CS551 Project 1 Document and Readme

Author: Yating Zhou, Lun Li, Guangwei Ban
CWID: A20379635, A20359332, A20381640

February 27, 2017

1 How to execute and use mshell(msh)?

Suppose the source code(mshell) is installed in /usr/src/mshell, in order to install and execute it, we need the following steps:

```
# cd /usr/src/mshell
# make
```

Now msh has been build succesfully in current directory(./), simply execute it by the following command:

```
# ./msh
```

2 Load profile and jump into home(/root)

Your shell shall first execute a PROFILE file which defines the PATH variable that will allow you to access programs provided in /bin and /usr/bin. Once the PROFILE file is executed, you will be in a HOME directory specified by you in the PROFILE file. The PATH and HOME variables do not replace those of the Ash shell from which your shell is instantiated.

The PROFILE file is called profile, it is in mshell folder. and we can load the param in PROFILE by the following:

```
# loadprofile [path of profile]
```

We can modify the home directory by modifying specified param. This part is implemented by file mprofile.c in read_profile function, the PROFILE file is gotten by stream.

3 Problem 3 Solution

Firstly, the command is input by main() function. Then it is parsed by precedence_parser() function and executed by function execcmd() in mshell.c. This process includes sequence and parallel processing and symbol ; & and() recognition. So it can execute complex command.

4 Problem 4 Solution

In order to set up an alarm during process running, we need to call waitpid() to hook the running process(in child), and count the child process's running time. kill(pid, SIGKILL) to kill the process if user want to kill it after waiting too long(5 second). But pay attention, we also need isValidateCMD(argv) to detect if this cmd is validate. Otherwise it will cause zombie. Simply use sleep(time) can check if this function work or not. If

```
# sleep 5
```

give us prompt: Do you want to terminate? [y/n]: and when we type in y, we back to terminate soon, then then the monitor function works. Now use

```
# sleep 4
```

and there will be no more prompt asking you to terminate. This indicate the monitor and terminate function work as expected[A more detailed test has been discussed in test data].

Note: default setting alarm is disable, you need to turn it on by two method:

- change setAlarm=0 to setAlarm=1 in ./profile

or

- executing command init_alarm to turn it on.

5 Problem 5 Solution

When user enter UP or Down arrow key, the main() function will trap it and then enter history() function in history.c. By this operation, the shell will load last command or next command stored in save[] array. At the same time, if you enter one command ,it will be stored in save[] array by save_cmd() in history.c.

When user enter TAB key, the main() function will detect it too and then search the command list to match with what you input. If there is only one command matches, it will be filled and you can execute it. If there are more than one commands matching the list, it will show out all the command.

6 Problem 6 Solution

Operator precedence is following as Wu-Hong Leung's per guide. Which means parentheses has the highest priority. Semicolon and ampersand have the same precedence expect semicolon execute left command and right command one by one, ampersand operator indicates both left and right commands are executing synchronously.

Hence, parentheses should be first parsed. If left parentheses occurs, we push the command string to stack one by one. Once the right parentheses occurs, we pop the command string from stack one by one.

Once semicolon(;) occurs with the form(leftcmd ; rightcmd), then we first place leftcmd in child process, parent process just wait child process to finish. When child process finish, rightcmd will be executed as design.
If amphersand(&) are parsed with the form(leftcmd & rightcmd), then we need to place leftcmd in child process and rightcmd in parent process, then only difference between (;) and (&) is that & do not need to wait child finish. More detailed test cases combining parentheses, semicolon and ampersand are discussed in test data.

7 Extra credit

- Implement Crtl + C exit function. When user want to terminate msh(mshell), it's pretty simply to use combination keys: Ctrl + C, then msh will ask you whether you want to exit or not.

• Enhance init_alarm() routine. User can simply type in init_alarm(0) or init_alarm(1) to turn off or turn on alarm monitor. But, when user type in an invalidate value, our version of init_alarm() will give user chance to select the desire setting he/she wants.
• There's a lot of patches to ensure our msh as robust as we can. For example, when we execute invalidate commandsthen isValidateCommand will prevent msh exhausting all the resources from system. This help to make our msh robust. You can search syscl as keywords to see similar patches to make our msh as robust as possible.