# An Introduction to Unix (concise version)

by Bernard Doyle

ver 4.2 (Web Systems) March 2015 Bernard.Doyle@uts.edu.au

## Outline

- This document is intended to provide an introduction to some basic features and commands of unix/linux.
- Unless you are already proficient with the basics of unix, you are strongly advised to work through this document and do the exercises in it.
- The features and commands described are assumed knowledge in linuxgym and are not described in linuxgym.

# The Unix Command Line Interface (CLI)

In this workshop we are going to look at the command line interface that unix operating systems provide to users.

# Unix Users and Accounts

- People who use a unix system are known as as “users”
- Users have usernames and passwords. The combination of a username and a password is an account. For example username:”fred” and password “HIcr0N!”
- Passwords should be cryptic but memorable.One way to create a password is to use the first letter of a phrase e.g. Here I come, ready or not!

## Unix users and accounts the root user

- There is one special user “root” who can do anything on a unix machine
- You will never normally be root on the university’s machines
- You can be root on your own computer running unix at home.

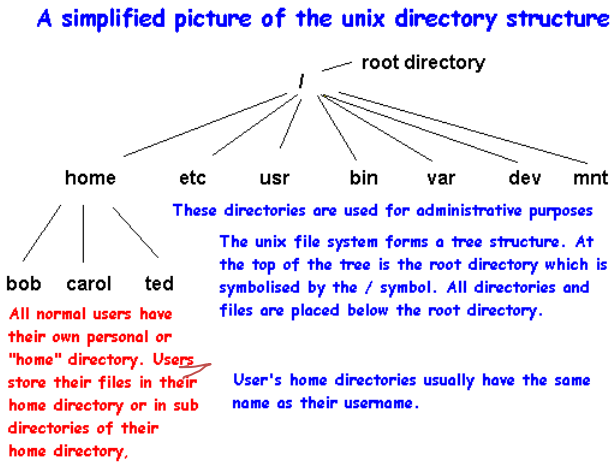
## Practical Exercise 1

- Log on to the machine in front of you using the supplied user name and password.

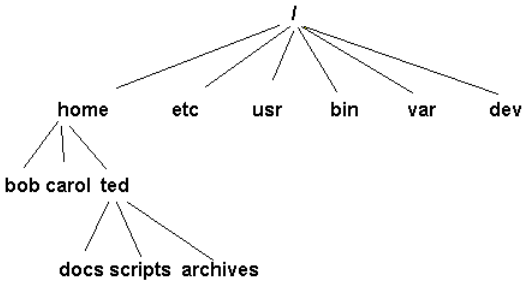
# Unix Directory Structure and Paths

- To specify the location of a file in windows we use terminology like:
  C:\Program Files\Office\word.exe
- In Unix we specify the location of a file in the following manner /home/fred/documents/job_applications/cv.txt
- Note that windows uses a backslash while unix uses a forward slash to separate directory and file names.
- The full specification of a file's location in unix is known as a pathname.

# Unix Directory Structure (1)



# Unix Directory Structure (2)




The diagram above shows a small portion of the directory structure for a unix computer. Directories are the unix equivalent of windows folders. In the diagram above the home directories of the three users bob, carol and ted are shown. Ted's home directory has 3 subdirectories - docs, scripts and archives.

# Home Directories

- Every ordinary user (like you) has a home directory.
- The home directory and its subdirectories of the home directory is where users store their personal files.
- The home directory has the same name as the user. In most unix systems (including the ones at UTS) the user's home directory is a subdirectory of the /home directory

# Specifying your home directory

A user's home directory can be specified in at least two ways:

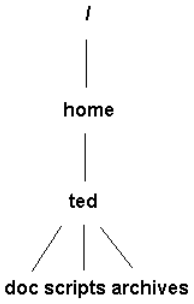
(1) By the absolute path to the directory

e.g. /home/fred

(2) By the using the tilde ( ~ ) character
(pron. "tild-duh"

i.e. ~ symbolises a users home directory

# Unix directories, subdirectories and parent directories




The diagram at the left shows the file and directory structure for the user ted. The diagram also illustrates the fact that every directory (except for the root directory) is a subdirectory of another directory. Similarly, all files are contained in directories.

