GNU and Linux commands

About my favourite file manager

Linux for Teachers

Nick Urbanik <nicku@nicku.org>

5, 6 December 2005

Outline

Contents

1 Why the Command Line? **2**
1.1 Advantages of the Command Line 2

2 Comparing GNU commands with DOS **2**

3 Isn't it too slow? **3**
3.1 Doing it even faster 4

4 How can I get help with these commands? **4**

5 What's the GNU, anyway? **5**

6 Some Things to Do **6**

7 Being the System Administrator **6**
7.1 Getting more software 6

8 Some Things to Read **7**

9 License Of This Document **7**

1 Why the Command Line?

Why the Command Line?

- Isn't that going back to the dark ages?
- Don't file managers make all that obsolete?
- Doesn't it take so much longer to type all those crazy commands?
- "Aren't you just telling us all this crap to waste our precious time we could spend learning something useful?"
- I'll tell you a secret

My Dirty Secret

- You may be better with a file manager than I am
- you may have had more practice than I have…
- … because my favourite file manager is the command line
  - … keeps umbrella handy to protect against the expected shower of rotten tomatoes
- Still here? Hmm, okay, I'll tell you why.

1.1 Advantages of the Command Line

Advantages of the Command Line

- Allows you to automate things
- Provides quick ways of getting at things similar to what you did before
- quicker than pulling down menus

2 Comparing GNU commands with DOS

Comparing GNU commands with DOS

- You are nearly as old as I am,
- so many of you are familiar with DOS commands.
- If you were just out of school, this comparison would be useless

GNU ↔ DOS

Action	GNU	DOS
list files (short)	`ls`	`dir /s`
list files (long)	`ls -l`	`dir`
copy files	`cp` ⟨*sourcefile*⟩ `...` ⟨*target*⟩	`copy` ⟨*sourcefile*⟩ ⟨*target*⟩
move files	`mv` ⟨*sourcefile*⟩ `...` ⟨*target*⟩	`move` ⟨*sourcefile*⟩ ⟨*target*⟩
rename files	`mv` ⟨*sourcefile*⟩ ⟨*target*⟩	`ren` ⟨*sourcefile*⟩ ⟨*target*⟩

GNU ↔ DOS

Action	GNU	DOS
change directory	`cd` ⟨*dirname*⟩	`cd` ⟨*dirname*⟩
make directory	`mkdir` ⟨*dirname*⟩ `...`	`md` ⟨*dirname*⟩
remove directory	`rmdir` ⟨*dirname*⟩ `...`	`rd` ⟨*dirname*⟩
show current directory	`pwd`	`cd`
show content of text file	`cat` ⟨*file*⟩	`type` ⟨*file*⟩

3 Isn't it too slow?

Doing it fast

- Bash shell provides command-line editing
  - Move: Home, End, ←, →
  - Delete words: Esc D, Esc Backspace
  - Delete to end of line: Control-k
- tab completion
  - type the first few characters of a command, file, ... and press the Tab key: the shell will complete the name
- A history of previous commands
  - press ↑, ↓

- **R**everse search through history
  - type any characters from a previous command and press Control-R repeatedly until you see the command you want

3.1 Doing it even faster

Doing it even faster with loops

- command line editing, tab-completion and history all make the command line fast...
- ... but the real speed comes with automation
- We can automate things with loops
- Here are the last few things I did with loops:
- **list PDF documents I wrote, with the number of pages:**

```
$ for i in *trans.pdf;do o="$(pdfinfo $i|egrep
'^Pages:|^Author:|^Title:')";if echo $o| egrep -q
'Author:  +Nick';then echo $i:;echo "$o";fi;done ↩
```

- **Set timestamp correctly in family photos:**

```
$ for i in 2005_09_11 2005_09_12 2005_09_13 2005_09_14
2005_09_15 2005_09_16 2005_09_17 2005_09_18;do pushd
$i;pwd;exif-timestamp-adjust.pl *.jpg;popd;done ↩
```

- **View all my teaching handouts:**

```
$ for i in *slides-beamer-handout.pdf;do xpdf $i&done ↩
```

- **Pretty-print my C++ programs:**

```
$ for i in *.cpp;do pretty-print-cpp $i| lpr;done ↩
```

- **Make a hundreds table for my son:**

```
$ for ((i=1;i<=100;++i));do echo -n "$i "; if ((i %
10 == 0));then echo '';else echo -n '& ';fi;done >
hundreds-table.tex ↩
```

4 How can I get help with these commands?

How can I get help with these commands?

- A few commands are built into the `bash` shell; you can get help for these by typing `$ help` ↩

- The other commands are in `/bin`, `/usr/bin`
- you can go there and have a look
- each one has a "*man page*"
- To read the page for `ls`, you can do `$ man ls` ↩
- To search for ⟨*word*⟩ in the `man` page, type /⟨*word*⟩
- To quit, type: q

But where is the woman command? Isn't this sexist?

- "`man`" is short for "manual"

5 What's the GNU, anyway?

What's the GNU, anyway?

- GNU stands for "GNU's Not Unix"
- It's the project started by Richard Stallman
- Aims to provide all we need without contaminating our computers with non-free software
- Enables membership of the Church of Emacs.

6 Some Things to Do

Some things to do

- Write a `for` loop to count up to 100: `$ for ((i = 0; i <= 100; ++i));do echo $i;done` ↩
- See all the `man` pages for programs in `/bin`: `$ cd /bin; for i in *;do man $i;done` ↩
- Do the same thing for the commands in `/usr/bin`
- Note: pressing Control-C will interrupt a loop that gets stuck going "forever"

7 Being the System Administrator

Being the system administrator

- You write to anything under your home directory, or in `/tmp` with your own account
- . . . but to write anywhere else, you need to be the system administrator and have `root` access.
- I highly recommend that you use `sudo` to become the system adminstrator.
- To do that you can follow the separate guide available at `http://nicku.org/lpic102/lpic/general-linux-2/lab/sudo/sudo.pdf`

7.1 Getting more software

Getting more software

- You can use `yum` to install new software
- You can become `root` by either:
  - using `sudo` ✔
  - . . . or by typing `$ su -` ↩ and then entering the `root` password ✘
- Then do: `$ sudo yum -y update` ↩ or `# yum -y update` ↩
- To install Blender, do: `$ sudo yum -y install blender` ↩
- To install Scribus, do: `$ sudo yum -y install scribus` ↩

8 Some Things to Read

References

[1] Dr. Peter Salus. *The Daemon, the GNU & the Penguin*. Grocklaw 2005. `http://www.groklaw.net/staticpages/index.php?page=20051013231901859`.

[2] GNU General Public License. `http://www.gnu.org/copyleft/gpl.html`

9 License Of This Document