FACULTY OF COMPUTERS AND INFROMATION, CAIRO UNIVERSITY

CS322: Computer Architecture and Organization
Year 2019-2020

First Semester

Assignment 4 – Version 2.0

Course Instructors:

Dr. Mohammad El-Ramly

Revision History

Version 1.0	Dr Mohammed El-Ramly	28 Nov. 2019	Version 1.0
Version 2.0	Dr Mohammed El-Ramly	30 Nov. 2019	Option 2 = 3~4 kids

CS322: Computer Architecture and Organization
Assignment 4 (4 or 7 marks) – Version 2.0

Cairo University, Faculty of Computers and Information

احبائى الطلاب ، أشكركم على تعبكم فى هذا المقرر و أدعو لكم بالتوفيق فيه و فى كل المقررات ، و أعتذر عن أى تقصير من جانبى و أرحب بأية مقترحات أو تعليقات (لا تشمل تقليل المنهج) ، و أرجو التزام الأمانة العلمية و عدم تقديم أى حل غير أصلى أو منقول من أى مصدر و إلا تعرض الطالب لعقوبات لا أحب أن أوقعها بأى طالب.

Objectives

1- Learn MIPS assembly and architecture.
2- Understand how programs and algorithms are represented in assembly.
3- Understand how the mapping from high level languages to low level languages works.

Instructions

1. **These instructions must be followed to get the full marks. يجب اتباع هذه التعليمات بكل دقة**
2. **Deadline is Wednesday 10 Dec. 2019 @ 11 pm**
3. **Form groups of 3 students.** (See option 2 below)
4. **Students have two options:**
   - **Option 1:** Solve the 12 given problems. (4 marks)
   - **Option 2:** Develop a MIPS simulator for a subset of MIPS with a GUI. (7 marks)
5. **Please submit only work that you did yourself. If you copy work from a friend or book or the net you will fail the course. لا تسليم حلول منقولة من أى مصدر يؤدى إلى الرسوب فى هذا المقرر. تغش الحل أو تنقله من أى مصدر و اسألنى فى أى شئ لا تفهمه لكن لا تنقل الحلول من النت أو زملائك أو أى مكان و لو عثرت على حلول من أى مصدر أو زميل لا تنقل منها أى شئ وقد أعذر من أنذر ولن يتم التهاون مع الغش مطلقا**

Task 0 (0 marks)

1. Get familiar with MIPS-32 assembly language and architecture.
2. Train on problem solving with MIPS assembly and using QTSPIM assembler.
3. Read this link for help https://www.cse.unsw.edu.au/~cs1521/18s2/notes/C/notes.html

Option 1 (4 marks)

Each team member will solve one set of the following programs. Each one includes 4 programs. **The ID with smallest least-significant digit will solve Set 1.** The next will solve **Set 2,** etc. Use MIPS assembly and deliver a working solution that correctly runs on QTSPIM. Solution must have:

- A header explaining the file content, author, version, date, etc.
- Proper coding style and sufficient comments to explain the solution.
- Must be free of any syntax or logic errors.
- Must work properly on QTSPIM simulator.
- Test your programs with sample data.
- Make any assumptions you need.
- **Do not copy any solution.**

CS322: Computer Architecture and Organization
Assignment 4 (4 or 7 marks) – Version 2.0

Cairo University, Faculty of Computers and Information

Set 1:

1- Write, run and test a MIPS program to implement the following C code segment.

```
if (g > h)
     g = g + h;
else if (g < h)
     g = g – h;
else
     g = g * h;
```

2- Write, run and test a MIPS program that takes an array of characters and prints the number of UPPERCASE letters and the number of LOWERCASE letters.

3- Write, run and test a MIPS program to execute the following nested C loop.

```
for(i = 0; i < a; i++)
     for(j = 0; j < b; j++)
        C[2 * i] = i – j;
```

4- Translate the following C program to MIPS assembly.

```
int main() {
     ...
     t1 = sumOdd (10);
     printf ("%d", t1);
     ...
}
int sumOdd (int n) {
     int i, result = 0;
     for (i = 0; i < n; i++)
          if ((i % 2) == 1)
               result += i;
     return result;
}
```

Set 2:

1- Write, run and test a MIPS program to implement the following C code segment.

```
if ((g <= h) && (g > 0))
     g = h;
else
     h = g;
```

2- Write, run and test a MIPS program that takes an array of characters and prints the number of vowels ('a', 'e', 'i', 'o', 'u') in it. Consider both upper and lower cases. E.g., "Utility" prints 3.

