Computer Architecture and Mobile Processor

Project #1

Simple Calculator

디나디아차리스타찬드라

Nadia Calista Candra The

Dankook University

2021 Spring

Freedays left: 5 days

1. Project Introduction

Making a simple calculator that can support 10 registers and perform basic arithmetic binary operations (addition(+), subtraction (-), multiplication (*) and division (/)). It will store the result in R0 and supports register move operation (M).

2. Motivation

To understand how operations are done by the CPU by seeing how the calculator operates. Furthermore, being able to see CPU's behaviour through the implementations. Not only that, to get used to the linux environment which is different from what is used in the past projects.

3. Concepts Used in CPU Simulation

ISA (Instruction Set Architecture) is used in the CPU simulation.

Simple input (text file) :

- Has 10 registers and 5 instructions
- One instruction per line
- Instructions are in a string format

File operations:

- Files are opened using fopen and closed using fclose
- Using getline to read the contents inside the file

String operations:

- Using strtok to split the strings into a series of token base on a delimiter (space)
- It is used to differentiate operands and operators from string

Atoi function:

- Takes string as an argument and returns it as an int value
- It is used to calculate between operands

4. Program Structure

The program runs in a loop in the main function.

```
fopen ()
+------------------------------------
|       while ()
|               if(expression 1)
|               if (expression 2)
|               if (expression 3)
+-----------------------------------
fclose()
```

In each stage, every if statement will print out the result in the screen. Expression 1 will take every hexadecimal input (which is already converted into an integer using the atoi function) from the file and will go inside another if statement to determine which operator it is. After determining which operator it is it will go into a statement and will print the result. The same goes with the other expressions (2 and 3) the only difference is that for expression 2 it will detect the operator ‘M’ and for expression 3 it will detect that both of the operands are registers. It will both (expression 2 and 3) go into another statement to determine which operator it is.

5. Problems and Solutions

At first, I wrote this program without knowing clearly what strtok did and did not know how to define the delimiter. This causes my string to not be able to be separated with one another. This problem leads to another problem which is not being able to differentiate which one is the operator or operand. However after understanding how strtok works, I am able to differentiate between operand and operator.

Not only that, I came upon a problem when wanting to differentiate a hexadecimal and a register, I tried combining different statements and puts it inside the same if statement but it doesn’t work, so I decided to make a different if statement for each category (registers, hexadecimal, and ‘M’). I also discovered a problem while debugging that says my atoi function is returning a value when it should return NULL. I got stuck on that problem for a while until I resolve it with using a break statement after the printing the result so it will return to a NULL value.

6. Build Environment

Compilation: Linux Assam server with GCC

File (input.txt) is in the same directory as calc.c

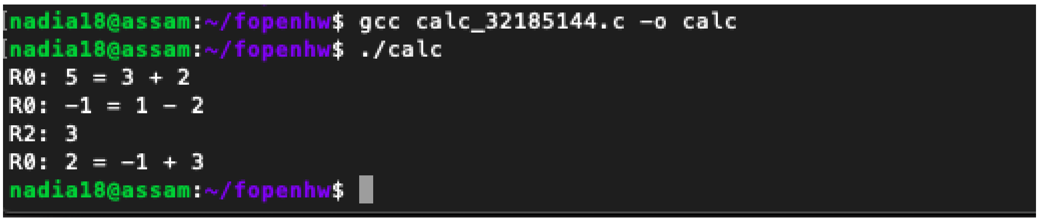
To compile:

gcc calc_32185144.c -o calc

To run:

./ calculator

7. Screen Capture



```
nadia18@assam:~/fopenhw$ gcc calc_32185144.c -o calc
nadia18@assam:~/fopenhw$ ./calc
R0: 5 = 3 + 2
R0: -1 = 1 - 2
R2: 3
R0: 2 = -1 + 3
nadia18@assam:~/fopenhw$ 
```

8. Personal Feelings

I feel okay after finishing this project and submitting it on time. Not only that, I am happy that I am able to learn in the Linux environment which I never used before. From this project I am curious on how to make this simple calculator in a way that is different with how I made it. I am also curious on how to implement other operands other than ‘M’ like Jump or Branch. If there are any chance to learn how to implement it in the calculator I would be really delighted to learn about it.