CS224 – Fall 2017 - Lab #2 v2

Creating Linked List Utility Routines in MIPS Assembly

Dates:

Section 1:	Wed, Oct. 18th	13:40-17:30 in EA-Z04
Section 2:	Fri, Oct. 20th	13:40-17:30 in EA-Z04
Section 3:	Wed, Oct. 18th	08:40-12:30 in EA-Z04
Section 4:	Thu, Oct. 19th	13:40-17:30 in EA-Z04

Purpose: to learn how to write, edit, debug and run advanced MIPS assembly language programs, using MARS, a MIPS simulator.

Summary

Part 1 (30 points): Preliminary Report/Preliminary Design Report: Linked list utilities (Due date of this part is the same for all groups)

Part 2 (70 points): LinkedList Utilities (parts A and B)

DUE DATE OF PART 1: SAME FOR ALL SECTIONS Dear students please bring and drop your preliminary work into the box provided in front of the lab before 3:59 pm on Tuesday October 17th. For late submissions there will be a substantial penalty.

LAB WORK SUBMISSION TIMING: You have to show your lab work to your TA by **12:15** in the morning lab and by **17:15** in the afternoon lab. Note that you cannot wait for the last moment to do this. If you wait for the last moment and show your work after the deadline time 20 points will be taken off.

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. Preliminary Work / Preliminary Design Report
(30 points, contains sections 1 to 3 & each section 10 points)

You have to provide a neat presentation prepared by Word or a word processor with similar output quality (for some Latex is the science fiction editor for undergraduates, you may challenge yourself by trying it). At the top of the paper on left provide the following information and staple all papers. In this part provide the program listings with proper identification (please make sure that this info is there for proper grading of your work, otherwise some points will be taken off).

CS224
Section No.: Enter your section no. here
Fall 2017
Lab No.
Your Full Name/Bilkent ID

Linked Lists:

Linked lists are important dynamic data structures, useful to a variety of algorithms because of their ability to grow and shrink. Utilities libraries for linked lists are therefore useful, so that common functions can be available to users, pre-written, tested and ready to go. In this lab, you will write 3+2 utility functions for linked lists, in MIPS assembly language. Your utilities will be combined with other utility programs such as **create_list** and **display_list**, and called by a **main** program.

In memory, the linked lists your utilities will work with are implemented as follows: each element consists of 2 parts: **pointerToNext**, and **value**. Each part is a 32-bit MIPS word in memory. The two parts are located in successive word addresses, with the **pointerToNext** being first. For example, if the byte address of the pointerToNext is 100, then the byte address of the value will be 104. Remember, MIPS memory is byte addressable.

In the last element of the linked list, the pointerToNext has value 0, in other words it is the null pointer. This means that there is not any next element. But of course the last element still has a value.

For this lab, you will write the following linked list utility routines as part of the preliminary report. You may start writing your program with the “skeleton.asm” file.

[10 points] 1. Write a MIPS program named as **_Lab2main** that works as the main program that calls linked list utility functions, depending on user selection. This program outputs a message, then lists the menu options and gets the user selection, then calls the chosen routine, and repeats.

[10 points] 2. Write a utility as part of the main program named as **create_list** that creates the contents of a linked list, element by element. It will first ask the user to enter the number of elements in the linked list (It will accept non-negative integer values: 0, 1, 2, etc).

[10 points] 3. Write a utility as part of the main program named as **display_list** which shows the contents of a linked list.

Notes:

- You may store the messages in .data segment.
- Make sure you initialize everything - do not assume that memory is initialized to 0.
- The utility programs that you will write must be callable from the main program by the names given in bold. *They must obey the MIPS convention for register usage, and thus will have no unexpected side-effects (we will test your functions in our context so if there is a problem with preserved registers your program can potentially execute incorrectly).* Your MIPS code should be well-commented (meaning most lines will have explanatory comments), it will use white space and indenting for easy readability, and each routine will have an explanatory header at the top, naming the registers and storage used and what they contain, and describing the algorithms used. Together these 3 characteristics make for self-documenting code; this style of code can be easily understood by anyone.

- If needed, you may write additional utility programs that serve as subroutines to make these 4 utilities easier to design and implement. In particular, sharing or reuse, and modularity, are key software design principles that should be followed.

- Since linked lists are dynamic data structures, their size cannot be known in advance. It is dynamically determined at run-time, according to the needs of the program. This means that dynamic memory allocation must be used, and the linked lists will be located in the heap. To do this, programs must ask the operating system to allocate heap memory. Since MARS and SPIM do not offer a way to return heap memory to the operating system, don't worry about that.

Part 2. Adding Utilities to Your Program
(35 points for each part)

For this lab, you will write the following linked list utility routines. Each will be worth 35 points when you show its working demo to the TA for your grade.

A) It will be given on the board by the TA
B) It will be given on the board by the TA

Part 3. Submit your code for MOSS similarity testing

Submit your MIPS codes for similarity testing to the Unilica > Assignment specific for your section. You will upload one file: **name_surname_SecNo_MIPS.txt** created in the relevant parts. Be sure that the file contains exactly and only the codes which are specifically detailed Part 1 to Part 3, including Part 1 programs (your paper submission for preliminary work must match MOSS submission). Check the specifications! *Even if you didn't finish, or didn't get the MIPS codes working, you must submit your code to the Unilica Assignment for similarity checking.* 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 ! All students must upload their code to Unilica > Assignment while the TA watches. Submissions made without the TA observing will be deleted, resulting in a lab score of 0.

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. Students will earn their own individual lab grade. The questions asked by the TA will have an effect on your individual 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.
7. For labs that involve hardware for design you will always use the same board provided to you by the lab engineer.