Linux development environment
Laboration 3

John-Patrik Nilsson
e-mail: jpatrik.nilsson@gmail.com
Skype: j-p.nilsson

February 10, 2010

1 Abstract

Almost all Unix-based systems use a command-line interface called a shell to provide the most basic interaction with the operating system and the applications. By using text-only as both input and output, this shell might seem as a legacy from the old non-graphical terminals and applications, but the fact is that it is widely used even today when shiny desktop effects seem to be the common norm. The likely reason for this is that it is still the best, and most powerful, tool for some tasks.

Most Linux distributions use BASH (Bourne Again Shell) as the standard shell.

Keywords: Shell, Linux, Unix, commands, bash.

2 Laboration 3

2.1 Introduction

During this laboration the basic usage of the bash shell for administrative purposes was covered.

2.2 Basic commands

All the following inputs are in fact small applications (programs).

Name: cp
Example usage: `cp <source> <destination>`
Description: Copies a given source to a given destination. If the destination already exists and it isn't a directory, the destination gets overwritten.
Example usage: `cp -R <source> <destination>`
Description: Copies the source recursively to the destination, i.e. if the source is a folder all the contents of that folder also gets copied.

Name: mkdir
Example usage: `mkdir <directory>`

Description: Create a directory with the name `<directory>`.
Example usage: `mkdir -p <parent_directory>/<directory>`
Description: Creates the required folder hierarchy if needed.

Name: rmdir
Example usage: `rmdir <directory>`
Description: Deletes (removes) the directory with the name `<directory>` if it the directory is empty, otherwise it exits with an error.

Name: rm
Example usage: `rm <file>`
Description: Deletes (removes) the file with the name `<file>`.
Example usage: `rm -r <directory>`
Description: Deletes (removes) `<directory>` and all its contents.

Name: find
Example usage: `find <path> -name <expression> -type f -print`
Description: Print the names of all files with names matching `<expression>` in the directory hierarchy rooted at `<path>` to stdout, with a trailing newline at the end of each filename.

Name: cd
Example usage: `cd <diretory_path>`
Description: Change the current directory to the given `<directory_path>`.

Name: pwd
Example usage: `pwd`
Description: Print the current directory to stdout.

Name: df
Example usage: `df -h`
Description: Prints the file system usage with user-friendly size formats.

Name: ps
Example usage: `ps -AH`
Description: Displays information about running processes on the system with the same user id as the invoker of the command has. The AH options enables a hierarchy view of all processes on the system.

Name: du
Description: Estimate disk usage of each file in each directory. If no argument is used the root path is the current directory.
Example usage: `du -ch <path> | grep total`
Example description: Pipes the total of disk space used by `<path>` to grep whereby filtering out every output except the total. Handy to check how much disk space a folder consumes.

Name: adduser
Description: Add a user to the system.
Example usage: `adduser --disabled-login <name>`

Example description: Add a user `<name>` to the system with login disabled until the account gets issued a password.

Name: deluser
Description: Removes (deletes) a user from the system.
Example usage: `deluser --remove-all-files <name>`
Example description: Removes (deletes) the user `<name>` from the system, and also removes all files with the same ownership as the user. Handy for clearing the system from random users added during laborations. ;)

Name: whoami
Description: Prints the effective user id.
Example usage: `whoami`

Name: whereis
Description: Used to locate binary, source and manual page file for a command.
Example usage: `whereis -b <command>`
Example description: Locate the binary `<command>` file.

Name: cat
Description: Concatenate files and print their contents to the standard output. Also usable to read the contents of a file.
Example usage: `cat /proc/cpuinfo | grep "model name"`
Example description: How big is yours?
Notes: Do not confuse with "lolcat". [1]

Name: more
Notes: Roger more. Alsa (also), apparently less is more. "More" translated to lolspeak is "moar".
Description: Text pager, one screen at a time.
Example usage: `more <file>`
Example description: If you need this you are probably reading the wrong report.

