

Lab Exercise

Operating Systems (Linux)



Getting started.

For Module csn08112

Uses file PuppyWary_5.3.7z

Background

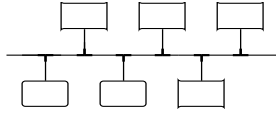
1) **Unix**

Unix is a commercial operating system (OS) originally developed for multi-user systems, it is very flexible, robust and powerful but has the reputation, amongst Windows users, of being difficult to learn. Unix is commonly thought of as being used primarily via a command line interface (CLI) although graphics is supported via the "X" graphics system. Many vendor's have developed their own versions of Unix with graphical user interfaces (GUIs). E.g. MAC OS X is a Unix based graphical operating system.

2) **GNU/Linux**

Born out of the Open Software Foundation, "Linux" is an "open source" OS which is suitable for desktops or servers, home users or businesses. It is very like Unix and anyone is free to modify any part of it so long as this work is also openly available. There are very many different versions of Linux (referred to as "distributions") available for free. The "distro" market is however, dominated by a handful of companies who make money through providing support services.

Like Unix, Linux is robust and powerful but has a reputation for being difficult to setup properly. "X" graphics provides the basic window drawing mechanism whilst the addition of a Window Manager and Desktop Environment creates a familiar GUI. Several desktop environments are available for Linux, some like Gnome and KDE are quite Windows-like, others are "light" versions more suited to small systems. An entire Linux



distribution with a "light" GUI can be fitted onto a small USB pen drive.

Linux Provision

It can be difficult to gain familiarity with Unix or Linux as most general purpose computing facilities use MS Windows exclusively. There are however other options for accessing Linux besides using dedicated Linux desktop machines.

1. Remote Access

a. Command Line Interface.

This can be done from any Internet connected machine via a remote session (e.g. telnet or ssh), however users will be limited to a text only interface and will not gain experience with a Linux GUI.

Note: telnet connections are not secure, all data (including passwords) will be transmitted over the network in plaintext, an ssh client is recommended (e.g. Putty).

b. Graphical User Interface

It is possible to view a graphical screen remotely using Xterm or VNC (Virtual Network Computing) thin-client technology. For example by running a VNC server on a Linux machine a user can connect via the VNC client and see the same desktop as a local user. "X" and VNC clients and servers are available for Linux or Windows platforms.

2. Local Access.

a. Real Hardware

i. Non-removeable Media

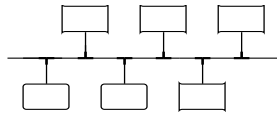
Installing a Linux distribution to machine's hard disk will provide the best performance and most complete range of applications and tools. However not everyone will have a spare machine available as a dedicated Linux desktop. The Linux install procedure accommodates "dual-booting" so that the user can choose whether to boot into Linux or the existing (e.g. MS Windows) operating system when the machine is first powered on.

ii. Removeable Media

Live CD-ROM/ Live USB Linux

These are lean Linux distributions specifically designed to run directly from a CD-ROM or USB pen drive rather than from the hard disk. No installation is required and no changes are made to the user's PC.

There are a large number of live CD-ROM distros available (usually in the form of an .iso file) but both DamnSmallLinux and Puppy Linux are noted for their



small size. Additionally the Puppy Linux live CD-ROM loads its entire OS in memory so that the CD-ROM can be removed after boot up is complete.

This is a very good option for users wishing to try Linux occasionally, however it does require that the machine can be booted from removeable media. This is often not the case in shared labs.

b. Virtual Hardware

Applications exist which create a virtual environment allowing software to be used as if it were running on completely separate hardware.

This effectively creates “virtual machines” (VMs) as separate entities using the facilities of the “host” system.

These “guest” virtual machines can be used to execute any software that could run on the host, independently of the host’s own operating system. VMs are useful for testing software, networking or other operating systems without interfering with the host machine’s own configuration.

Microsoft Virtual PC, Oracle VirtualBox, VMware and Xen from Citrix Systems are a few examples of virtual machine platforms.

i. Linux on a VM

By running a VM platform on a non-Linux (e.g. Windows) host, the user can load a Linux OS VM image and effectively use Linux as if it were on a separate machine.

