CS224 - Fall 2021- Lab #3 (Version 2: Oct. 30, 9:08 am, See p. 3)

MIPS Assembly Language with Program as Data, Recursion, Linked Lists

Dates: Section 1, Wednesday, 3 November, 13:30-17:20
Section 2, Thursday, 4 November, 13:30-17:20
Section 3, Wednesday, 3 November, 8:30-12:20
Section 4, Friday, 5 November, 13:30-17:20
Lab Location: EA-Z04

TAs

Section 1: Kenan Çağrı Hırlak, Pouya Ghahramanian
Section 2: Kenan Çağrı Hırlak, Soheil Abadifard
Section 3: Ege Berkay Gülcan, Sepehr Bakhshi
Section 4: Alper Şahıstan, Soheil Abadifard (Tutor: Burak Öçalan)

TA/Tutor name (x No of labs) email address, Office Hour, Office

Alper Şahıstan (x1): alper.sahistan@bilkent.edu.tr, Friday 11:30-12:20, EA-407
Ege Berkay Gülcan (x1): berkay.gulcan@bilkent.edu.tr, Thursday 14:30-15:20, EA-507
Kenan Çağrı Hırlak (x2): cagri.hirlak@bilkent.edu.tr, Friday 12:30-13:20, EA-425
Pouya Ghahramanian (x1): ghahramanian@bilkent.edu.tr, Wednesday 12:30-13:20, EA-407
Sepehr Bakhshi (x1): sepehr.bakhshi@bilkent.edu.tr, Friday 13:30-14:20, EA-507
Tutor Burak Öçalan (x1): burak.ocalan@ug.bilkent.edu.tr, Monday 18:15-20:00, EA-Z04

Purpose: This lab aims learning **1.** Fundamentals of dynamic data structure construction in assembly language. **2**. Implementation of recursion in assembly language. **3**. Practicing Von Neumann's stored program concept.

Note: Use the $s registers for the implementation of all of the subprograms, this is a requirement. In your implementations use the traditions of professional MIPS programmers. You are free to use the linked list construction and linked list printing subprograms that we studied in the classroom. You may modify them according to your needs.

Summary

Preliminary Work: 50 points

1. GenerateSortedLinkedList (35 points)
2. DuplicateListRecursive (15 points)

Lab Work: 50 points

1. InstructionCount (20 points)
2. RecursiveDivision (15 points)
3. DisplayReverseOrderRecursive (15 points)

Important Notes for All Labs

1. Try to complete the lab part at home before coming to the lab. Make sure that you show your work to your TA and answer his questions to show that you know what you are doing before uploading your lab work and follow the instructions of your TAs. In all labs if you are not told you may assume that inputs are corrects. For all works when needed please provide a simple user interface for inputs and outputs.
2. You are obliged to read this document word by word and are responsible for the mistakes you make by not following the rules. Your programs should be reasonably documented (purpose etc.) and must have a neat presentation in terms of variable names, subprogram names. indentation, comments, blank lines etc.
3. **If we suspect that there is cheating we will send the work with the names of the students to the university disciplinary committee.**

DUE DATE OF PRELIMINARY WORK: SAME FOR ALL SECTIONS

No late submission will be accepted.

a. Please upload your programs of preliminary work to Moodle by 9:30 am on Wednesday November 3 for similarity testing by MOSS.

b. For preliminary work upload a file that includes all programs in proper order with the filename **StudentID_FirstName_LastName_SecNo_PRELIM_LabNo.txt** Only a NOTEPAD FILE (txt file) is accepted. Any other form of submission receives 0 (zero).

c. Note that the Moodle submission closes sharp at 9:30 am and no late submissions will be accepted. You can make resubmissions before the system closes, so do not wait for the last moment. Submit your work earlier and change your submitted work if necessary. Note that only the last submission will be graded.

d. Do not send your work by email attachment, they will not be processed. They have to be in the Moodle system to be processed.

e. At the beginning of your submission files include the following make sure that each of them is in a separate line: Course No.: CS224, Lab No., Section No., Your Full Name, and your Bilkent ID.

DUE DATE OF LAB WORK): (different for each section) YOUR LAB DAY

a. You have to demonstrate your lab work to your TA for grading. Do this by **12:00** in the morning lab and by **17:00** in the afternoon lab. Your TAs may give further instructions on this and they may make changes. If you wait idly and show your work last minute, your work may not be graded. Make sure that you follow your TA's instructions.

b. At the conclusion of the demo for getting your grade, you will **upload your Lab Work** to the Moodle Assignment, for similarity testing by MOSS. See lab part submission details below.

c. Aim to finish all of your lab work before coming to the lab, but make sure that you upload your work after making sure that your work is analyzed by your TA and/or you are given the permission by your TA to upload.

d. At the beginning of your submission files include the following make sure that each of them is in a separate line: Course No.: CS224, Lab No., Section No., Your Full Name, and your Bilkent ID.

