

Mathematics for Real-World Systems Centre for Doctoral Training
Introduction to Computing, 2019

Introduction to Bash & Ubuntu

The Ubuntu terminal application provides a bash shell command line interface which provides powerful computing tools, but can be challenging for new users to adopt. These slides include notes and exercises which provide some of the very basics in command line usage, and aim to help new users begin to adapt to this computing approach.

1: File Manipulation
2: Package Management

Chris Norman

Introduction

Bash (Unix shell)

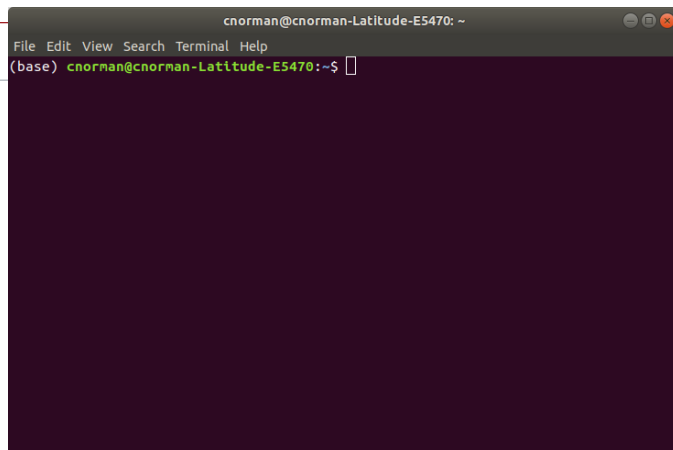
From Wikipedia, the free encyclopedia

GNU Bash or simply **Bash** is a [Unix shell](#) and [command language](#) written by [Brian Fox](#) for the [GNU Project](#) as a [free software](#) replacement for the [Bourne shell](#).^{[7][8]} First released in 1989,^[9] it has been used widely as the default [login shell](#) for most [Linux](#) distributions and [Apple's macOS Mojave](#) and earlier versions. A version is also available for [Windows 10](#).^[10] It is also the default user shell in [Solaris 11](#).^[11]

Bash is a [command processor](#) that typically runs in a [text window](#) where the user types commands that cause actions. Bash can also read and execute commands from a file, called a [shell script](#). Like all Unix shells, it supports filename [globbing](#) (wildcard matching), [piping](#), [here documents](#), [command substitution](#), [variables](#), and [control structures](#) for [condition-testing](#) and [iteration](#). The [keywords](#), [syntax](#), [dynamically scoped](#) variables and other basic features of the [language](#) are all copied from [sh](#). Other features, e.g., [history](#), are copied from [csh](#) and [ksh](#). Bash is a [POSIX-compliant](#) shell, but with a number of extensions.

The shell's name is an acronym for *Bourne-again shell*, a [pun](#) on the name of the [Bourne shell](#) that it replaces.^[12] and on the common term "[born again](#)".^{[13][14]}

A [security hole](#) in Bash dating from version 1.03 (August 1989),^[15] dubbed [Shellshock](#), was discovered in early September 2014 and quickly led to a range of [attacks](#) across the [Internet](#).^{[16][17][18]} Patches to fix the bugs were made available soon after the bugs were identified.



- The aim of these exercises is for you to gain familiarity with using command line inputs, as oppose to the graphical interfaces many are more comfortable with.
- The command line allows for fast and flexible processing, and is a powerful (often essential) tool for scientific computing.
- As would be done in practice, these exercises should be attempted using any & all available resources (I.e. cheat sheets and the internet).

File Manipulation

- 1) Begin by opening a new terminal window using **Ctrl + Alt + T**
- 2) Navigate to your *Documents* directory using **cd**
- 3) List the contents of your *Documents* directory using **ls**

Tasks should be completed only using the terminal but it may be helpful to visualise the process through a file browser window!

- 4) Create a new directory in *Documents* called *Terminal_Exercises* using **mkdir**
- 5) Create two sub-directories within it called *Transfer* and *Exhibit*

*Try to get used to navigating up and down your filesystem using **cd**!*

File Manipulation

- 6) Create a text file in the *Transfer* directory called *SentientAI.txt*, containing the text “Hello World” using **nano**

nano is a command-line program build into Ubuntu, it is not a powerful text editor (c.f. atom), but it is quick for small file changes!

- 7) Copy *SentientAI.txt* from *Transfer* to *Exhibit* using **cp**

- 8) Re-name the *SentientAI.txt* file in *Exhibit* to *FalseAlarm.txt* using **mv**

Re-naming a file is equivalent to moving its contents from one path to another!

- 9) Delete the directory *Transfer* and all of its contents using **rm**

You can perform file operations from any directory! Try to repeat some of the above exercises from your home directory taking care to fully specify the correct file path!

Package Management

- Use **apt update** to 'update' the list of available upgrades for installed packages
- Use **apt upgrade** to actually fetch and install the new versions of installed packages

SUDO – “Run as superuser”

Some commands, like **apt update**, can only be performed by the device 'superuser' (root). In order to run a command as *superuser*, it must be preceded by **sudo**, which will prompt a request for the *superuser*'s password.
You are the *superuser* of your personal device.

The **apt** command utilises Ubuntu's *Advanced Packaging Tool*, and is a convenient way to manage many software packages in Ubuntu via command line. Alternatives you may come across include Ubuntu's *Software Centre* (a GUI); **pip** for Python packages; and *.deb* (Debian) files.

Package Management

- Attempt to run the software **sl** by typing it as a command in the terminal
- Install **sl** using **sudo apt install sl** and then attempt to run it again

*sl accepts options in the same way as other commands
e.g. **sl -a**, **sl -l**, and **sl -F***

- Uninstall **sl** using **sudo apt remove sl**

Some commands (e.g. **matlab**) may not, by default, be executable from any directory. When a command is typed in the terminal, your machine searches for the corresponding executable in a number of directories specified by your *PATH*.

- Display the contents of your *PATH* using **echo \$PATH**

Executables can be added to the *PATH*, most commonly via *symbolic link*: **ln -s file link**