

# 2021 Fall CPSC 240

## Practice Assignment 4 Power Unlimited

### Description

An EE major is in the lab configuring a circuit for experiments. This is a direct current (DC) circuit with a single battery as power source and an single device such as a bulb or a fan as consumer.

That woman student can measure the resistance  $R$  of the consuming device and the current flowing from the battery  $I$  (an upper case  $I$ ). She contacted your for a program that will compute power with the circuit.

Here are the quantities.

$R$  = resistance of the device measured in ohms

$I$  = current from the battery measured in amps

$P$  = power consumption measured in watts.

The equation that applies is  $P = I^2 \times R$ . Your job is to make a program that accepts  $I$  and  $R$  and computes  $P$ .

### Dialog with clean data inputs.

Welcome to Power Unlimited programmed by Janice Gutierrez.

We will find your power.

Please enter your name. You choose the format of your name: Henrik Hertz, Software Engineer.

Welcome Henrik Hertz, Software Engineer

Please enter the resistance in your circuit: 4.5

Please enter the current flow in this circuit: 10.25

Thank you Henrik Hertz, Software Engineer. Your power consumption is 472.78125 watts.

The main function received 472.78125 and will keep it.

Next zero will be returned to the OS. Bye

### **Dialog: Invalid data**

Welcome to Power Unlimited programmed by Janice Gutierrez.

We will find your power.

Please enter your name. You choose the format of your name: Henrik Hertz, Software Engineer.

Welcome Henrik Hertz, Software Engineer

Please enter the resistance in your circuit: 45.Y7X6

Invalid input detected. You may run this program again.

The main function received -1.0 and will keep it.

Next a zero will be returned to the OS. Bye

### **Dialog: invalid data**

Welcome to Power Unlimited programmed by Janice Gutierrez.

We will find your power.

Please enter your name. You choose the format of your name: Susan Gupta, Senior Programmer

Welcome Susan Gupta, Senior Programmer

Please enter the resistance in your circuit: 6.5

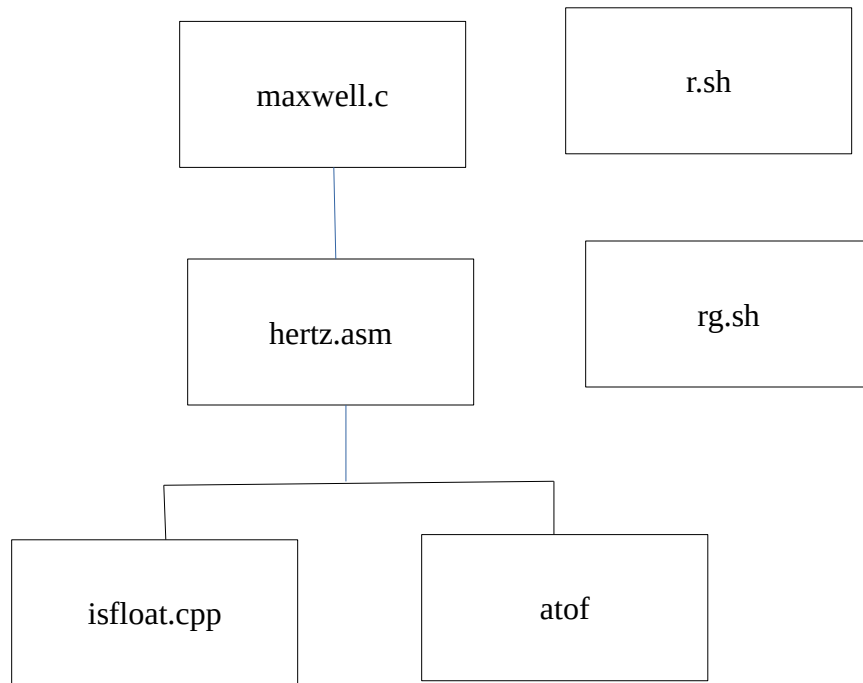
Please enter the current flow in this circuit: 10.55.25

Invalid input detected. You may run this program again.

The main function received -1.0 and will keep it.

Next a zero will be returned to the OS. Bye

## Program structure



maxwell.c is the usually main function. Put it into C language.

hertz.asm is the function that performs the main activities of this program. hertz.asm does all of the following.