The term "parent" directory is used to describe the directory immediately above a file or directory.

A directory may have zero, one or many subdirectories and files below it. However, a directory or file can have only ONE directory above it. The directory immediately above a file or directory is known as the "parent directory" and it uses the .. symbol.

## Using Directories

- Unix Directories are like Windows folders
- Directories can contain files and/or other directories.
- If you have an account on a computer running unix, you will almost certainly create subdirectories of your home directory to keep your files organised.
- The command for creating a directory is mkdir <directory name>

## Interacting with Unix

- Users can interact with unix either through a Graphical User Interface (GUI) or through a CLI using a terminal or terminal emulator.
- Terminals interact with unix in text mode only. Input is through the keyboard only, output is in text through a 25x80 character display. (Sometimes the 25x80 can be increased).
- Terminal emulators (such as putty, terminal, konsole and xterm) run in a GUI but emulate a standard terminal.
- Almost everything we do in this practical will be through a terminal emulator program

# Unix Commands options and arguments

## Unix commands have a structure

command argument1 argument2 argument3 ...

There are two types of arguments:

(1) Command line options

(2) Ordinary arguments

# Unix Commands options and arguments

Options begin with a - sign

e.g. `uname -a` — Command line option

Ordinary arguments don't begin with a - sign

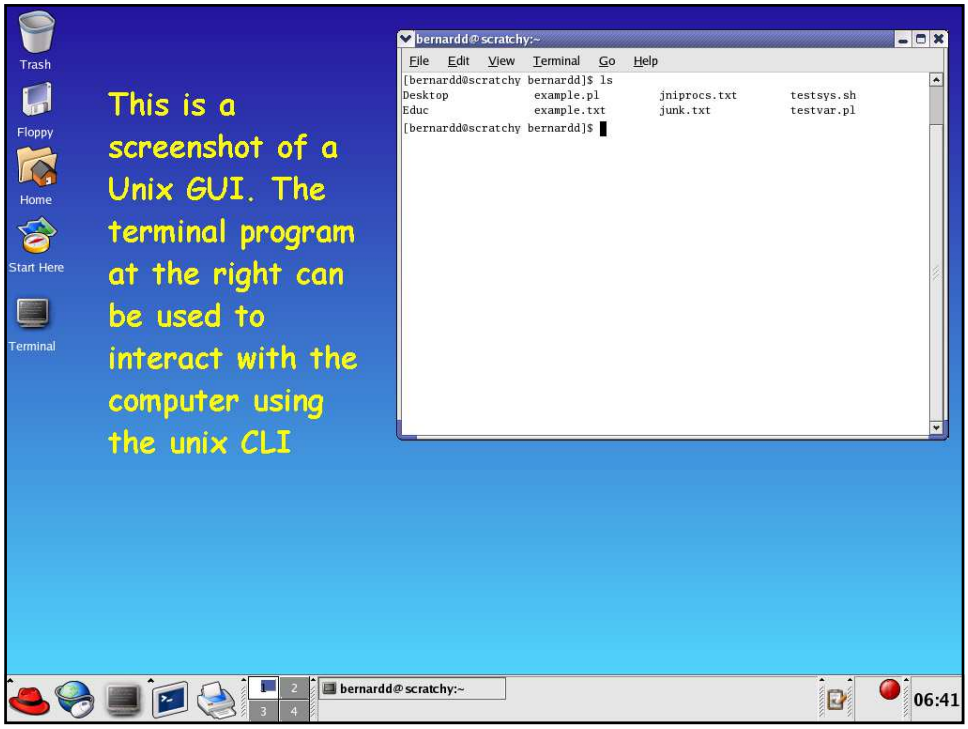
e.g. `ls *txt` — Ordinary command line argument

Options and ordinary arguments can both be used with the one command. Options always precede ordinary arguments.

e.g. `ls -a *pdf`

Depending on the command, more than one option and argument may be used. Options can sometimes be combined as well.