3- Write, run and test a MIPS program to execute the following nested C loop.

```
for(i = 0; i < a; i++)
    for(j = 0; j < i; j++)

        C[i] += j;
```

CS322: Computer Architecture and Organization
Assignment 4 (4 or 7 marks) – Version 2.0

Cairo University, Faculty of Computers and Information

4- Translate the following C program to MIPS assembly. **Use the same structure. Do not write a clever shorter program.**

```
int main() {
     ...
     t1 = isOdd (10);
     printf ("%d", (t1 ? "odd" : "even");
     ...
}

int isOdd (int n) {
     return !isEven (n);
}

int isEven (int n) {
     return ((n % 2) == 0));
}
```

Set 3:

1- Write, run and test a MIPS program to implement the following C code segment.

```
if (g >= h)
     g++;
else
     g--;
```

2- Write, run and test a MIPS program that takes an array of characters, then reverses it in memory and prints the reversed version.

3- Write a MIPS program to execute the following nested C loop.

```
for(i = 0; i < a; i++)
    for(j = i; j >= 0; j--)
        C[i] *= j;
```

4- Translate the following C program to MIPS assembly.

```
int main() {
     ...
     t1 = fact(8);
     t2 = fact(3);
     t3 = t1 + t2;
     ...
}
int fact(int n) {
     int i, result = 1;
     for (i = n; i > 1; i--)
          result  = result * i;
     return result;
}
```

CS322: Computer Architecture and Organization
Assignment 4 (4 or 7 marks) – Version 2.0

Cairo University, Faculty of Computers and Information

Option 2 (7 marks) – The group here can be 3 to 4.

Write a MIPS simulator that consists of assembler and virtual machine.

Assembler:

The assembler is able to take an assembly program written in a subset of MIPS-32 instruction set and translate it into machine language. The assembler should be able to do the following:

1- Provide a GUI interface and allow the user to load a program:
   - By typing in the instructions directly.
   - Or by copying and pasting.
   - (Optional) By loading from a file.
2- Support the following instructions:
   - The basic functions without pseudo-instructions and without function calls.
   - Loads and stores: **`lw`**, **`sw`**
   - ALU operations: **`add`**, **`addi`**, **`sub`**, **`and`**, **`or`**, **`andi`**, **`ori`**, **`sll`**, **`slt`**, **`slti`**, **`lui`**
   - Branches/jumps: **`jr`**, **`j`**, **`beq`**, **`bne`** and **labels**. Assembler must decide jump destination.
3- The assembler will take as input the text of the assembly program and will output a list of machine language instructions. (See the assumptions in MIPS Virtual Machine section below)
4- Note: Your assembler does not work like QTSPIM. So no need to support data declarations, data types, data and text sections, etc. You will get input programs in pure MIPS assembly.

MIPS-32 Virtual Machine:

The virtual machine simulates the structure of MIPS-32 processor **and can run the machine language program produced by the assembler**. Your machine should be able to do the following:

1- Provide a GUI to allow the user to see the status of the **Program Counter,** MIPS **registers** (or at least the important ones), the relevant part of the **memory** (were used variables are stored) and the **currently executing instructions** and its type (R, I, J), parts like opcode, registers, etc.
2- The machine will have buttons to execute the next instruction or the entire program.
3- Make the following assumptions:
   - Memory has only two segments: Text (Program) segment and Data segment.
   - The program is always loaded starting from memory location **0**. (Start of text segment)
   - Data is stored starting from memory location **0x00001000**. (Start of Data segment)
4- A close app is here https://www.facebook.com/groups/1749816315311531/permalink/1789809447978884/

Design and Development:

1- Develop the program in JAVA. Take advantage of JAVA Swing GUI components.
2- **Write file headers with date, author, version, etc. and which Java version / tools you used.**
3- Use object-oriented design properly.
   - First think of the classes your system should have in this system.
   - Next, think of the responsibility of each class and its attributes and methods to do its job.
   - Next, think of the external **API** (**public methods**) that other classes can use in this class.
   - Finally, think of how these classes cooperate to achieve the task of the program.
   - (Advanced - Optional) Consider using the **Command Design Pattern**.

