ECE 695 Project 1

Xin Tang

Overall Design:

There are mainly three part of the program.

cmdline.c: This part is used to parse the user input to command strut.

myshell.c: This part is used to execute the command

main.c: this is the entry of the program. main function connects the other two parts. In addition, the main function can monitor the status of the child process to avoid zombies.

When user type in a command string in the shell, the main function gets the string and send to cmdline part. Cmdline use this string to construct the command token by token. After that, a list of command structs is created and send to myshell part. The myshell part will fork a child process to execute this command list.

Noteworthy Features:

1. Zombies will be cleaned as soon as they are created. This feature is achieved by using 'SIGCHLD' and a handler. The handler only contains waipid() statement with option 'WNOHANG'. The stopped child will send a signal to the parent and the handler will kill the stopped the child.
2. The program can report the background process when it is created. The background command will call 'report_background_job' to report its job id and process id.
3. Checked all the allocate memory and tried my best to avoid memory leak.
4. Fulfill the requested 'cd' error message.
5. Passed all the test cases.

Future Improvement:

This part is mainly focused on background process monitoring. There are several shortcomings for the current background process monitoring

1. In order to report the background process, one global variable, job_id, is created.
2. The job id will add 1 for each background process and never go back. However, for the real bash, when all the background process terminated, the job id will be reset to 1.
3. Cannot report the stopped child process like read bash. For example: [1] + Done Sleep 1.

I think all these problem can be improved by using a hash table. The hash key is the pid and the hash value is the command and job id. When the parent receive a stopped signal, catch the stopped pid by waitpid() function. Then I can get all the information about this command from the hash table and report this stopped process. All these things will be done in the handler. In addition, when the hash table is empty, reset the job id back to 1.