e.g. `ls -a -l *doc`
`ls -al *doc` — These two commands are the same




# Getting Oriented in Unix (1)

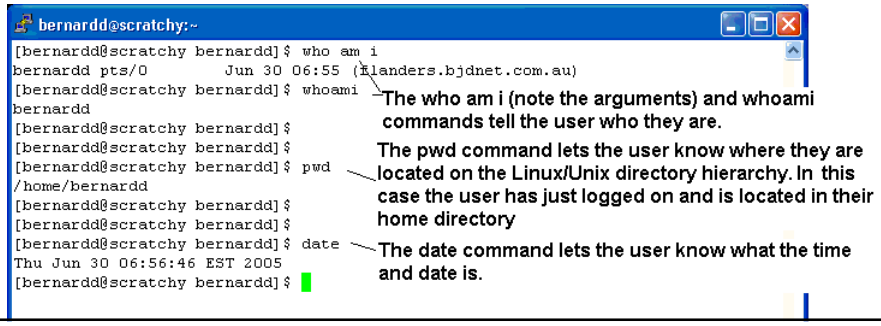
When we are logged onto a computer running unix we are always located somewhere in the unix directory tree. Usually when we first log in, we are located in our home directory. Our current location is our present working directory

Five commands are useful for orienting ourselves in a unix system. These are pwd, whoami, who, uname and date.

| Action | Unix Command |
|---|---|
| Where are we ? i.e. what's our present working directory ? | `pwd` |
| Who are we ? | `whoami` |
| Who else is logged on ? | `who` |
| What variety of unix are we using? | `uname` |
| What is the date and time ? | `date` |

# Getting Oriented in Unix (2) an example session

The picture below shows the effect of running the whoami, pwd and date commands in a terminal session on a unix computer.



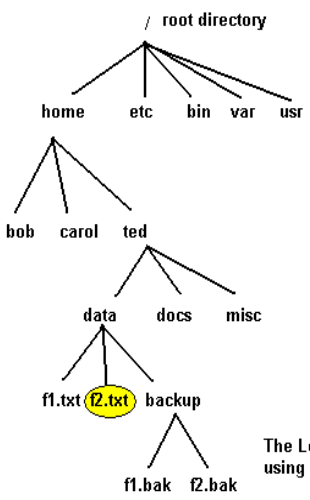

# Practical Exercise 2

- Use the commands pwd, date, uname, whoami, and who to orient yourself and find out who else is on the computer you are logged into.

# Some more unix commands

| Command | What it does |
|---|---|
| `cd` | changes the working/current directory |
| `mkdir` | creates a directory |
| `ls` | lists the contents of a directory |
| `cp` | copies a file. |
| `rm` | removes a file (be careful using this) |
| `mv` | renames a file (equivalent to copy a file and delete the original) |
| `rmdir` | removes an empty directory (be careful using this) |

# Navigating Directories – Absolute and Relative Paths




In this diagram of part of the file and directory structure of a computer running unix, we see that the user ted has 3 subdirectories. In the subdirectory data, he has 2 files and another subdirectory called backup.

The location of files and directories can be specified using absolute and relative pathnames. Absolute pathnames specify the location of a file relative to the root directory.

The absolute location of f2.txt is /home/ted/data/f2.txt

From the misc directory the relative location of f2.txt is ../data/f2.txt

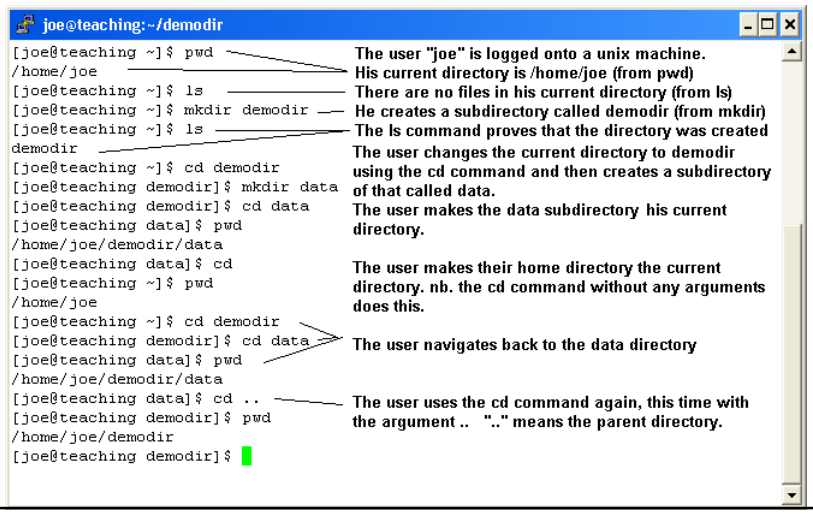
The .. notation means the parent directory of the working directory.

