Ho Chi Minh City University of Technology

Faculty of Computer Science and Engineering
Course: Computer Architecture Lab (CO2008)

Assignment

CALCULATOR

Ho Chi Minh City, April 2024

Contents

1 Outcomes

After finishing this assignment, students can proficiently use:

- MARS MIPS simulator.
- Arithmetic & data transfer instructions.
- Conditional branch and unconditional jump instructions.
- Procedures.

2 Introduction

A calculator is typically a portable electronic device used to perform calculations, ranging from basic arithmetic to complex mathematics [1].

Figure 1: A simple electronic calculator

The calculator idea starts from long ago where people would use items in their possession to count. After some time, when civilization became more and more advanced, mechanical calculators emerged and came to be one of the fundamental tools for a person. At the time, such tools can only offer very few functionalities, including addition and subtraction. After the invention of the first solid-state electronic calculator in the 1960s, calculators began to have memory storing capabilities. Therefore, complex instructions can be calculated. Nowadays, calculators are inseparable from the modern society. From elementary schools to university and marketplaces, these pocket-size devices can be found supporting people daily calculations anywhere in the world.

A simple modern calculator should be able to solve basic arithmetic such as addition, subtraction, multiplication, division, and utilizing memory. Some other frequently used functions include factorization, finding Lowest Common Multiple (LCM), Greatest Common Divisor (GCD), square roots, exponents, and converting bases.

3 Requirements

In this assignment, your task is to create the best calculator that you can using MIPS assembly. The requirements are listed in the below sections.

3.1 Input

The program should be able to receive input from the user. The user should input a string of instructions after the prompt: "Please insert your expression:".

The input will be a string of expression such as: "(10-4+9)/(9+3*2+M)". The length of the string should be 100 characters max.

The program should be able to detect and prompt the user if there is a wrong character in the input string. The prompt should be: "You inserted an invalid character in your expression"

Valid characters include: "0 1 2 3 4 5 6 7 8 9 + - * / . M ˆ ! ( )"

The user should be able to insert multiple expressions in a single running instance of the program. For example, after getting the result of "1+2", the user can input another expression "M+1" to get the next result. This cycle goes on until the program is terminated by the word: "quit".

3.2 Output

The program should output the calculated result of the expression inserted by the user. It should come after the prompt: "Result:". In case the result is an infinite decimal number, it should display 16 numbers after the decimal point.

3.3 Instructions

Your program must be able to handle these instructions:

3.3.1 Addition

The addition operator is indicated by the symbol "+". It is a binary operation (requires 2 operands). Example: input: "2+3"; output: "5"

3.3.2 Subtraction

The subtraction operator is indicated by the symbol "-". It can be a binary or unary operation. In the former case, it works exactly like a normal subtraction symbol. In the latter case, it needs a number/variable to come right after it. In this case, it acts as a negative symbol. Example: input: "1-5"; output: "-4".

3.3.3 Multiplication

The multiplication operator is indicated by the symbol "*". It is a binary operation (requires 2 operands). Example: input: "5*3"; output: "15"

3.3.4 Division

The division operator is indicated by the symbol "/". It is a binary operation (requires 2 operands). Example: input: "3/5"; output: "0.6"

3.3.5 Factorization

The factorization operator is indicated by the symbol "!". It is an unary operation (requires 1 operand). Factorization can only be done with a non-negative integer. Example: input: "3!"; output: "6"

3.3.6 Exponent

Exponent is a binary operation (requires 2 operands). It is indicated by the symbol "ˆ". For simplicity, the base can be non-integer while the exponent is an integer. Example: input: "3ˆ5"; output: "243"

3.3.7 Parentheses

Parentheses are indicated by the two symbols "(" and ")". They help organize and group operations together in order to perform them in a different way. In the program, the first parenthesis must be an opening one "(" while the last parenthesis must be the closing version ")". Furthermore, they must come in pairs and logically ordered. For example: input "(1+))(2)" is considered invalid. A valid example should be: input: "(1+3)/2"; output: "2".

3.4 Decimal point

The program must be able to process decimal numbers. Decimal points (indicated by ".") must be used in order to represent any kind of non-integer numbers. Example: Valid decimal numbers: "0.9", "12.22345"; Invalid decimal numbers: "M.2", "2.M", ".M", ".2", "2.", "M.".

3.5 Memory

The program should be able to save the result of the last calculation into the memory. This last result can be reused in the next expression in the place of the symbol "M". "M" represents a number but only as a variable, not the number itself. Basically, "M" can't be used in situations such as "M.1", "1.M". When first run, if the user uses the "M", it will have the value "0".

3.6 Log file

Aside from printing everything to the terminal, the program needs to write out the user input as well as the result of all the expressions in one session into a file named "calc_log.txt".

3.7 Test cases

Test cases will not be provided to you. You will have to use your own physical calculator and test the program yourself.

3.8 Report

A report about your work should be made for this assignment. Images of test runs as well as description of the functions and logic are anticipated. Flow charts and other visualization of your ideas are also encouraged. The report must contain basic information such as name, ID, class, and subject.

4 Submission

Students are requested to submit the MIPS program(s)/source code (.asm files) and the Assignment report to BK E-learning system (BKEL) earlier than the last lab session of your class. Assignment must be done individually.

Students have to demonstrate program(s) on MARS MIPS during the last lab session.

Students who do not show up during the demonstration time will get 0 for the assignment.

The report should not contain code. Instead, students should present the algorithms as well as the idea in your implementation.

There will be no deadline extensions.

5 Plagiarism

Similarity less than 30% in MIPS code is allowed. In other words, you will get 0 for assignment if your answers are similar to another student's more than 30%. Note that, we will use the MOSS tool developed by Stanford for checking similarity (https://theory.stanford.edu/~aiken/moss/).

6 Rubric for evaluation

6.1 Friendly interface - 1 point

Students can design and implement an amicable user interface so that users can use easily without any confusion (1 points).

Students can design and implement a friendly user interface; However, some parts of the program caused confusion (0.5 points).

Student can implement the program but with little to no effort to make the interface match the description (0.25 points).

Student can implement the program but fundamental prompts and format are not available (0 point).

6.2 Application implementation - 6 points

Students can implement an excellent application without any errors found (6 points).

Students can implement a good application with some minor errors, user doesn't need to restart the application to continue (4.0 - 5.5 points).

Students can implement the application with some errors that prevent players from using the calculator (2.0 - 4.0 point).

Students cannot implement the application or lack too many crucial functionalities (0 - 2.0 points).

6.3 Report - 3 points

Students write such an excellent report that others can understand without any difficulty (3 points).

Students write a good report but it's quite simple or lack of some information to clarify the implementation (1.5 - 2.5 points).

Students write a report with a lot of code embedded without any explanation (0.5 - 1.0 points).

Students with no report submitted (0 point).

References

[1] Wikipedia, "Calculator." [Online]. Available: https://en.wikipedia.org/wiki/Calculator