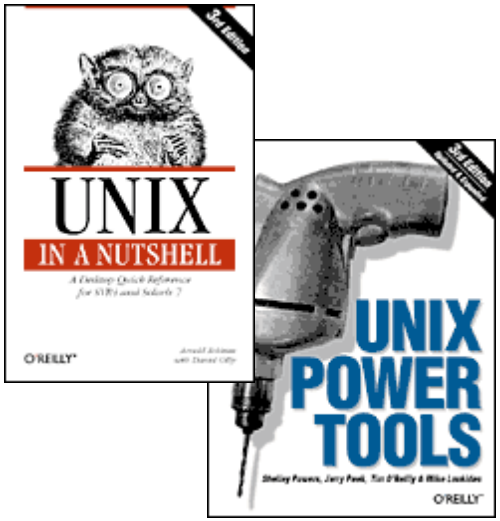
Introduction to Unix



http://linux.oreilly.com/

Unix tutorial

Outline

- Basic background in Unix structure
- Directories and files I (listing, navigation, filenames)
- Directories and files II (working with files & their content)
- Redirection (output, input, pipes)
- Editors
- Access rights, processes
- Other useful Unix commands

Unix - basic background (oversimplified)

UNIX is a common operating system (suite of programs which make the computer work) made up of three parts; the kernel, the shell and the programs.

The kernel

The kernel of UNIX is the hub of the operating system: it allocates time and memory to programs and handles the filestore and communications in response to system calls.

The shell

The shell acts as an interface between the user and the kernel.
When a user logs in, the login program checks the username and password, and then starts another program called the shell. **The shell is a command line interpreter (CLI)**.

The commands are themselves programs: when they terminate, the shell gives the user another prompt (~> on our systems).

Filename Completion

By typing part of the name of a command, filename or directory and pressing the **[Tab]** key, the shell will complete the rest of the name automatically.
If the shell finds more than one name beginning with those letters you have typed, it will beep, prompting you to type a few more letters before pressing the tab key again.

History

The shell keeps a list of the commands you have typed in. If you need to repeat a command, use the cursor keys (**arrows**) to scroll up and down the list or type "**history**" for a list of previous commands.

Unix - basic background (oversimplified)

Files and processes

"Everything in UNIX is either a file or a process."
A **process** is an executing program identified by a unique PID (process identifier).

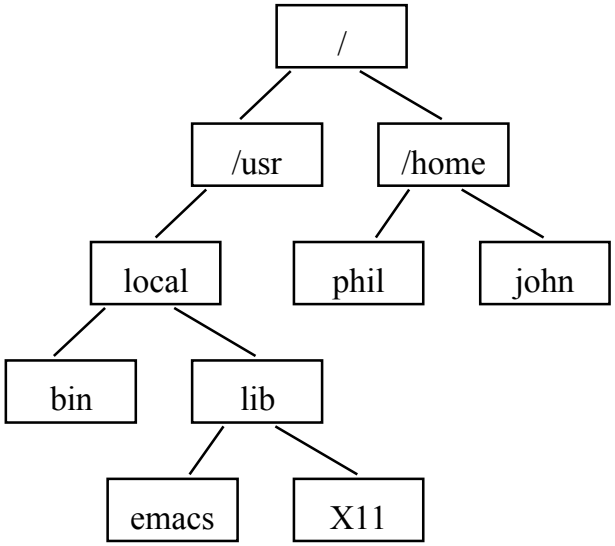
A **file** is a collection of data. They are created by users using text editors, running compilers etc. It is identified by a name and an logical address (or path).

Examples of files:

- a document (report, essay etc.)
- the text of a program written in some high-level programming language
- instructions comprehensible directly to the machine and incomprehensible to a casual user, for example, a collection of binary digits (an executable or binary file)
- etc…

Unix - basic background (oversimplified)

The directory structure



Starting an terminal session (CLI)

1)  

2) An terminal window will appear with a Unix prompt, waiting for you to start entering commands.