CS322: Computer Architecture and Organization
Assignment 4 (4 or 7 marks) – Version 2.0

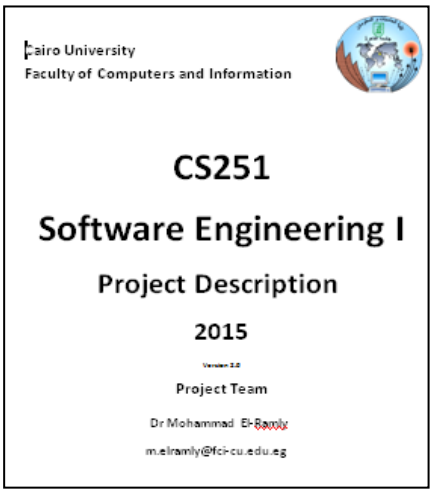
Cairo University, Faculty of Computers and Information

4- How will you parse and assembly to (1) find if an instruction is valid or not, (2) what each instruction is and (3) the labels that accompany some instructions?
   - There is two ways to do so.
   - The first is ad-hoc parsing بالفهلوة using Java regular expressions. Here are some links to learn more. But you kind find better resources:
     - http://zetcode.com/java/regex/
     - https://stackoverflow.com/questions/5688704/regular-expression-for-assembly-instructions
     - https://www.journaldev.com/634/regular-expression-in-java-regex-example
   - The second is using a proper parser or generating one using a parser generator. This **is the ideal but harder way.** A parser generator is a tool that generates a parser for a language given a "**grammar**" representing this language. Luckily there is are popular parsers for Java like JavaCC and there is a MIPS Assembly grammar written for it. Will they work well together, I do not know. There is also a MIPS Assembly parser that may work directly for you. Here are some links to start your search:
     - https://stackoverflow.com/questions/5200756/which-is-the-best-way-to-parse-a-file-containing-assembly-language-using-java
     - https://javacc.org/
     - http://www.cse.iitd.ernet.in/~nvkrishna/courses/winter07/grammar+spec/mips.jj
     - http://www.cse.iitd.ernet.in/~nvkrishna/courses/winter07/a7.html
     - http://dmitrysoshnikov.com/compilers/mips-assembly-parser/

CS322: Computer Architecture and Organization
Assignment 4 (4 or 7 marks) – Version 2.0

Cairo University, Faculty of Computers and Information

Submission Instructions


Cairo University
Faculty of Computers and Information

CS251

Software Engineering I

Project Description

2015

Project Team

Dr Mohammad El-Ramly

m.elramly@fci-cu.edu.eg

1. **Team will submit in acadox.**
2. Submit in acadox a zip file with the following:
3. PDF report with a header as in the figure that includes:
   - Team name, **and emails,** assignment details, etc.
   - MIPS solutions to the problems given for **option 1**.
   - Screen shots and user manual **for option 2.**
4. MIPS assembly code / Java code nicely organized in folders and named properly.
5. Name file CS322-A4-Option1-ID1-ID2.pdf / .zip

```
+
|____ A4-Option1
|         +
|         |_ Set1-ID
|         |      +
|         |      |- Problem 1
|         |      |- Problem 2
|         |      |- Problem 3
|         |      |- Problem 4
|         |
|         |_ Set2-ID
|                +
|         |      |- Problem 1
|         |      |- Problem 2
|         |      |- Problem 3
|         |      |- Problem 4
```

6. **PLEASE do not submit huge files**
7. Team members are expected to help each other but not do work of others.
8. **All team members must understand the details** of all solutions and be able to explain them
9. TA can ask any team member about any of the programs developed and its code**.**

Marking Criterion

Option 1:

1. 1 x 4 1 mark for each correct problem solution
2. -1 for failing to explain the code of other students
3. -1 for low quality code with bad style.
4. -4 for submitting non-original code.

Option 2:

1. 2 for developing a nice looking correctly working GUI
2. 2 for the assembler and translating the assembly into machine language
3. 2 for the virtual machine that is able to correctly execute the machine language code
4. 1 for nice and clean code that is well-commented and documents.
5. -2 for failing to explain any part of the program.
6. -7 for submitting non-original code.