Part 1. Preliminary Work (50 points)

1. GenerateSortedLinkedList (35 points): Write a program that creates a linked list by getting input from the user and generates a sorted version of that linked list. Use the guidelines provided in the following for implementation.

1. Main invokes a subprogram to generate a linked list by user interaction.
2. Main invokes sorted link list generation subprogram
3. The sorted linked list generation subprogram
    a) First generates a duplicate of the linked list passed by using a iterative subprogram. (Note that passing the list head is sufficient for passing a linked list to a subprogram.)
    b) Then starts incrementally creating a sorted linked list by getting the minimum element from the duplicated linked list. Finding the minimum value is done by a separate subprogram: it finds the minimum value and deletes that number from the duplicate linked list.
    c) Adds that minimum value to the end of the newly linked list you start to create.
    d) Repeats the steps b and c until the task is done.
4. Main prints the original linked list and the newly created linked list in an easy to understand version

Test you program by interactively creating the following linked list with the data values: (6, 10, 5, 1, 8, 9, 1). Note that for this input the sorted linked list contains (1, 1, 5, 6, 8, 9, 10).

2. DuplicateListRecursive (15 points): ~~Iteratively~~ Recursively generate a copy of a linked list. Test this by a main program that provides linked list creation and duplication and printing. Use (1, 5, 6, 8, 9, 20) in your test. Note that your main program must provide a user interface for creating, and printing the original and the generated duplicate string.

Part 2. Lab Work (50 points)

1. InstructionCount (15 points): Write a subprogram to access your program as data and give the number of I-type and R-type instructions. You must have a subprogram for the count and to this subprogram pass the address of two labels that are the positions of the first instruction of the main and the last instruction of the text segment. Provide the results in a meaningful way. A simple output is sufficient. Please note that in MARS Settings: Self-modifying option must be turned on.

2. RecusiveDivision (20 points): Write a subprogram to perform integer division by successive subtractions. Return the quotient. Assume that the numbers are positive numbers. The user interface provided by the main program should allow user to try it any number of times. When one of the numbers is zero it should stop.

3. DisplayReverseOrderRecursive (15 points): Write a recursive subprogram to print a linked list in reverse order. Your program should also have subprogram to populate and print the linked list. Make sure that you display the linked list in both ways for verification.

Part 3. Submit Lab Work for MOSS Similarity Testing

1. Submit your Lab Work MIPS codes for similarity testing to Moodle.
2. You will upload one file. Use filename **StudentID_FirstName_LastName_SecNo_LAB_LabNo.txt**
3. Only a NOTEPAD FILE (txt file) is accepted. No txt file upload means you get 0 from the lab. Please note that we have several students and efficiency is important.
4. *Even if you didn't finish, or didn't get the MIPS codes working, you must submit your code to the Moodle Assignment for similarity checking.*
5. Your codes will be compared against all the other codes in the class, by the MOSS program, to determine how similar it is (as an indication of plagiarism). So be sure that the code you submit is code that you actually wrote yourself!
6. At the beginning of your submission files include the following make sure that each of them is in a separate line: CS224, Lab No., Section No., Your Full Name, Bilkent ID.
7. For your preliminary and lab works to be graded you must attend the lab.

Part 4 . Cleanup

1. After saving any files that you might want to have in the future to your own storage device, erase all the files you created from the computer in the lab.
2. When applicable put back all the hardware, boards, wires, tools, etc where they came from.
3. Clean up your lab desk, to leave it completely clean and ready for the next group who will come.

LAB POLICIES

1. You can do the lab only in your section. Missing your section time and doing in another day is not allowed.
2. The questions asked by the TA will have an effect on your lab score.
3. Lab score will be reduced to 0 if the code is not submitted for similarity testing, or if it is plagiarized. MOSS-testing will be done, to determine similarity rates. Trivial changes to code will not hide plagiarism from MOSS—the algorithm is quite sophisticated and powerful. Please also note that obviously you should not use any program available on the web, or in a book, etc. since MOSS will find it. The use of the ideas we discussed in the classroom is not a problem.
4. You must be in lab, working on the lab, from the time lab starts until your work is finished and you leave.
5. No cell phone usage during lab.
6. Internet usage is permitted only to lab-related technical sites.