1. Why does the cd command have to be executed internally (i.e., in the parent process)?
2. Try exiting from your shell by entering an end-of-file (EOF) character, i.e., Control-D. How does this differ from entering the quit command?

The quit command exits the shell through the program (internally). Using an EOF character exits the shell via an external command coming from the user.

1. Try running a command — that produces printed output — in the background. Why is it a good idea to redirect standard output and standard error in this case?

When a process is running in the background, it usually means another process is running in the foreground.

1. Write and test a simple shell script for your program, e.g., consisting of a few basic commands like cd, cat, and ls and ending with quit. What is the difference(s) between running your script by typing mysh *script\_file* and mysh <*script\_file*?

With the former method (mysh *script\_file*) passes the script in as a command line parameter to the shell. This means the shell would have to read the file to retrieve the commands to be given. On the other hand, the latter (mysh <*script\_file*) pipes the data from the file into the shell from stand output; as if it were typed by a user. This method operates the same as if a user typed several commands at once and no changes to

1. What steps are necessary to turn a script file written for your shell into a standalone command (i.e., one that you can execute just by typing its name)? (Make sure that it’s your shell that’s executing the script and not some other shell like bash or /bin/sh by making “quit” the last command in a test script; bash and /bin/sh should not recognize “quit” as a valid command.)