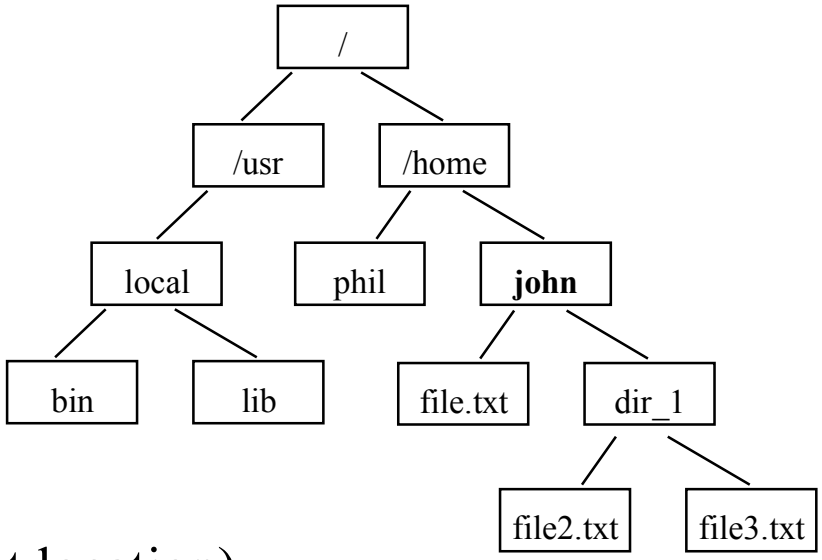
Unix - Directories and files I

ls lists files in a directory

ls -la lists all (**a**) files in long (**l**) format

mkdir creates a new subdirectory

rmdir removes an **EMPTY** directory



pwd print working directory (path, current location)
(/home/john)

cd changes current directory
Note: typing **cd** with no argument or typing **cd ~** always returns you to your home directory.

. in UNIX means the current directory
.. in UNIX means the parent of the current directory

Unix - Directories and files II

cp copies files

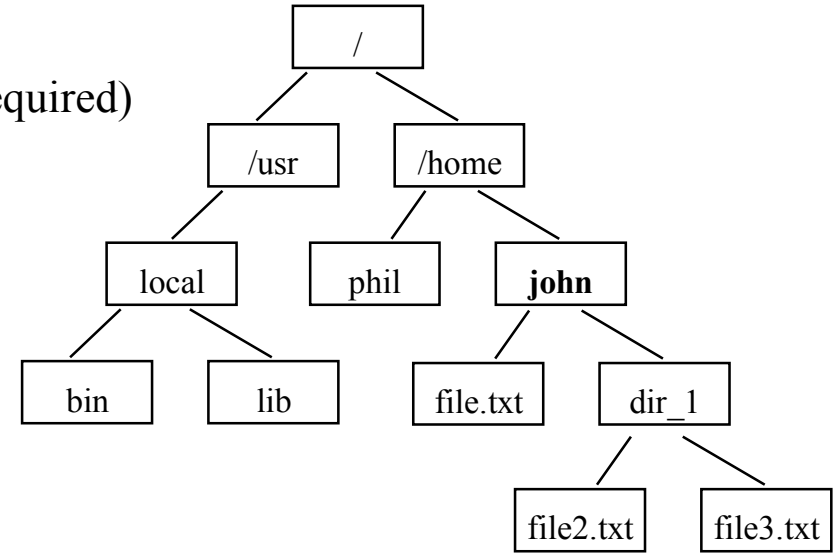
copy orig.file copy.file (pathname may be required)

mv moves/renames a file

mv file1 file2

rm removes/deletes a file

rm -i file2 (-i : interactive)



clear clears all text

find searches the directory tree for a file

find . -name lostfile -print

***** matches any number of characters in a file (or directory) name

? matches only one character.

