CS224 - Fall 2024 - Lab #3 (Version 1: October 14, 18:36)

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

Dates: (TAs - Tutor)

Section 1: Wed, 23 Oct, 13:30-17:20 in EA-Z04 (TA: Kadri, Soheil; Tutor: Deniz)
Section 2: Thu, 24 Oct, 13:30-17:20 in EA-Z04 (TA: Onur, Soheil; Tutor: ...)
Section 3: Thu, 24 Oct, 8:30-12:20 in EA-Z04 (TA: Onur, Pouya; Tutor: Mert)
Section 4: Fri, 25 Oct, Fri 13:30-17:20 in EA-Z04 (TA: Berkan, Kadri; Tutor: Umut)

TA Full Name (email address: @bilkent.edu.tr)
Berkan Şahin
M. Kadri Gofralılar (kadri.gofralilar)
Onur Yıldırım (o.yildirim)
Pouya Ghahramanian (ghahramanian)
Sepehr Bakhshi (sepehr.bakhshi) (Coord.)
Soheil Abadifard (soheil.abadifard)

Tutor Full Name (email address: @ug.bilkent.edu.tr)
Deniz Çatakoğlu (deniz.catakoglu)
Mert Emre Yamalı (emre.yamali)
Umut Başar Demir (basar.demir)

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. See the program linkedList HowToCreate HowToPrint in the Moodle assignment folder that show how to create and print a linked list.

Summary

Preliminary Work: 60 points

1. Generate Summary Linked List (40 points)
2. Program as Data - Register Count (20 points).

Lab Work: 40 points

1. Recursive Division (15 points)
2. Recursively Display Linked List in Reverse Order (25 points)

Implementation Notes and Requirements for Subprograms

You are not allowed to use $t registers in the subprograms. In subprograms only use $s registers to get used to the MIPS software development traditions. When you enter to a subprogram save $s registers you use in the subprogram to the stack. When necessary save $ra register too. For passing arguments to subprograms use argument ($a) registers and for returning results to the caller use $v0 and $v1 registers. If you do not follow this rule 10 points will be taken off from your grade.

You can only use a limited number of pseudo instructions in your lab work; such as: lw and sw instructions (like: lw $t0, a; sw $t0, a), load address (like: la $t0, a), and conditional instructions (like: bgt, blt ...). If you do not follow this rule some points will be taken off from your grade.

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

You are obliged to read this document word by word and are responsible for the mistakes you make by not following the rules.

1. 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.
2. Try to complete the lab part at home before coming to the lab. Make sure that you show your work to your TAs and answer their questions to show that you know what you are doing before uploading your lab work and follow the instructions of your TAs.
3. In all labs if you are not told you may assume that inputs are correct.
4. In all labs when needed you have to provide a simple user interface for inputs and outputs.
5. 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.
6. **If we suspect that there is cheating we will send the work with the names of the students to the university disciplinary committee. You can experiment with ChatGPT for learning; however, you cannot use/modify the code generated by it. Such an act is classified as plagiarism. Note that MOSS is capable of detecting ChatGPT code. Furthermore remember that, the code you use from ChatGPT can also be used by another student in the course. Make sure that the code you submit is really yours and has been internalized.**

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, 23 October, 2024.

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 ınterface." (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 (60 points)

1. Generate Summary Linked List (40 points):The input to the subprogram is a linked list as defined here: Each node contains Key and Data link fields. Consider the following example input linked list:

--> (15, 20) -->(15, 10) --> (15, 3) (20, 40) --> (55, 17) --> (75, 2) --> (75, 20)

The summary link list generated:

--> (15, 33) -->(20, 40) --> (55, 17) --> (75, 22)

Note that original input linked list is retained. In each node the first number indicate a product number plays the role of key and the second number indicates the number of copies sold for that product. In the input linked list, for the product number 15 the number of copies sold is 20, 10 and 3. We have 33 for product number 15 in the summary linked list since it is the summation of 20, 10 and 3 associated with this product (15).

The subprogram also returns the summary link list size in terms of number of nodes.

The input link list is sorted according to product number and we may have one or more nodes associated with a product number.

Test you program by interactively creating the input linked list. Provide a meaningful user interface in terms for getting and displaying the values.

2. Register Count (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 within the range of 0 to 31.

Part 2. Lab Work (40 points)

1. Recusive Division (15 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. Display Linked List in Reverse Order Recursively (25 points): Write a recursive subprogram to print a linked list in reverse order. As input use the linked list structure defined above in Preliminary Work Program 1.

```
For example, the linked list --> (15, 33)  -->(20, 40) --> (55, 17) -->  ( 75,  22)
will be displayed as            --> ( 75,  22) -->(55, 17) --> (20, 40) -->  (15, 33)
```

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. Comparison of your programs with other students's programs: The MOSS plagiarism detection tool determines how similar they are to other programs (as an indication of plagiarism). So be sure that the code you submit is code that you actually wrote yourself ! You are not allowed to use web resources that solves the assigned programs.
6. The effectiveness of MOSS for chatbot code is quite good, with a detection rate of much higher than 50%. (The answer is provided by chatbot.)

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 works for ChatGPT code too. 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.