The user can switch between the host’s OS (e.g. Windows) and guest OS (Linux) just by selecting different desktop windows. The guest VM OS can be given access to the hosts storage and interfaces and a Linux GUI can also be used (see Figure 1).

ii. linuxzoo.net

This is a system which provides remote users with access to Linux VMs. Users can administer their own VMs as if they were real hardware. Each VM can be managed from any web browser and uses a remote command line interface usually via a telnet or ssh client (e.g. PuTTY).

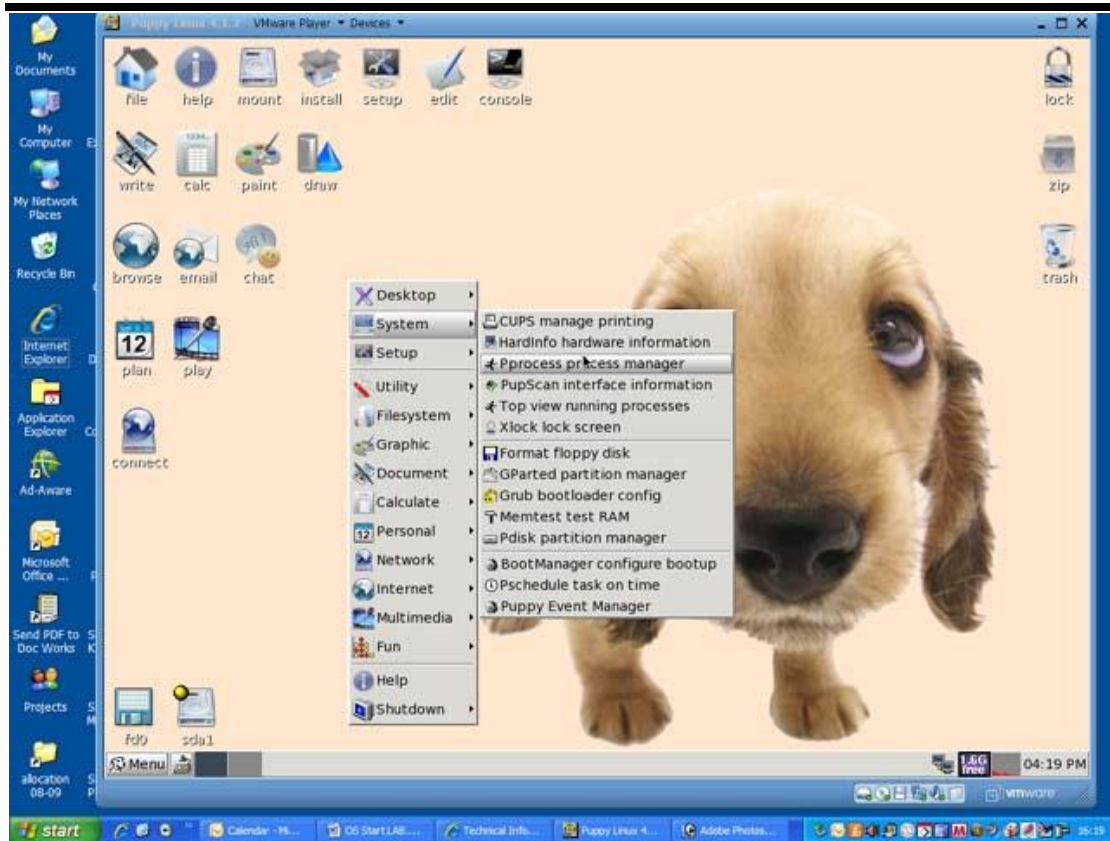
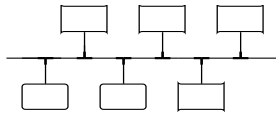
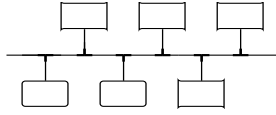


Figure 1 - Puppy Linux in Windows VMware



Exercise 1 – Linux on a VM

Using PuppyLinux with VMware.

To complete this exercise you will need:-

- A Windows PC with VMware Player¹ installed
- A copy of the Puppy Linux VM distro. (PuppyWary_5.3.7z)
- A USB pen drive with at least 1GB of space.