Unix - Directories and files II

cat concatenates/displays **all** the content of a file on the screen

less displays the content of a file **one screenful at a time**
less file.txt
then press spacebar to display the next screen or use arrow keys to move up or down row by row.
(Type / followed by a word to find it in the file. Press Q to quit).

more (see less)

head prints the first lines of a file on the screen
(10 by default or specify the desired number of lines with "**-**")
head -20 file.txt diplays the first 20 lines from the file "file.txt"

tail prints the last lines of a file on the screen
(10 by default or specify the desired number of lines with "**-**")
tail -20 file.txt diplays the last 20 lines from the file "file.txt"

Unix - Directories and files II

diff compares two files and prints how they differ

diff file1 file2

wc displays a count of lines, words and characters in a file (word count)

grep searches a file for a string and displays all the lines containing this string

grep word file.txt displays all lines containing "word"

grep "following words" file.txt displays all lines containing "following words"

grep -c word file.txt displays the number of lines containing the string

grep -e "one" -e "two" -e "three" file.txt displays all lines containing "one" or "two" or "three"

grep -v "word" displays all lines that do not contain "word"

Unix - Directories and files II

od The od command copies sequentially each input file to standard output and transforms the input data according to the specified options.

od -c file.txt displays some non-graphic characters as C-language escapes (see more options: man od)

null	\0
backspace	\b
form-feed	\f
new-line	\n
return	\r
tab	\t

Important:
Unix & all Unix flavors: **\n**
Mac OS : **\r**
Windows:**\r\n**

Unix - Redirection (output, input, pipes)

Many processes initiated by UNIX commands:
- write to the standard output (the terminal screen)
- and take their input from the standard input (the keyboard)

The > symbol is used to redirect the output of a command in a file.
If the file exists, its content is replaced, if not, it is created.

cat file1 file2 > file3 redirects the differences into file3

The >> symbol is used to redirect the output of a command in a file.
If the file exists, the output is appended, if not, the file is created.

cat file4 file5 >> file3 appends the differences into file3

The | symbol (pipe) uses the output of the 1st command as the input of the 2nd

grep word file.txt | wc

Unix - Editors

emacs

Emacs is the extensible, customizable, self-documenting real-time display editor.
(http://www.gnu.org/software/emacs/emacs.html)

pico

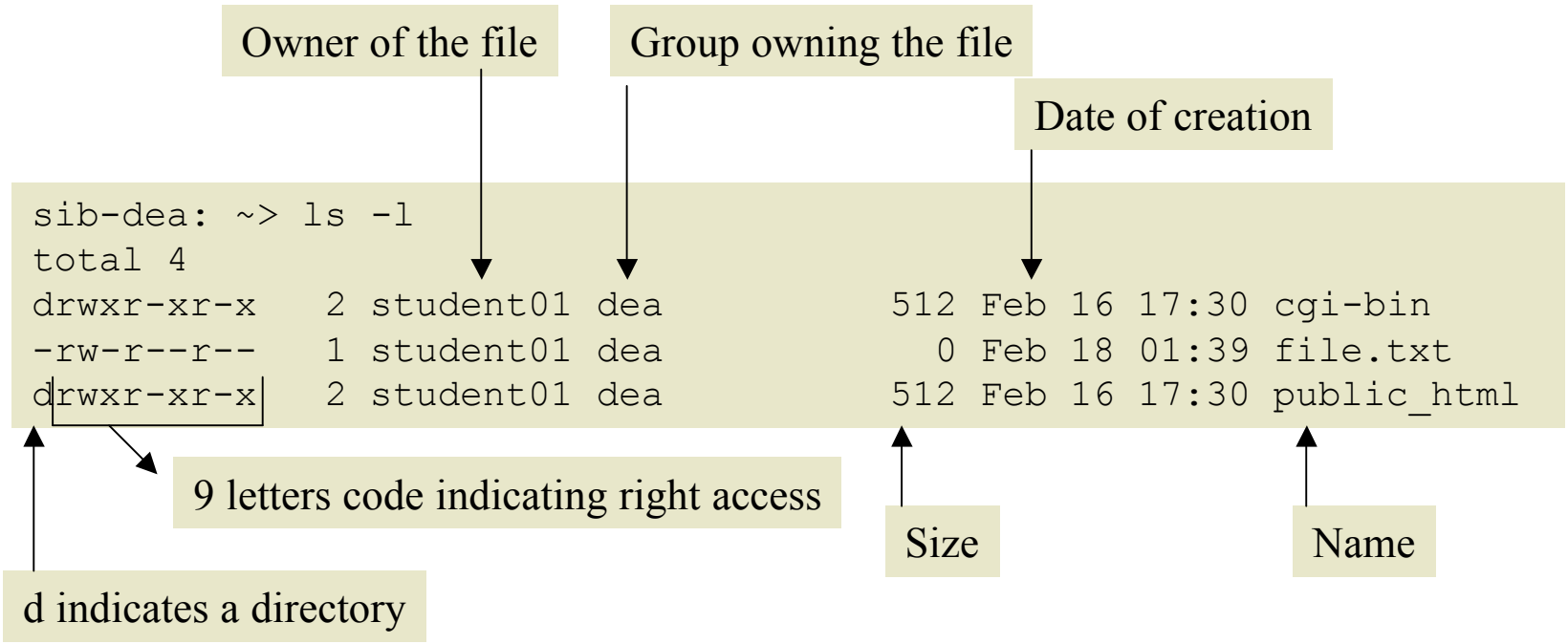
A Unix text editor
(http://www.indiana.edu/~ucspubs/b103/)

