

Operating Systems: Lab 1 self-testing examples

January 16, 2009

You may find these examples useful when preparing your shell program. However, remember to consult the assignment text in order to understand the potential differences between your program and the existing implementations of other shells.

Simple commands

Try programs that do or do not exist and list the system calls in use. If any of the programs fail, what is printed? Where? What happens to the child process?

```
% date
% hello
```

Commands with parameters

```
% ls -al -p
% ls parser*
```

Redirections with in and out files

```
% ls -al > tmp.1
% cat < tmp.1 > tmp.2
% diff tmp.1 tmp.2
```

Running background programs

```
% emacs &
% emacs &
% emacs
% ps
```

Try to look at the parent process that is waiting for the child process. Run the list of commands several times and use `kill` (or `xkill`) to see after which command it is possible to generate a prompt.

```
% ps
```

Check if there are any zombies created. Notice that the shell program includes a `while` loop that collects the return values of the children.

Process communications (pipes)

```
% ls -al | wc
% ls | sort -r
```

Does the prompt appear after the output of the command?

```
% ls | wc &
```

When does the prompt appear now?

```
% cat < tmp.1 | wc > tmp.3
% cat tmp.1 | wc
% cat tmp.3
```

Compare the two last outputs. Are they the same?

```
% abf | wc
% ls | abf
% grep apa | ls
```

What are the outputs? When the prompt does appear? Use `Ctrl-D` if necessary to let the `grep` finish and to let the shell process take over.

Built-in commands

Try each of the following commands and explain for each why do we need to have them as built-in commands?

```
% cd ..
% cd lab1
% cd tmp.tmp
% cd ..
% cd lab1 | abf
% ls
```

Does the command `ls` work?

```
% cd
```

Is there an error? Use `pwd` to see which the current working directory is.

```
% grep exit < tmp.1
```

Does the shell quit here, or does it consider `exit` as a text string to find in a file?

```
% ..exit
```

And here? (replace the dots with spaces)

```
% grep exit | hej
```

Is there an error here? Does the prompt appears?

```
% grep cd | wc
```

Does an output appear? Does it appear after pressing `Ctrl-D`?

```
% exit
```

Are there any zombies now?

```
% ls | wc-
```

Are there any errors here?