Semester, 2018
COMPUTER SCIENCE AND ENGINEERING

Computer Organization Laboratory

Assignment-2: MIPS-32 Assembly Language Programming

Full Marks: 20

Time allowed: 6 hours

INSTRUCTIONS: ATTEMPT BOTH PROBLEMS. Make one submission per group of your source code on Moodle. Name your submitted source files following the format `Assgn_2_Prob_1_Grp_<Group_no>.s` (e.g. `Assgn_2_Prob_1_Grp_25.s`), etc. Inside each submitted file, there should be a clear header describing the assignment no., problem no., semester, group no., and names of group members. Liberally comment your code to improve its comprehensibility.

1. [**Binary Search in MIPS-32**] Write a complete MIPS-32 program to collect an array of eight integers (in ascending order) by the user, and then perform *Binary Search* to find whether a value entered by the user exists in the array. **Collect the numbers from the input console using a loop inside a function `form_array`, and store in memory in an array called "`array`". Do not store the numbers as scalars in eight different non-contiguous locations or in eight different registers**. After the input numbers are collected from the user, there should be sanity checking to ensure that the numbers in the array are actually in ascending order using a function `check_ascending_array`; if they are not, print an error message and exit. Otherwise, perform binary search by calling the function `bin_search`, which always returns its result in register `$v0`. If the search is successful, the array index at which the element was found is returned, and if the search is unsuccessful, the function returns -1. Depending on the returned value, from inside **main** print the result to the user with a proper message. **Follow all usual register usage conventions for recursive and non-recursive MIPS-32 function calls.** (5 marks)

2. [**Recursive GCD Calculation in MIPS-32**] Write a complete MIPS-32 program to calculate and display the GCD of two non-negative integers collected from the user, using a recursive function `find_gcd`. After the input numbers are collected from the user, there should be sanity checking to ensure that the integers are non-negative using a function `check_non_negative_values`; if they are not, print an error message and exit. Otherwise, print the GCD from inside the `main` function with a proper message. **Follow all usual register usage conventions for recursive and non-recursive MIPS-32 function calls.** (5 marks)

3. [**Mergesort in MIPS-32**] Write a complete MIPS-32 program to collect an array of eight integers by the user, and then perform *Mergesort* to sort the array. **Collect the numbers from the input console using a loop inside a function `form_array`, and store in memory in an array called "`array`". Do not store the numbers as scalars in eight different non-contiguous locations or in eight different registers**. In your code, have all the usual functions like the (recursive) `merge_sort`, the `merge` routine, etc. After sorting, print the sorted array on the console with a proper message. **Follow all usual register usage conventions for recursive and non-recursive MIPS-32 function calls.** (10 marks)