

CS224 - Fall 2022 - Lab #3 (Version 2: October 17, 14:05)

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

Dates:

Section 1: Wed, 2 Nov, 13:30-17:20 in EA-Z04
Section 2: Thu, 3 Nov, 13:30-17:20 in EA-Z04
Section 3: Thu, 3 Nov, 8:30-12:20 in EA-Z04
Section 4: Fri, 4 Nov, Fri 13:30-17:20 in EA-Z04

TA name (x No of labs): email address

Kenan Çağrı Hırlak (x2): cagri.hirlak@bilkent.edu.tr
Pouya Ghahramanian (x2): ghahramanian@bilkent.edu.tr
Sepehr Bakhshi (x2): sepehr.bakhshi@bilkent.edu.tr
Soheil Abadifard (x1): soheil.abadifard@bilkent.edu.tr

TAs; Tutor:

Section 1: Pouya Ghahramanian, Sepehr Bakhshi
Section 2: Kenan Çağrı Hırlak
Section 3: Pouya Ghahramanian, Sepehr Bakhshi
Section 4: Kenan Çağrı Hırlak, Soheil Abadifard

Tutor: Yarkin Kurt (x1): yarkin.kurt@ug.bilkent.edu.tr, Lab Section 3
Tutor: Efe Yakar (x1): efe.yakar@ug.bilkent.edu.tr, Lab Section 3

Recitation: For all sections, Monday 17:30-19:30 (by Yarkin Kurt), Place: 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.

Notes: 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. If you need you may define and use additional subprograms other than the ones defined in this document. Make sure that you work is presentable and easy to follow.

Summary

Preliminary Work: 50 points

1. Merge two sorted linked list (30 points)
2. Register count: Determine the number of times a given register number is used in a program (20 points).

Lab Work: 50 points

1. RecursiveMultiply (15 points)
2. DisplayReverseOrderRecursive (35 points)

Important Notes for All Labs About Attendance, Performing and Presenting the Work

1. You are obliged to read this document word by word and are responsible for the mistakes you make by not following the rules.
2. Not attending to the lab means 0 out of 100 for that lab. If you attend the lab but do not submit the preliminary part you will lose only the points for the preliminary part.
3. 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.
4. In all labs if you are not told you may assume that inputs are correct.
5. In all labs when needed you have to provide a simple user interface for inputs and outputs.
6. Presentation of your work

You have to provide a neat presentation prepared in txt form. Your programs must be easy to understand and well structured.

Provide following six lines at the top of your submission for preliminary and lab work (make sure that you include the course no. CS224, important for ABET documentation).

CS224

Lab No.

Section No.

Your Full Name

Bilkent ID

Date

Please also make sure that your work is identifiable: In terms of which program corresponds to which part of the lab.

7. **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 PRELIMINARY WORK: SAME FOR ALL SECTIONS

No late submission will be accepted. Please do not try to break this rule and any other rule we set.

- a. Please upload your programs of preliminary work to Moodle by 13:30 on Wednesday Nov 2, 2022.
- b. Please note that the submission closes sharp at 13:30 and no late submissions will be accepted. You can make resubmissions 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.
- c. Please familiarize yourself with the Moodle course interface, find the submission entry early, and avoid sending an email like "I cannot see the submission interface." (As of now it is not yet opened.)

- 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. Use filename **StudentID_FirstName_LastName_SecNo_PRELIM_LabNo.txt** Only a NOTEPAD FILE (txt file) is accepted. Any other form of submission receives 0 (zero).

DUE DATE PART 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. If you wait idly and show your work last minute, your work may not be graded.
- 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 below for the details of lab work submission.
- c. Try to finish all of your lab work before coming to the lab, but make sure that you upload your work after making sure that it is analyzed by your TA and/or you are given the permission by your TA to upload.

Part 1. Preliminary Work (50 points)

1. MergeSortedLists (30 points): Write a program that merges two sorted linked lists. Both lists can be empty. The generated linked list must not contain repetitions and original lists remain the same as before. The subprogram returns two results: list head of the generated list and its size in terms of number of nodes. For example (for the lists L1 and L2),

if L1: (1, 3, 5, 22) and

L2: (1, 5, 12) the result linked list

L3: (1, 3, 5, 12, 22) and its size is returned as 5.

1. Main invokes a subprogram to generate the first linked list by user interaction. Main invokes a subprogram to display the linked list and its size (number nodes in it).
2. Main invokes the same subprogram to generate the second linked list by user interaction. Main invokes the subprogram that displays the linked list and its size.
3. Main invokes MergeSortedLists program
Main invokes the subprogram that displays the linked list and its size.

Test your program by interactively creating the linked lists L1 and L2 given above. In your demo to your TA show that it works in other extreme cases: 1. Both lists are empty, 2. one of them empty.

2. RegisterCount (20 points): Write a subprogram that counts the number of times the given register number is used in all instructions of the subprogram. For example, in the instruction `add $t0, $t0, $t1` the register number 8 (\$t0) is used twice times. In the instruction `lw $t1, 0($t0)` the register number \$t0 is used once. The input is the register number and the output is the number of times that register is used in the subprogram. In MARS Settings SelfModifyingCode option must be turned on. The main program must provide the user interface and it should stop when the user enters a number not withing the range of 1 to 31.

Part 2. Lab Work (50 points)

1. RecursiveDivision (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.

2. DisplayReverseOrderRecursive (30 points): Write a recursive subprogram to print a linked list in reverse order. For example if the linked list contains (1, 2, 3) in this order, it will be printed as (3, 2, 1) Your program should also have subprograms to interactively populate and print the linked list. Make sure that you display the linked list in both ways (like 1, 2, 3 and 3, 2, 1) 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 !

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.