The Location of a file or directory can be specified using an absolute or relative path.

# Relative and Absolute Paths

- An absolute path starts at the root directory and navigates through the directory structure to a particular file or directory. An absolute path always leads to the same location.
- A relative path starts from where the user is currently located i.e. their present working directory. A relative path can lead to different locations depending on where the user is located.

# Creating directories and navigating through them



# Copying Files

**Files are copied in unix using the cp command**

**example :**

**To make a copy of the file f1.txt called f1.bak the command is :**

```
cp f1.txt f1.bak
```

**The syntax of cp is**

```
cp original_file_name destination_file_name
```

# Getting Help
# the man command

There is an online help facility in all unix installations. This is accessed using the man command and the data the command provides are known as the “man pages”

As an example, to access the man pages for the ls command type in `man ls`

Other sources of help:

A brief summary of a command can often be obtained by typing in the name of the command with the argument `--help`

Some unix systems (such as linux) have an extra help system known as “info”. This can be accessed by typing in info followed by the name of the command.

# Some Tips

- To clear the screen, press the control and L keys simultaneously.
- Previously executed commands can be accessed by pressing the up arrow key on the keyboard.
- Tab completion enables users to key in long and confusing filenames with a minimum of keystrokes.

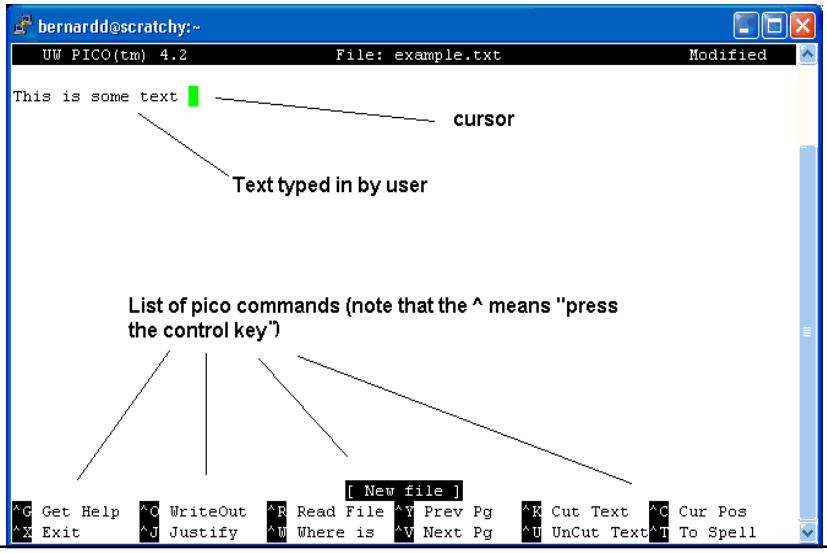
# Editors

Editors are an important tool that enable users to create and alter text files. Unix systems often have a number of editors:

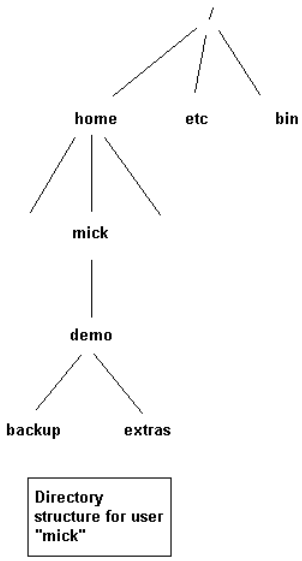
- nano (sometimes called pico) – easy to use editor. (The same editor with 2 different names).
- vi – powerful and installed on most unix systems. Learning vi is compulsory in Web Systems.
- emacs – powerful and usually installed on most unix systems, reasonably steep learning curve.