You are going to obtain a complete Linux OS installation to store on your USB pen drive. Using VMware you will be able to run this distribution (including the Linux GUI) within Windows on the host PC.

Any changes you make to your Linux files or system configuration will be saved on your pen drive OS image.

Copy the Linux OS image

1. Find the location of the PuppyWary_5.3.7z file provided for you.
2. Extract the contents of the zip file to a folder on your USB pen drive if you have no pen drive save it to your personal hard disk area.

Run VMware Player

Start the VMware Player application:-

start ► All Programs ► VMware ► VMware Player

Click on **Open** and navigate to the Puppy Linux folder on your USB pen drive.

Select the ".vmx" file then click **Open**.

The Puppy Linux image should boot up in a VM window and start the GUI.

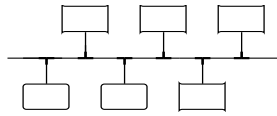
Configure your Virtual Machine

In order to use this Linux image it may need to be configured according to the user's preference in the same way as an OS running on real hardware.

To control the guest VM's cursor, either press **<Ctrl> + G** or just click in the VM window.

To go back the host machine press **<Ctrl> + <Alt>**

¹ VMware Player is a free download available from <http://www.vmware.com/>



Navigating the File System (GUI)

There are many different graphical file system managers available for Linux systems. The Puppy distro features “ROX” which is simple to use and supports “drag and drop”.

Start ROX

Click on the “Home” icon (labelled “**file**”), top right of the screen. This will start ROX and open the home directory of the current user “root” (/root).

Note: As a short hand, the current user’s home directory can be referred to as “~” (pronounced tilde).

Navigate into “my-documents” by clicking once on the directory icon for my-documents.

Notice that the window title bar changes to “~/my-documents”.

Create a new directory in my-documents.

Right click within the directory window.

Select

New ► Directory

The new directory name should be:-

/root/my-documents/**mydir01**

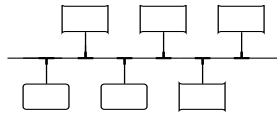
Click [create]

Navigate into mydir01 as above and create a new blank file (**myfile01**) in the same way.

Click on the myfile01 icon to open it using the default editor (Geany).

Enter some text into the file, save the file and close Geany.

Try creating some more directories inside mydir01 and create some files inside them too.



Navigating the File System (CLI)

Unlike graphical menu-orientated interfaces, using the command line efficiently means remembering the various command names.

Linux commands mimic the original Unix names and are brief, often just 2 or 3 letter mnemonics.

Fortunately it is possible to navigate the file system and begin basic file manipulation with knowledge of just a few of these commands.

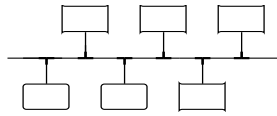
Basic Linux commands

Command	pwd	short for:	print working directory
Description	Shows the full pathname of the directory that the user is currently working in.		
Example	# pwd /home/user01		
GUI equivalent	an "open folder" window or icon		

Command	ls	short for:	list
Description	Shows the contents (file names) in the working directory.		
Example	# ls clipart temp mfile		
GUI equivalent	the contents of an "open folder" window.		

Command	cat	short for:	catalogue
Description	Shows the contents of a (text) file.		
Example	# cat myfile This is the first line This is the second line etc.		
GUI equivalent	Double clicking a .txt file icon.		

Command	cd	short for:	Change Directory
Description	Allows the user to move about the file system by changing the present working directory. Note: To leave a directory and go back up to the parent use "dot dot"		
Example	cd docs cd ..		
GUI equivalent	double clicking a folder icon.		



Command	cp	short for:	Copy
Description	Makes a new copy of a file or directory tree.		
Example	<code>cp oldfile newfile</code>		
GUI equivalent	Select – Copy - Paste		

Command	mv	short for:	Move
Description	Changes the location or name of a file or directory		
Example	<code>mv docs documents</code>		
GUI equivalent	Rename or drag-and-drop		

Command	rm	short for:	Remove
Description	Removes (deletes) a file, directories can be removed with <code>rmdir</code> .		
Example	<code>rm myfile</code>		
GUI equivalent	Select + Del		

Start a Command Line Interface

Click on the console icon on the PuppyLinux desktop (right-most icon on the top row).

