Middle East Technical University | Department of Computer Engineering

CENG 331

Computer Organization

Fall 2013-2014

Programming Assignment 1

Due date: November 8th, 2013, Friday, 23:55:50

This assignment aims to get you familiar with MIPS and assembly programming. You will code a MIPS program that will further develop your skills in writing MIPS code, with a focus on implementing function and system calls.

1. Problem Definition

In this assignment, you are expected to write a program that evaluates logic expressions. Logic expressions consist of variables which are combined by operators. A variable name is only one of alphabetical character such as "a, A, H, o". Variables can take values 0 or 1. The operators combining variables together into a logic expression are as follows:

/ operator:		
a1	a2	a1 / a2
0	0	0
0	1	1
1	0	1
1	1	0

* operator:		
a1	a2	a1 * a2
0	0	0
0	1	1
1	0	1
1	1	1

. operator:		
a1	a2	a1 . a2
0	0	0
0	1	0
1	0	0
1	1	1

The precedence and the associativity of the operators are as follows (top: highest precedence, bottom: lowest precedence):

Operator Associativity
/ left-to-right
* left-to-right
. left-to-right

An expression can contain any number of variables and operators. For example, the following are valid expressions:

- ✓ `A.B*A*C/D*C/A/e`
- ✓ `A.a/A.C/D.C/C`

2. Specifications

You will read your input from stdin, and write your output to stdout. For these operations, you can use MIPS's operating system calls, namely read string and write string. Your program will first read a logic expression from standard input. The expression consists of at most 200 characters (including both variables, operators and assignments). The logic expression will be terminated by "#". After reading the logic expression, your program will read the values of the variables in the logic expression in the format below (the variable name, the assignment sign and the value).

Var1=<value> Var2=<value> ... VarN=<value>

There is only one space character between each variable assignment. There is no space before and after "=" sign.

All the variables in the logic expression will be given only one valid value (0 or 1). The list of variable values will be ended by "#". There is no space before or after the first "#" and there is no space before the second "#". You can see the example input in "How to run" section.

Your program will evaluate the logic expression with the provided variable assignments. In addition, your program must contain the procedure "***evaluate:***". This procedure should take at least one argument: the logical expression (a reference to this expression). The form of this expression is up to you, it may be a string, an array etc. You can use additional arguments if you need. Then the procedure must return the evaluated value. Then you should print the result as follows:

Result: <value>

There will be only one space between ":" and <value>, and Result will be at the beginning of the line. Make sure that you display Result: <value> on the next line after # is given.

Notes

- We will not test your program with invalid expressions or invalid assignments/values. The input to your program will be error-free.
- There will not be spaces given after the first #.
- **Write comments in your code.** Comments for each line are not required, but include enough comments in order to understand to steps (pseudo-code level) (Points will be taken off if missing).
- You will use **QTSpim** (Windows/Linux), or **spim** (Linux console) to execute/debug your MIPS assembly code.

3. Submission

Submission will be done via COW. Submit a single MIPS assembly source file named hw1.s that can be run with the SPIM simulator.

Note: Your homework will be graded on department inek machines. The following command sequence is expected to run your program on a Linux system:

$spim -file hw1.s

4. Resources

- QTSpim: http://spimsimulator.sourceforge.net/
- D. A. Patterson and J. L. Hennessy, Computer Organization and Design: The Hardware/Software Interface, 3rd Edition (Chapter 2)
- Recitation Documents

5. How to run

```
$spim -file hw1.s
A.B*C/D*C/a#A=0 B=0 C=1 D=1 a=0#
Result: 0

$spim -file hw1.s
A.b/C*A/b.C*a#A=1 a=0 b=0 C=1#
Result: 1
```

6. Regulations

1. Programming Language: MIPS

2. Late Submission: Submissions can be delayed with a penalty of $8 \times (\text{latedays})^2$.

3. Cheating: We have zero tolerance policy for cheating. People involved in cheating will be punished according to the university regulations.

Cheating Policy:

Students/Groups may discuss the concepts among themselves or with the instructor or the assistants. However, when it comes to doing the actual work, it must be done by the student/group alone. As soon as you start to write your solution or type it, you should work alone. In other words, if you are copying text directly from someone else - whether copying files or typing from someone else's notes or typing while they dictate - then you are cheating (committing plagiarism, to be more exact). This is true regardless of whether the source is a classmate, a former student, a website, a program listing found in the trash, or whatever. Furthermore, plagiarism even on a small part of the program is cheating. Also, starting out with code that you did not write, and modifying it to look like your own is cheating. Aiding someone else's cheating also constitutes cheating. Leaving your program in plain sight or leaving a computer without logging out, thereby leaving your programs open to copying, may constitute cheating depending upon the circumstances. Consequently, you should always take care to prevent others from copying your programs, as it certainly leaves you open to accusations of cheating. We have automated tools to determine cheating. Both parties involved in cheating will be subject to disciplinary action. [Adapted from http://www.seas.upenn.edu/~cis330/main.html]

4. Newsgroup & Questions: You must follow the newsgroup (metu.ceng.course.331) for discussions and possible updates on a daily basis. Use the newsgroup for your questions.

5. Evaluation: Your program will be evaluated automatically using black-box technique so make sure to obey the input/output specifications.

6. **The homework must be done individually, so there will not be any team work.**