vi

The Unix text editor
(http://www.unix-manuals.com/refs/vi-ref/vi-ref.htm)
vim, gvim, etc…

Unix - Access rights, processes

Each file (and directory) has associated access rights, which may be found by typing **ls -l** :



Unix - Access rights, processes

Access rights on files.

• r (or -), indicates the presence (or absence) of permission to read and copy the file

• w (or -), indicates write permission (the permission to change a file)

• x (or -), indicates execution permission (the permission to execute a file)

Access rights on directories.

• r allows users to list files in the directory

• w means that users may delete files from the directory or move files into it

• x means the right to execute files in the directory.

owner			group			others		
r	w	x	r	w	x	r	w	x

Unix - Access rights, processes

Changing access rights on files: chmod

u	USER
g	GROUP
o	OTHERS
r	READ
w	WRITE
x	EXECUTE
-	TAKE AWAY PERMISSION
+	ADD PERMISSION
a	ALL

chmod a+x file.pl

chmod g+w file.pl

owner			group			others		
r	w	x	r	w	x	r	w	x

Unix - Access rights, processes

ps displays information about all processes with their associated PID and status, type

bg moves the current process to the backround (can be done by adding & after the command)

fg moves the current process to the foreground

jobs lists backround and suspended processes

kill stops a process (use ps or jobs to find the process id)
kill %1 (kills the job number 1)

ctrl-C stops the current process

Other useful Unix commands

date displays current date and time

passwd to modify the users's password

tar creates (or extracts) a tarball from (to) a list of files

tar -cvf tarball.tar subdir/*

tar -xvf tarball.tar

(the **-z** option compacts the files by gzip. when -z is used it is advised to use the "gz" extension in addition to tar)

logout logs current user out

exit logs current user out and closes the shell

The most useful Unix command

man The man command (short for "manual")
displays the manual page for the specified command at your terminal.
To page through the output, press the Return key. To exit, type Ctrl-c.

man grep (Lists help information on the "grep" command.)

If you don't know which command you want? Well …
An alternative is to use the "-k" option to tell man to do a keyword lookup for manual pages containing the keyword.

man -k route (This command displays a one-line summary from each manual page matching the keyword "traceroute"
Look through the matches for a likely candidate manual page.)