This opens up a window with a “#” command prompt.

Verify the pathname of the Present Working Directory

```
# pwd
```

It should be “/root”

Navigate into my-documents

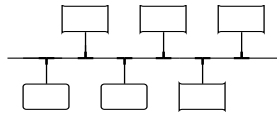
```
# cd my-documents
```

View the directory content

```
# ls
```

Use the “cd” command again to navigate into the “mydir01” directory that you made earlier and view the content.

Now show the content of the file you created using the “cat” command.



```
# cat myfile01
```

Navigate back up to ~/my-documents

Note that the ~ character is used as a shorthand way to refer to a user's own home directory.

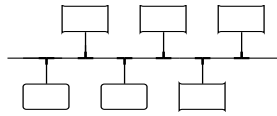
```
# cd ..
```

Use the `ls -l` option to see more detail about files and directories in my-documents.

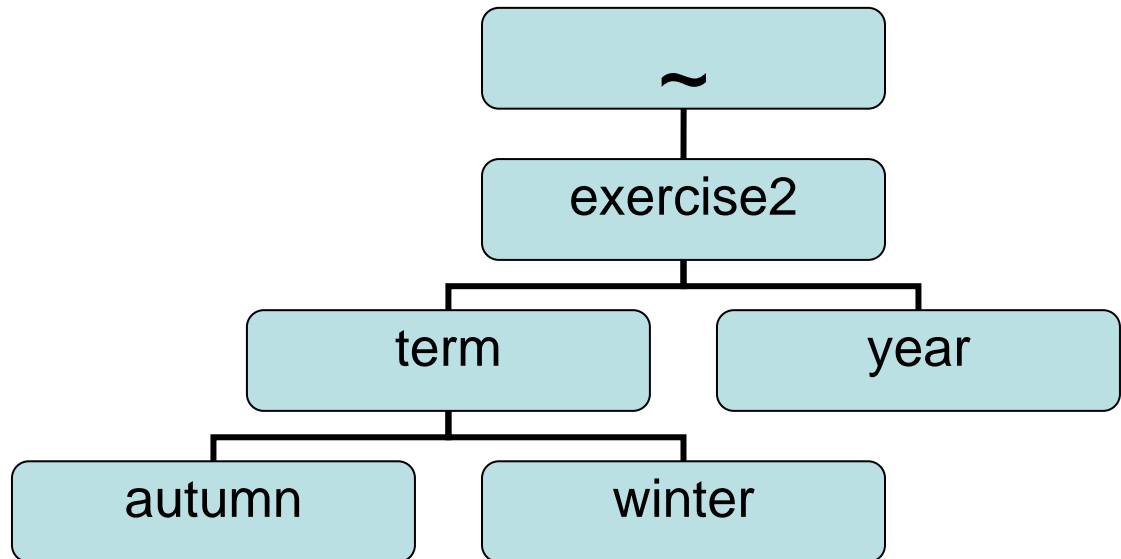
```
# ls -l
```

Return to your "home" directory.

```
# cd ~
```



Exercise 2 – Directories, Pathnames & WildCards



1) Command output redirection.

The output from any command can be saved in a file instead of being displayed on the screen if the command is followed by

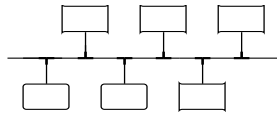
> myfile

where myfile is the name of the file to store the redirected output.

">" will cause a new file to be created or an existing file overwritten.

">>" will cause the output to be **appended** to a file

- a) Click on "console" to open a terminal window.
- b) Using **command line only** create the directory structure shown above inside your home (~) directory.
Use the **mkdir** and **cd** commands; **rmdir** if you make a mistake and wish to remove an empty directory.
- c) Check your directory tree using a graphical file manager tool, e.g. ROX (Puppy Linux)



-
- d) We will use the “cal” command here simply to create some file content.

“cal” displays a calendar of the current month (default), the specified month or it can also give a whole year.

Example: # cal 9 2013

displays a calendar for the 9th month (September) of 2013.

