

# Intro to the Dash shell

**Author:** Daniele Pizzolli  
**Contact:** [ors@tovel.it](mailto:ors@tovel.it)  
**Copyright:** 2010 Some Rights Reserved <http://creativecommons.org/licenses/by-sa/3.0/>  
**Location:** Villa Sant'Ignazio — Trento  
**Version:** Draft  
**Note:** Please send your feedback.

## Recap from the previous lessons

- You should be familiar giving some command at the interactive shell prompt

```
$ mkdir tmp && cd tmp
$ ls
$ ls | wc
```

- You should be able to create and manage files

```
$ ls > dir.list.txt
$ cat < dir.list.txt | bzip2 > dir.list.txt.bz2
$ bzip2 dir.list.txt
$ bzip2 -tvvv dir.list.txt.bz2
```

- You should be able to do some text file processing using standard unix tools

```
$ bzip2 dir.list.txt.bz2 | wc -l
$ touch new_file
$ diff -Naur <(bzip2 dir.list.txt.bz2) <(ls) # Bashism(?): Process Substitution
```

## Bash and Dash

As you should remember the history of unix is quite complicated and so is the story of the unix shell(s). Few variants survived and most common, at least on GNU/Linux distributions are [Bash](#) and [Dash](#).

## Some info from man pages

- Bash

```
bash (1) - GNU Bourne-Again SHell
```

```
Bash is intended to be a conformant implementation of the Shell and Utilities portion of the IEEE POSIX specification
```

```
dash (1) - command interpreter (shell)
```

```
dash is a POSIX-compliant implementation of /bin/sh that aims to be as small as possible
```

## What POSIX means? (TODO: english correct form)

POSIX: Portable Operating System Interface

TODO: more on posix

After the unix wars: a try to standardize the shell and standard utilies behaviour.

## What POSIX means to us? (TODO: english correct form)

- Basically we have hardly see any command or syntax that is not POSIX compliant.
- When you break the POSIX compatibility using a bash feature you create a bashism.
- You could use `checkbashisms` from the package `devscripts` to check if your script contain bashisms.

```
/usr/bin/checkbashisms
checkbashisms (1)      - check for bashisms in /bin/sh scripts
```

## Where shell scripting is used?

- Try yourself:

```
find /bin/ /sbin/ /usr/bin/ /usr/sbin/ /etc/ -executable -type f 2>/dev/null | \
xargs file | grep -si 'shell script'

find /bin/ /sbin/ /usr/bin/ /usr/sbin/ /etc/ -executable -type f 2>/dev/null | \
xargs file | grep -si 'shell script' | sort -k2

find /bin/ /sbin/ /usr/bin/ /usr/sbin/ /etc/ -executable -type f 2>/dev/null | \
xargs file | grep -si 'shell script' | wc -l

find /bin/ /sbin/ /usr/bin/ /usr/sbin/ /etc/ -executable -type f 2>/dev/null | \
xargs file | grep -si 'shell script' | grep -i posix | wc -l
```

- If you have more time you could search your entire system for shell script. How?

## Where shell scripting is used? (2)

- To hardcode options and parameter for flexible command.
- To start or stop standard services: look in `/etc/init.d`.

## Why shell scripting is used?

You could start benefit from shell scripting at any level.

Your first shell script could simply be a long long interactive shell command which uses a hard to remember options or parameter list.

During the course we add some complexity to our scripts to achieve more flexible commands.

## Caveats

Unfortunately modern shells still show *subtle differences* and awkward syntax.

The shell flexibility (eg. A variable could be a command, an option, an argument...) hinders the easy writing of secure shell scripts.

## Saving last typed command

In bash:

```
FCEDIT=your_favorite_editor fc
# OR
history 1 > file
# OR
use typescript
```

and then edit your file at least adding in the first line

```
#!/bin/sh
```

and remember to make the file executable with `chmod +x filename`.

## The simplest shell script

```
#!/bin/sh
```

- What is the output?
- What is the purpose?
- ... nothing...

## Another simple shell script

```
#!/bin/sh -e
```

- What is the difference?
- What is the purpose?

## Another simple shell script

```
#!/bin/sh
```

```
ls
```

- What is the difference?
- What is the purpose?

## Another simple shell script

```
#!/bin/sh
```

```
ls
```

- What is the difference?
- What is the porpuse?