

2024 Fall CPSC 240

Assignment 6 Pure Assembly

Introduction

The application purpose of this assignment is to compute the resistance in an electric circuit.

The education purpose is to gain experience programming in pure assembly.

The term pure assembly is used by some computer scientists, but not by all. In brief the term refers to making programs that use the native assembly language of the micro processor, but do not use other supporting languages like C and C++. In a certain sense pure assembly is the opposite of hybrid assembly.

Application background

In electricity there are units of measure like these:

Quantity	Symbol	Unit	
Power	P	Watt	
EMF	V	Volt	//EMF = Electromagnetic Force
Current	I	Amp	
Resistance	R	Ohm	
Time	T	Second	
Work	W	Joules	

One of the formulas used in electricity is $R = F/I$ //I is upper case I; F = EMF

Make an application program that will input EMF and I, and then output R.

Continue next page

Sample dialogue

Welcome to Electricity brought to you by Henry Finkelstein.
This program will compute the resistance in your direct current circuit.

Please enter your full name: Roberto Gonzalez Macias
Please enter the career path you are following: Game developer
Thank you. We appreciate all game developers.

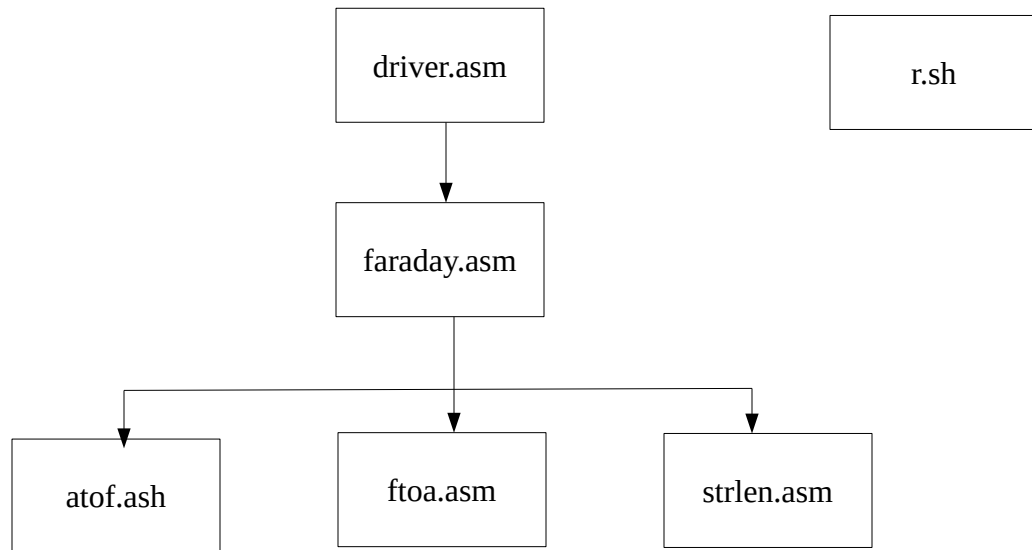
Please enter the EMF of your circuit in volts: 24.5
You entered 24.500000
Please enter the current in this circuit in amps: 7.2
You entered 7.200000
The resistance in this circuit is 3.402778 ohms.
Thank you Roberto Gonzalez Macias

The driver received this number 3.402778, and will keep it until next semester.
A zero will be returned to the Operating System

Footnotes: It may be difficult to append an 's' to the person's career path. Try your best.

Color codes
Yellow: drive
Green: Electricity
Pink: Keyboard

Traditional calling structure



You will recognize this diagram as a common calling structure for a program of 5 files. You have experience with these kinds of calling structures earlier in this course.

Simplified calling structure using one macro

We had a lecture on the use of macros as a storage place for blocks of code. Here is an opportunity to try one (or more) macros. Macros have advantages over functions. Recall these facts from the lecture.

- Macros do not make activation records

- Macros do not backup the GPRs and do not restore GPRs

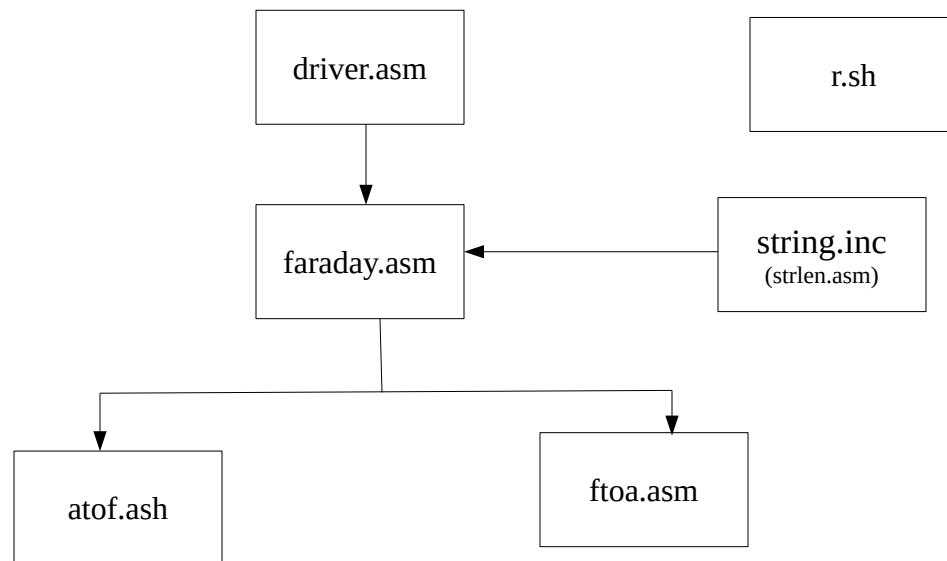
- Macros have no ret instruction

- Macros can accept parameters of different types

One caution about macros

- They use the same registers as the calling function. The programmer should be cautious when selecting registers to use inside the macro.

Suggestion. This is optional. Learn how to use a macro. Take one of the three functions `atof`, `ftoa`, or `strlen`. Let's you picked `strlen`. The calling diagram may look like this.



Now you can see that `strlen` is inside of the file `string.inc`, and that file is included to `faraday.asm`. Using macros is not required for Assignment 6, but it may be required for the final programming test.

Conclusion

Due date for Assignment 6 is **Dec 17 at 12:00noon**. This is in the midst of Finals week. You should try to finish before you begin serious study for the last test.

Make your source file appear professional as if you were going to submit them to the employment officer at the company where you most prefer to work.

Don't omit a source file by mistake. Don't forget the bash file. Missing parts of a program usually result in a zero score.

Do no submit pdf files. Is that easy to understand?

Do not sent a link to your program posted in the cloud. I will not try to download an file stored in a cloud. This is a serious matter. Don't contact me about files you left in a cloud. I am not going to read them or attempt to download them.

Send thie complete set of files belonging to this program to the usual place.

cpscgrader@proton.me

The grader and I both have access to the proton mail site. We will determine how to divide the work between us.

Record your name and preferred email in the professional area of your source files.

In the subject line place: "**Cpsc240-xx Assignment 6**" where, of course, xx is your section number.

If you make a program that satisfies part of the requirements of Assignment 6, then you will receive proportional credit for part completed. There is nothing wrong with partial credit.

There is a whole lot wrong with program that do not compile or crash during execution. Please don't submit those programs. They will receive zero anyway, so why send a zero quality program for review?