Using the command line, navigate to the winter directory and use redirection to create a file called “sept13” which contains a calendar for September 2013

The command is:-

```
# cal 9 2013 > sept13
```

- e) Check the content of the sept13 file

```
# cat sept13
```

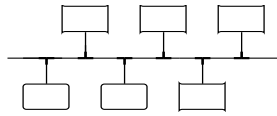
In the same way that you used “cal” to create sept13, create calendar files for **oct13**, **nov13**, **dec13**, **jan14** and **feb14**.

Hint: Use the ↑up cursor key to scroll back through previous commands.

- f) Create a file containing the full 2013 year calendar

```
# cal 2013 > year13
```

Check the seven files exist, use the file manager if you wish.



2) Pathnames

Absolute pathnames are references to files or directories in the file system which begin with either:-

- “/” meaning the filesystem ROOT directory
- “~” meaning the user’s HOME directory

Relative pathnames do not start with either of these characters and their use depends on where in the file system the user is currently working. Relative pathnames start with the name of a sub-directory or with “./” meaning the parent directory.

- a) Without changing out of directory “winter” and without using absolute pathnames (“/” or “~”) move (mv) the file **oct13** into the directory “**autumn**”.

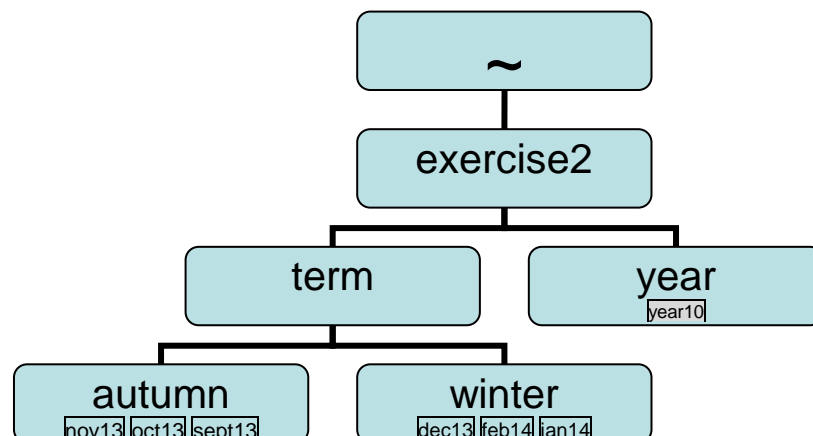
```
# mv oct13 ../autumn
```

Similarly move the file **sept13** into the autumn directory.

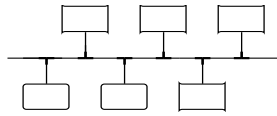
Now move the file **year13** into the **year** directory, be aware that **year** is at a level above your current working directory .

- b) Using **cd** navigate into the **autumn** folder.
Without changing directory, move the **nov13** file from the winter folder into the current folder.
Hint: Use “.” (dot) to indicate present directory.

c)



Check that the seven files created are now in the directories shown above.



3) Appending.

>> is used to redirect the output of a command and append it to the end of an existing file.

- a) Navigate to the **year** directory, now type a command to display the content of the **jan14** file.

```
# cat ../term/winter/jan14
```

Now using “cat” and “>>”, append the content of the jan14 file onto the end of the year13 file.

Check the content of the year13 file, it should end with January 2014.

4) Wildcards

Wildcards can be used to allow multiple files to be manipulated by a single command. This is achieved by using special characters so that expressions will match multiple filenames.

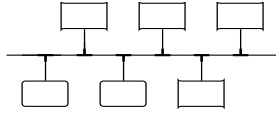
“?” Matches any single character

“*” Matches any group of characters.

- a) Whilst remaining in the **year** directory use the **ls** command with a wildcard to list only files in the winter sub-directory which contain calendars for 2014.
- b) Using the **cat** command with a wildcard, display the content of these files (jan14, feb14).

Repeat this command this time redirecting the output to the file **year14**.

Check that year14 is in the year directory and contains calendars for February and January.



Transferring/Saving Files

All the files that you create within the Linux virtual machine will be saved in the VM's file system (vmdk file). As this can only be read by VMware, it makes it difficult to copy files to a Windows machine later.

It is easy to transfer files to on-line storage if the VM has a working network interface. A virtual machine will normally connect to the network using the host machines real network adaptor.