Name: less
Description: It's like more, but better.
Example usage: `less <file>`

Name: tail
Notes: Cats have tails.
Description: Print the last part of the file(s) to standard output.
Example usage: `tail -f <file>`
Example description: Prints the last part of the file to standard output each time the file is updated (written to). Useful for keeping an eye on log files in

[1] A widespread phenomenon across the Intertubes (*icanhascheezburger.com*) where questionable pictures of cats (and things that might be seen as cats, but really aren't) gets labeled with "lolspeak"; a form of writing where one tries to break as many grammatical rules as possible. — *By this point the effective caffeine level of the authors biological system had far exceeded the recommended levels and thus the proffessionalism of its standard output should be object to critizism.*

real-time.

Name: echo
Description: Print a string to stdout.
Example usage: `echo "meow"`
Example description: Prints meow plus a newline.

Name: which
Description: Prints the path to the binaries which could be executed in the current shell session.
Notes: Similar to `whereis -b`, but there is at least one big difference; `which` searches in the places defined in the shell sessions PATH variable, whereas `whereis -b` searches in the "standard Linux places", whatever that means. (See man whereis.)
Example usage: `which <command>`

Name: wget
Description: A non-interactive network downloader. Wget does not need user presence or even a logged on user to complete the download once started. It support the HTTP, HTTPS and FTP protocols.
Example usage: `wget --no-clobber --input-file <file> --bind-address=<address> --server-response --limit-rate=<mbit/sec>m --random-wait --quota=<quota>m --ignore-length --user-agent=""`
Example description: Download the URLs listed in `<file>`, skipping URLs with the same name as files already present in the folder, limiting transfer rate to `<mbit/sec>`, printing server responses to stdout, and also binding the transfers to one of your active `<addresses>`. The options here also enables wgets ability to wait a random amount of time between each retrieval, to work around evil services that would otherwise reject such programs as wget. The quota options tells wget to stop downloading when the `<quota>` (in megabytes) has been met. Also, these options also ignores HTTP headers with fake lengths.

Name: grep
Description: Prints lines from stdin or ¡file¿ to stdout matching a pattern.
Example usage: `cat laborationreport.txt | grep -i spellingerror`
Example descrption: Scan laborationreport.txt and print all lines containing instances of spellingerror (ignoring case) to stdout.

Name: sort
Description: Sorts lines of collected input and prints it to stdout.
Example usage: `cat <list> | sort -r`
Example description: Read lines from `<list>` and sort them in reverse alphabetical order via pipe, then print them to stdout with cat.

Name: wc
Description: Count the number of words, newlines and/or byte counts in a file or from standard input.
Example usage: `wc -w laboration3.tex`
Example description: Prints the number of words in this document (the latex source, not the PDF).

2.3 Pipes and redirection

`ls / > lsoutput.txt`
Sent the results of `ls /` via a pipe to lsoutput.txt. No output to standard out, because the command did not contain any errors.

`ls /hh > lsoutput2.txt`
The command exited with an error printed to standard error because the file/directory hh did not exist. However, the lsoutput2.txt file was created, but did not contain any information.

`ls / 2> lserror.txt`
Sent standard error from command `ls /` to lserror.txt, but since the command exited without error lserror.txt was empty.

`ls /hh 2> lserror2.txt`
Sent standard error from command `ls /hh` to lserror2.txt, and since the command exited with an error (there was no such file/directory) the message was printed to lserror2.txt.

Standard input (stdin), output (stdout), and error (stderr) is used to provide a command-line interface between an application and another part, whether it is a user or another application. Stdin is the information (input) given to the application, which then responds with output via the same channel (the command-line interface). The output can be classified as either a success (stdout) or a failure (stderr). All this information can be manipulated by the standard shell tools such as pipe (`|`) and redirection (`>, >>, 2>, 2>&1`).

A References

All the man pages for the commands discussed in this document.

icanhascheezburger.com.