- Declare two arrays of char in the .bss segment

- Inputs a resistance number into the first array of char (first string)

- Calls isfloat to validate that recent input

- If the input was valid hertz calls atof to convert it into a number placed into a safe xmm register.

- Inputs a current number into the second array of char (second string)

- Calls isfloat to validate that newest input

- If the input was valid hertz calls atof to convert it into a number placed into another save xmm register.

- hertz function arithmetically computes power.

- Calls printf to output power

hertz function returns a copy of the computed power to maxwell.

Maxwell outputs this received number.

Special handling: if at any time isfloat returns a false value to hertz then hertz does not compute power. But rather, hertz may display a message “run the program again” or “try again to input a valid number”. As programmer you decide the kind of message to display.

isfloat.cpp is commonly shared among students. I have been told that there are lots of copies of it posted in many places. Students have told me it is posted in Discord with instructions how to use it. Since it is written in C++ you can easily view its source code and verify what is its true prototype.

atof is found in the C++ standard library called cstdlib. To use atof in a C++ function do the following

```
#include <cstdlib>
using std::atof;
.....
char myinput [ ] = "34.752";
double mynum = atof(myinput);
```

I am fairly sure there is a C library which also contains atof, but I did not look it up.

But we are fortunate. We asm programmers do not need to know the name of the library where atof is stored. We simply declare atof as extern at the top of hertz.asm. Then the C++ linker will find it.

Last comment about linking. According to what I have experienced the C++ linker known as g++ searches in libraries of both C and C++. However, the C linker known as gcc searches only in libraries belonging to the C language. I don't have documented proof of this, but that is my experience working with both C and C++.

## Running the program.

Make a bash file to run the program in manual mode. You've done this in previous assignments.

Make a second bash file to run the program in GDB mode. You did this in Assignment #3.

Here are the commands to be tested in GDB mode.

Set a break point at "main".

Additional break points are optional.

Run the actual program with the r command.

Step forward slowly until you see that the call to the hertz function is the next statement to be execute.

Show the value in rsp

Step forward into the hertz function until your are asked to enter your name. Enter your name and press enter as usual.

Use a gdb command to view the name string declared in the .bss segment. You should see your own name appear.

Step forward until the value for "power" has been computed. Use a gdb command to display the value stored in the register holding the power.

Use gdb commands to complete the execution. Probably the 'c' command is suitable at this point.

Sometimes gdb will ask a question if you really want to quit. Of course you answer y.

Use the q command to quit.

We are only interested in three outputs from gdb. The three outputs are highlighted in yellow. All of the rest of the input.txt file is navigation. You are free to set the navigation in any way that works best for you.

Next. Put all of those commands in a text file with a name like input.txt. You choose a name.

Run the program with the command below

```
./rg.sh < input.txt          or      sh rg.sh < input.txt
```

Now you are done. Have some snacks and drinks to celebrate your success. Most engineers have junk food near by when designing new stuff.

## **Answering the mail.**

During the most recent week or two weeks I have received a lot of questions by mail asking about placing quotes on input data.

If the program says “Please enter your name”, then you should answer without quotes.

If gdb asks you “do really want to quit y or n?”, then you should answer y or n without quotes.

In fact, I cannot remember a single time in my whole life where I placed quotes around input to gdb. There must be a legitimate use of quotes someplace, but I can't find that usage.

**Suggested date** for submission: October 21, 2021.

Submission is optional. These are practice assignments. They serve as practice for the next midterm.

If you would like me to review your program and send back my opinion, do feel free to send the source files to for review. It is good for you to hear the feedback from a neutral party. You know if you are on track.

## **After thoughts**

Why learn gdb? The answer is in large scale software projects. Large scale means 50 or more software engineers creating a program with half million lines of executing statements. If the project is human life critical then it has to be 100% perfect in its outputs. The software development team will use a debugger tool to step through the whole program tracing every value from its initial input to its final use. They will use a debugger.

Why use this debugger because there are others? There are two key reasons: GDB is widely used in the industry and second it is free of charge. I don't want you to buy software in order to do assignments.

Put gdb on your employment resume. It is a valuable skill.