Homework 7 - Simple Shell

Due by 5:00 p.m. Thursday, 12/4/14

For this assignment you will extend your shell/command-line interpreter from homework 3. The new functionality will include executing commands in the background using the `&` symbol; redirecting the standard input, standard output and standard error streams to or from files using the `<`, `>` and `2>` symbols; redirecting and appending the standard output and standard error streams to files using the `>>` and `2>>` symbols; and piping the standard output of one process to the standard input of another using the `|` symbol.

Here are some sample executions showing the required functionality:

- Execute a command in the background:
  ```
  Myshell> cp bigFile bigFileCopy &
  ```
- Redirect the standard output of a command to a file:
  ```
  Myshell> ls > outfile
  ```
- Redirect and append the standard output of a command to a file:
  ```
  Myshell> ls –l >> outfile
  ```
- Redirect the standard error of a command to a file:
  ```
  Myshell> ./a.out 2> error.log
  ```
- Redirect and append the standard error of a command to a file:
  ```
  Myshell> ./a.out 2>> error.log
  ```
- Redirect the standard input of a command to come from a file:
  ```
  Myshell> grep disk < outfile
  ```
- Execute multiple commands connected by a single shell pipe. For example:
  ```
  Myshell> ls –al /usr/src/linux | grep linux
  ```

To create background processes, you can simply omit the call to `wait` or `WaitForSingleObject` in POSIX and Win32, respectively. To redirect standard input, standard output or standard error to or from a file, you will need to open the file with the appropriate access mode and then replace the target stream with the file descriptor. This can be accomplished using the `close` and `dup` families of functions in POSIX and by altering the `hStdInput`, `hStdOutput` and `hStdError` attributes of the `STARTUPINFO struct` in Win32. To create a pipe, you can use the `pipe` or `CreatePipe` functions in POSIX and Win32, respectively, and then redirecting the standard input and standard output streams to the pipe handles in the same way as you did with file descriptors.

You should submit your source code files (one for each platform) and a short writeup in pdf format that includes a description of what you did and the compilation and execution output from each of your programs. Demonstrate all required functionality, and include some commands with multiple command-line arguments. For each file you use to demonstrate redirection, print the contents of the file on the command-line using the `cat` or `type` commands on Linux and Windows, respectively. (You can run `cat` from within your shell because it is an executable file, but you will need to exit your shell before running `type`.) Submit everything to the regular submission link on iLearn, and then submit just the writeup to the TurnItIn link to generate an originality report.