2.21 Basic Shell Exercises

What should I focus on?

You should:

- clearly understand what redirection is, and what a pipe is
- understand that it is the *shell* that expands the filename generation symbols (`*`, `?` and `[...]`), and not `echo`, `ls`,...
- understand the differences between these three ways of quoting: `'...'`, `"..."`, and quoting a single character with `\`
- understand how to put a process into the background, bring it to the foreground, and how to interrupt it (*not* with `Control-z`!)
- understand the difference between using `Control-c` and `Control-z`
- understand that subshells have their own working directories.

The exercises

1. *Redirection*

   (a) Try typing the following commands exactly as they appear here. Refer to sections 2.6 on page 39 to § 2.9 on page 42 first. Also refer to section 2.13 on page 46 before doing the last exercise.

   ```
   $ cat > newfile 2> newfile.error
   $ car > newfile 2> newfile.error
   $ car
   $ Space Shuttle
   $ echo
   $ echo foo
   $ ech
   $ ech foo
   $ cat > newfile 2>&1
   $ car > newfile 2>&1
   $ cat < newfile
   $ echo foo | cat > newfile 2>&1
   $ ech foo | cat > newfile 2>&1
   $ echo foo | car > newfile 2>&1
   $ (ech foo | cat) > newfile 2>&1
   ```

   After each step, examine the contents of `newfile` and `newfile.error` by typing:

   ```
   $ cat newfile
   $ cat newfile.error
   ```

   Make sure you understand what happens in each case, ask the tutor if you are not sure.

2. *Filename expansion and Quoting*

(a) Do the following in the `/bin` directory; refer to section 2.18 on page 51 first.
   i. List all filenames with exactly three characters.
   ii. List all filenames with exactly three characters in which the second character is a vowel.
   iii. List all filenames with a, b, c, or d as the last character.
   iv. Construct a command to print the number of filenames consisting of exactly three characters. (You may combine commands together in a pipeline. You may find the wc utility useful here; check `man wc` for more information.)
   v. Construct a command to print the total number of files with exactly two, three or four characters in their name. (Again, you may find the wc utility useful.)

(b) Compare the effect of the following commands. Refer to section 2.19 on page 52 first.

```
echo $HOME
echo "$HOME"
echo '$HOME'
echo *
echo "*"
echo '*'
echo $HOME/*
echo "$HOME/*"
echo '$HOME/*'
```

(c) Change back to your home directory and try to create a file with the name `*`. Was this a sensible thing to do? How would you delete it? (Be *very* careful!)

(d) Create a file called `--file`. Try to remove this. Use the `rm man` page to help you.

3. *Background processes and `nohup`*

   Refer to section § 2.10 on page 43 to
   $sec:basic-shell-and-tools-approach-background-and-nohup first.

   (a) Start the command `sort /dev/random` in the background in your current shell
   (b) Bring it back to the foreground and terminate it by typing ^C
   (c) Start it again, and once more so that you have two copies running in the background
   (d) Bring them to the foreground and terminate them in the order you started them
   (e) Start the same command in the background, and terminate it using `kill`

4. *Grouped commands*

   Compare the following command sequences, and make sure you understand the differences. Refer to section § 2.13 on page 46 first.

   (a)
```
$ cd /tmp
$ cd /usr; ls
$ pwd
```
   (b)
```
$ cd /tmp
$ (cd /usr; ls)
$ pwd
```
   (c)
```
$ sleep 5; sleep 5 &
```
   (d)
```
$ (sleep 5; sleep 5) &
```

   Check you can use your history to get at and repeat any of the commands you have typed.

