Rust Programming Project Proposal

Nicholas Scalzone
scalzone@pdx.edu

Project name: Rusty-Pipes

Topic Area: Engineering software

Project Vision: When I worked as a mechanical engineer, I used a program called Pipe-Flo Professional to model fluid systems. My goal is to recreate the basic functionality of that program, but with command line inputs rather than a GUI (per Professor Massey's recommendation).

I plan to implement a program that essentially calculates segments of pipe "node to node", with the start node requiring pressure, flow rate, and fluid data. From there, the Darcy-Weisbach equation will be used to calculate the pressure at the other end of any pipes attached to that node, at which point the end nodes of each of those pipes will now have pressure and flow data. After that, rinse and repeat for pipes connected to those nodes, until the full pipe system has been modeled.

For the user interface, I'm thinking a simple command line for user inputs, with options the user selects such as:
A - View current system elements (will output table showing nodes, and pipes connected to them)
B - Add elements
C - Calculate current system (will output results in table format)
D - Revise start node pressure and flow
S - Save system to file
I - Import previously saved system
Q - Quit

A basic idea for how the table would look is:

Pipe	Start Node	End Node	Start Pressure	Starting Flow Rate	Pipe Size	Pipe Length (ft)
Pipe1	Node1	Node2	45PSI	20GPM	2"	40
Pipe2	Node2	Node3	-	-	1"	10
Pipe3	Node2	Node4	-	-	1 ½"	3
Pipe4	Node4	Node5	-	-	1"	3

In this model, we have one 2” pipe, which branches into a 1” line and a 1 ½” line, which reduces to 1” after the first 3ft of length.

From a programming standpoint, I think the biggest challenge will be figuring out how to handle cases like Pipe3 above, where the flow that a 1 ½” pipe could contain will be different than the flow that a 1” pipe can, so the program will need to look ahead to some extent when it makes calculations.

The save/import seems like it could be tricky as well, but I think as long as the only thing you can import are previously saved files that should take the challenge out of formatting the files for the user, and should make the import smoother on the programming side.

From a data structure standpoint, I think there should be a Node class, since the calculation goes node to node. Pipe should be a struct, and each node can have an array of pipes (the pipes leaving that node) as data members, along with the pressure and flow data.

Assumptions:
For the purposes of this project, I will not be adding in fittings, valves, and other hardware that result in increases in head loss in a pipe system. I will start with assuming all pipes are Schedule 40 Carbon Steel to avoid having large amounts of pipe size and material roughness data to sort through and store. In the future, I may expand to include plastic (PVC, etc.), aluminum, copper, and other such piping materials commonly used in industry.
I will also start with the assumption that the fluid is water, at 70 degrees Fahrenheit. All of these are inputs that a real engineer would need to verify, and are variables I’ll need to input into the equation, so they may be something I can incorporate easily depending on how the implementation of the basic structure goes.