

Programming Project Progress Report

Sean Canavan Nick LeClair Steven! Lorenzen

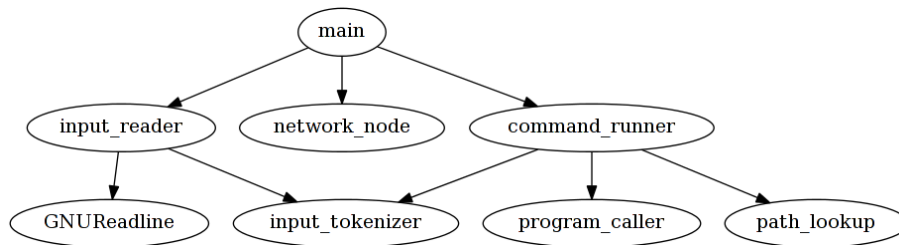
April 4, 2012

1 Modules in sh142

Our implementation of sh142 will include multiple modules broken up by task responsibility.

Our main modules deal with input reading, program execution, and networking. Each of those main modules has submodules for more finely grained implementation details.

The figure below details our dependency graph.



2 Main Module APIs

2.1 input_reader

`char* readline(char*)` A thin wrapper around GNU Readline.

2.2 input_tokenizer

`char*[] tokenize(char*)` Break the input up into individual tokens separated by whitespace. Quoted bodies are kept together with quoting stripped.

2.3 command_runner

`void invoke_command(char*[])` Invoke the tokenized command as a sh142 command. This is the stage where input will be parsed before being delegated elsewhere.. for-loop expansion and PATH and DATA lookup will also be substituted here.

2.4 path_lookup

`char* resolve_data(char*)`

Test for presence of filename in the specified DATA path. If, present return the absolute path to the file otherwise NULL.

`char* resolve_executable(char*)`

Test for presence of filename in the specified PATH variable. If present return the absolute path to the file otherwise NULL.

2.5 program_caller

`uncmd(char* cmd, char* argv[])`

Run the command with the given arguments return the exit code of the command.

`uncmd(char* cmd, char* argv[])`

Run the command with the given arguments in the background, return the pid of the created job.

2.6 network_node

TBD

3 Work Plan

4 Test Plan