CST 334: Operating Systems

Dr. Glenn Bruns

# MSH version 1

**Purpose**: This is the first of several assignments in which you develop your own shell. The purpose of these assignments is to give you practical experience in using the Linux process API and a good understanding of how the bash shell works.

**Instructions**: You will write your own, bash-inspired command-line interpreter called msh. The code for a command interpreter usually has this form:

initialize

while (true) {

print prompt

accept user input

process input

}

I have provided starter code. Copy file /home/CLASSES/brunsglenn/cst334/hw/hw2/msh1.c on mlc104 to a directory of your own. Compile and run the code. When you run it you will see the prompt 'msh>'. If you enter text at the prompt it will do nothing but will print the prompt again. If you enter 'ctrl-d' the program will exit.

Your job is to extend the code in the following ways:

* + split the input string s into an array of strings (array toks[])
  + if there are no tokens, just prompt for more input
  + if the first token is 'exit', exit the program
  + if the first token is 'help', print some help information
  + if the first token is 'today', print today's date
  + otherwise, print the tokens that were input

Here is an example of what your msh code should do: ($ is the bash prompt)

$ ./msh1

msh> help

enter 'help', 'today', or 'exit' to quit

msh> today

01/18/2020

msh> exit

$

$ ./msh1

msh> wc -l

token: 'wc'

token: '-l'

msh> ls -t

token: 'ls'

token: '-t'

msh>

Implementing the 'exit' and 'help' commands is easy. There are two parts to this assignment that will take a little thinking:

1. splitting the input string s into tokens array toks[].

You must use function strtok() or strtok\_r() for this. The man page for strtok() is scary, but using strtok() is not hard. To get the first token in string s, you call strtok(s, " "). The second argument means that s should be split into tokens based on space characters. To get the second token in s, you call strtok(NULL, " "). If will return NULL if there is no second token. To get the third token in s you call strtok(NULL, " ") again. In summary, you provide the string s in the first call but NULL in later calls, and keep calling until NULL is returned.

1. implementing the 'today' command

You must use Linux commands time() and localtime() for this. Try ‘man 2 time’, and ‘man localtime’. You will first call time() and then call localtime(). It’s a little tricky. You must use system calls or C library functions to get the time, not user commands (look at the man page for 'man' if you don't know the difference).

For these two parts, I recommend you write separate small test programs to make sure you understand how to use the calls.

Please do not just immediately google for solutions on how to use strtok() and how to use time() and localtime(). Try reading the Linux man pages first and writing some test code. Google around if you get stuck.

Important things to remember:

* only change msh1.c where you see comments YOUR CODE HERE
* you must use strtok() or strtok\_r() for splitting input string s into tokens, and the tokens must go into array toks[].
* you must use time() and localtime() to implement command 'today'.

**Testing your code**. On mlc104, the directory /home/CLASSES/brunsglenn/cst334/hw/hw2 contains five test files test1.sh, test2.sh, …, test5.sh. Copy these to the directory where you developed your file msh.c. Each test should give exit status 0, like this:

$ ./test1.sh

$ echo $?

0

You need to run test1.sh first, as it will compile your code and produce binary file 'msh' that is used by the other tests. The directory above also contains files 'Makefile' and 'run-all.sh', which you can look at and play with if you want.

**Submission** : Submit your msh1.c file on iLearn. Here are instructions on [copying files to/from mlc104](https://docs.google.com/document/d/1TY6waCCHFkSBYCWKsHUelg9KH2yjXS2VOAyZLIPu93U/edit?usp=sharing).

**Grading**. 40 points for functionality and 10 points for tidy code. For functionality, I will run 6 automated tests on your code. For tidy code, I will look for proper commenting, indentation, and use of white space.