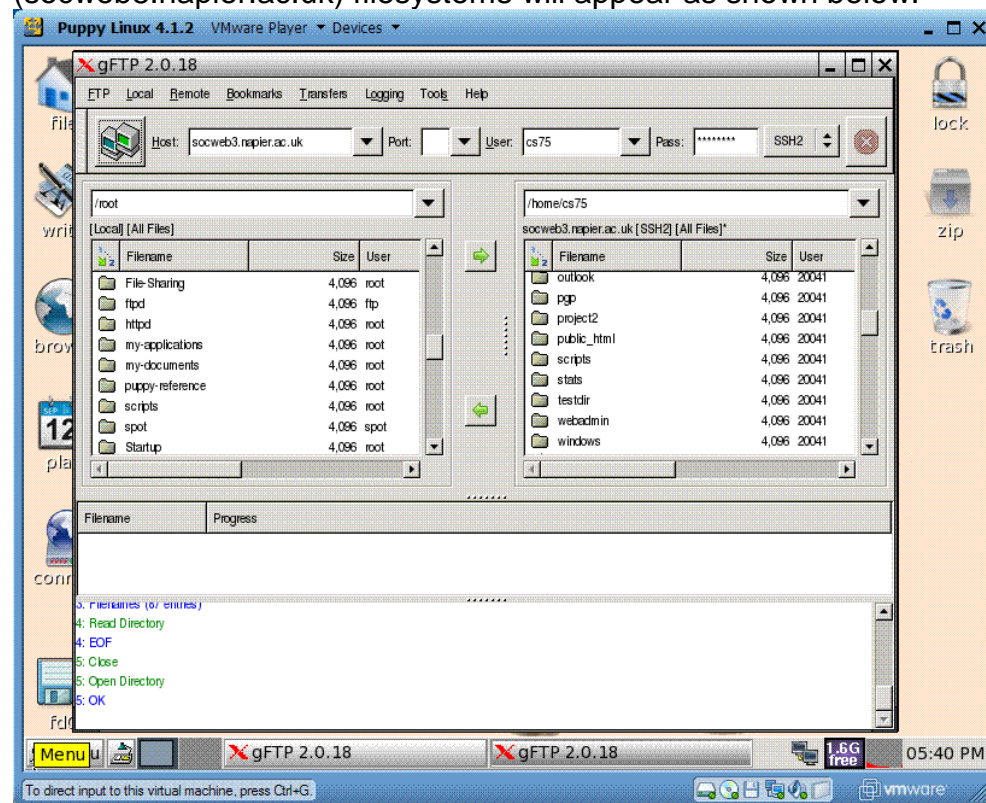
If this is configured, a secure FTP connection can be setup to provide access compatible on-line file storage.

Start gftp

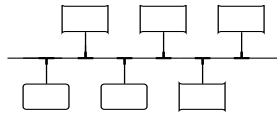
Click on the icon (labelled “**gftp**”), top left of the screen.
This will start gftp pre-configured to connected to the university sftp service.

You will be prompted for a user id and password, use your usual university login credentials.

You should now be connected to the central Linux service.
The contents of your local (PuppyLinux) filesystem and the remote (socweb8.napier.ac.uk) filesystems will appear as shown below.



The left pane shows your Puppy Linux files and the right your remote file storage.



A file can be transferred between the two systems by selecting it and simply clicking one of the green arrow icon buttons in the middle.

Use this method to save copies of your coursework files on the central system safely in case your PuppyLinux VM image ever becomes corrupted.

Note that the “public_html” directory shown above is the main directory for your web space. Any html files located in here would normally be publicly accessible on the university web server.

If you wish to manage these remote files via a Linux command prompt, you can do so by using Secure Shell (ssh)

```
# ssh 09876543@socweb8.napier.ac.uk
```

(where 09876543 is your matriculation number).

Another option for transferring files between your PuppyLinux VM and a remote host is by using a standard web browser.

Start the web browser

Click on the “www” icon at the left of the screen. This will start with a custom page for this module.

Select the “MyDrive” link and this will take you to your personal network file storage.

Once logged in you can upload files from, or download files to, your PuppyLinux VM.

Practice transferring some of the Linux files you have made to network storage, you will need to do this in future to keep backup copies of you coursework.