Each of these editors can be started by typing in the name of the editor followed by the name of the file to be edited.

# The nano/pico interface




# Practical Exercise 3




Directory structure for user "mick"

(1) Create a subdirectory called "demo" in your home directory. Make demo your current directory (i.e. your working directory) and create subdirectories of demo called backup and extras.

(2) Using pico, create a a file called f1.txt in the demo directory.

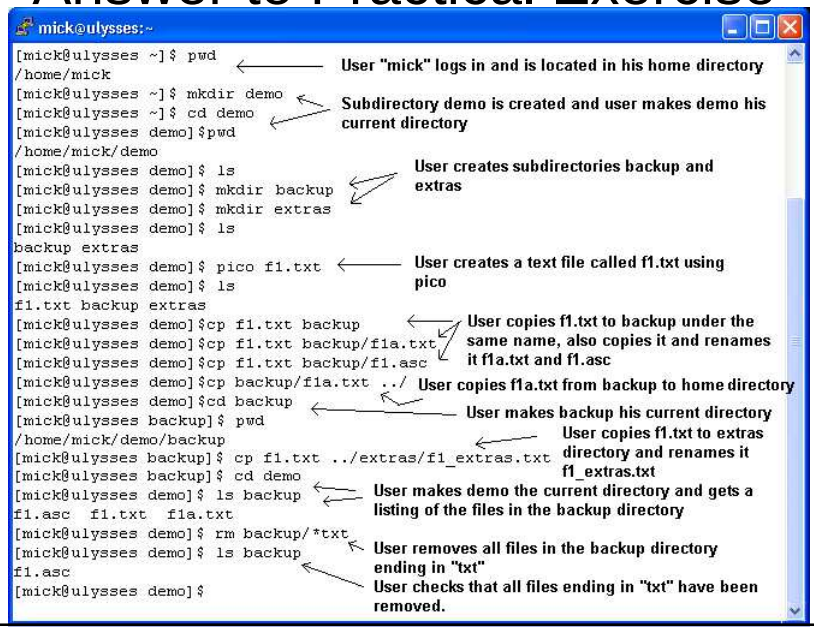
(3) Copy f1.txt to the subdirectory backup under the same name. Then copy f1.txt to the subdirectory backup but this time name the copy f1a.txt Copy f1.txt again to backup this time naming the copy in backup f1.asc

(4) While demo is still your current directory, copy f1a.txt from backup to your home directory.

(5) Make backup your current directory and copy f1.txt from backup to extras, renaming the copy in extras f1_extras.txt

(6) Make demo your current directory. Delete all files in backup that end in "txt"

# Answer to Practical Exercise




# Summary of Unix Commands Learnt

| | |
|---|---|
| whoami | rmdir |
| uname | cp |
| date | mv |
| pwd | rm |
| ls | nano |
| cd | vi/vim |
| mkdir | man |

# Summary of Concepts

| | |
|---|---|
| Command Line interface | Graphical user interface |
| file | directory |
| Unix file system hierarchy | Present working directory |
| Home directory | Command options |
| Command arguments and options | Owner, group, rest of world |
| Pathnames | Absolute path |
| Relative path | Permissions – read, write and execute |

# Useful practical tips

| | |
|---|---|
| Command Line tab completion | Command line history |
| command --help<br>gives a quick and dirty summary of command options (usually) | Cntrl-l to clear a terminal screen |
| Cntrl-c to kill a command that has taken over the screen | Cntrl-d to log off |

# Where to from here ?

There are a number of things you can do:

- Get your own unix system (Linux, FreeBSD, NetBSD, OpenBSD, Open Solaris) *
- Buy a Macintosh – the operating system is a variety of Unix
- To get your own version of unix, download linux/freebsd iso files from the internet. A good place to start is http://distrowatch.com
- To install unix/linux the easiest option is to use virtualbox or VMWare. These can also be downloaded from the net. A more difficult approach is to burn the iso files to a DVD and then installing onto a physical computer (partitioning may be necessary).
- Use the computers at UTS running unix systems (Solaris and Linux)