CS224 - Spring 2021 - Lab #3 ((Version 1: February 24, 12:13 am)

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

Dates:
Section 1: Mon, 8 Mar, 8:30-12:20 in EA-Z04
Section 2: Wed, 3 Mar, 13:30-17:20 in EA-Z04
Section 3: Tue, 2 Mar, 13:30-17:20 in EA-Z04
Section 4: Mon, 8 Mar, 13:30-17:20 in EA-Z04
Section 5: Fri, 5 Mar, 8:30-12:20 in EA-Z04
Section 6: Fri, 5 Mar, 13:30-17:20 in EA-Z04

TAs:
Section 1: Zülal Bingöl, Ergün Batuhan Kaynak
Section 2: Zülal Bingöl, Kenan Çağrı Hırlak
Section 3: Alper Şahıstan, Kenan Çağrı Hırlak
Section 4: Ege Berkay Gülcan, Ergün Batuhan Kaynak
Section 5: Hüseyin Eren Çalık, Yusuf Dalva
Section 6: Ziya Erkoç, Yusuf Dalva

TA name (x No of labs) email address:
Alper Şahıstan (x1): alper.sahistan@bilkent.edu.tr
Ege Berkay Gülcan (x1): berkay.gulcan@bilkent.edu.tr
Ergün Batuhan Kaynak (x2): batuhan.kaynak@bilkent.edu.tr
Hüseyin Eren Çalık (x1): eren.calik@bilkent.edu.tr
Kenan Çağrı Hırlak (x2): cagri.hirlak@bilkent.edu.tr
Yusuf Dalva (x2): yusuf.dalva@bilkent.edu.tr
Ziya Erkoç (x1): ziya.erkoc@bilkent.edu.tr
Zülal Bingöl (x2): zulal.bingol@bilkent.edu.tr

Lab Attendance and Rotation Policy Reminder and Time for the Next Lab:
All labs will be done online and attendance is mandatory. You have to attend the online labs and submit the lab work by following the instructions of your TA. Before submission your TA will do a brief interview with you to understand your knowledge of the work you plan to submit. Note that preliminary works are submitted before all labs and are graded only if you attend the lab and submit your lab work.

Next Lab - Lab4 Days: Due to the lab rotation policy the first Lab4 session will be with Sections 5 & 6 on March 19, Friday. The lab days for Sections 1 to 4 are on the following week starting with March 22, Monday.

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

You are obliged to read this document word by word and are responsible for the mistakes you make by not following the rules. As indicated earlier, your programs should be reasonably documented must have a neat syntax in terms of variable names, comments.

Summary
Preliminary Work:
Part 1 (20 points) **InstructionCount**: Access your program as data and count add (R type) and lw instructions.

Part 2 (20 points) RecusiveDivision: Perform division by successive subtractions.

Lab Work:
Part 3 (20 points) DisplayReverseOrderRecursively: Recursively print a linked list in reverse order.

Part 4 (20 points): DuplicateListIterative (20 points): Iteratively generate a copy of a linked list.

Part 5 (20 points): DuplicateListRecursive (20 points): Recursively generate a copy of a linked list.

Note try, study, and aim to complete lab part at home before coming to the online lab.

DUE DATE OF PART 1 & 2 (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 Tuesday Mar 2.
b. Please 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 the last moment. Submit your work earlier and change your submitted work if necessary. Note that only the last submission will be graded.
c. Do not send your work by email attachment they will not be processed. They have to be in the Moodle system to be processed.
d. 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 3 & 4 & 5 (LAB WORK): (different for each section) YOUR LAB DAY

a. You have to demonstrate (using Zoom) 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 Part 3 & 4 & 5** to the Moodle Assignment, for similarity testing by MOSS. See Part 6 below for details.

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. We use zoom to track your lab attendance.

If we suspect that there is cheating we will send the work with the names of the students to the university disciplinary committee.

Part 1 & 2. Preliminary Work (40 points)

You have to provide a neat presentation prepared in txt form. Provide following five 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

Make sure that you identify what is what at the beginning of each program by using proper comments.

Important Notes

1. **Assume that all Inputs are Correct**. In all lab works, if it is not explicitly stated, you may assume that program inputs are correct.
2. **Provide simple text messages to help our TAs to understand and validate your program**. Make sure that your program can be validated by text output you provide in your programs. This is mandatory.
3. **In the subprograms you are not allowed to use $t registers.** Instead of $t registers use $s registers and make sure that you push $s registers you use to the stack. Before returning to the caller make sure that you restore the values of the $s registers from the stack.
4. **A linked list construction and listing program is provided**. Use it for any case you find appropriate. See the related Moodle folder.

Part 1.

instructionCount (20 Points): Write a non recursive subprogram that counts the number add (e.g., "add $t0, $t1, $t2") and lw (load word) instructions and returns these counts as its results. The address of the first instruction to be considered in counting is given in $a0, and the address of the last instruction to be considered in counting is given in $a1. Your main program should invoke this subprogram twice: first for the main (itself), and for the second time for the subprogram.

As you see, in this program your main program becomes an input data and in the second call the subprogram becomes its own data (like an MD operator operating itself). Provide meaningful output in main so that you and your TA can validate the correctness of your program.

Note that, in our practice to simplify things, we write MIPS programs in a single source file; therefore, all labels are at the global scale. For example, if you label the first instruction of your subprogram with the label L1, and its last instruction with the label L2, you can pass these addresses to the subprogram with load address instructions in main (la $a0, L1; la $a1, L2).

Part 2.
RecursiveDivision (20 points): Write a **recursive** MIPS subprograms that finds the division of a positive number by another positive number by successive subtractions. Your main program must provide a user interface: It should get a pair of numbers from the user provide the answer, and should ask the user if the user wants to continue, if so it should ask the next pair of numbers, etc.

Part 3 & 4 & 5. Lab Work (60 points)

In the following parts first construct a linked list that contains ten numbers (1, 2, 3, ... 10 in this order). For the linked list construction use the program provided to you (modify it as needed). Display the linked list again with the program you are given to make sure that the linked list is constructed properly. After that do what you were asked in each part below. You must provide simple easy to follow program output with proper messages for the user (your TA).

Implement each part in a separate source program for easy grading and to prevent confusion during grading.

Part 3.
DisplayReverseOrderRecursively (20 points): Write a **recursive** program to display the elements of a linked list in reverse order (10, 9, ... 1). The list head is passed in $a0. Provide a simple user interface.

Part 4.
DuplicateListIterative (20 points): Write a **non recursive** subprogram to duplicate a linked list. When called $a0 points to the original list. It returns the new list head in $v0. Provide a simple user interface.

Part 5.
DuplicateListRecursive (20 points):

Write a **recursive** subprogram to duplicate a linked list. When called $a0 points to the original list. It returns the new list head in $v0. Provide a simple user interface.

Aim to complete lab part at home before coming to the online lab.

Part 6. Submit Your Code 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 7. Lab Policies (Reminder)

1. As indicated in Lab1: Attendance is mandatory and the preliminary work is graded only if you submit your lab work with the observation/permission of your TA.
2. You can do the lab only in your section. Missing your section time and doing in another day is not allowed.
3. The questions asked by the TA will have an effect on your lab score.
4. 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.
5. Your lab attendance is tracked by the Zoom system. Please also see lab policies no. 1 item.