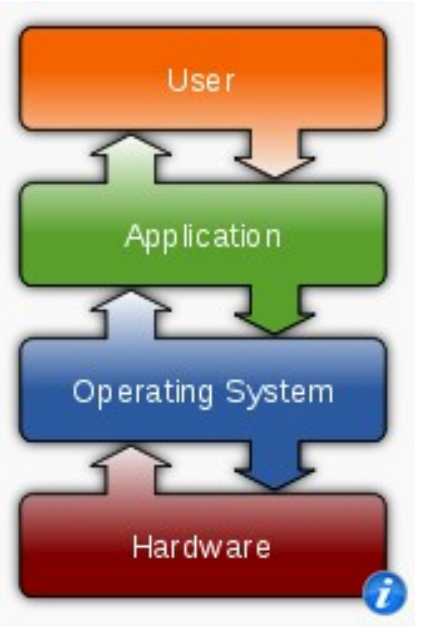
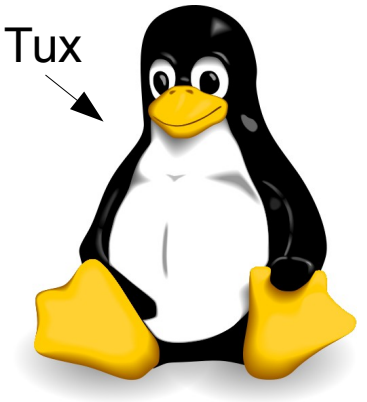
UCC IEEE Tutorial

Introduction to Linux

What is Linux?

- Linux is a computer operating system like Microsoft Windows or Apple Mac OSX.
- Free and Open-Source
- Kernel development is led by Linus Torvalds
- Uses GNU tools, hence GNU/Linux
- Many different distributions of Linux exist, .e.g. Ubuntu, Fedora, OpenSUSE, Red Hat

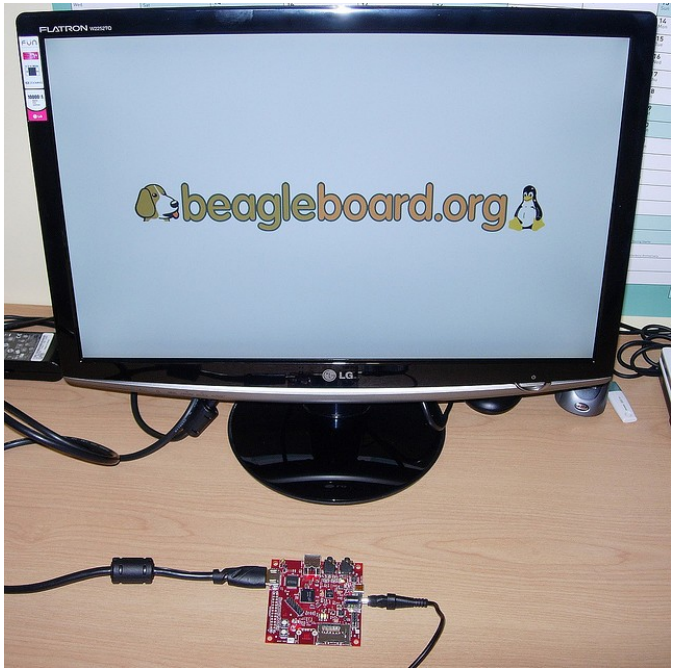




Why learn Linux?

- Linux is everywhere.
- Used in Engineering companies
- CAD tools we use for EDA
- A good programming environment
- CV and professional development

Devices running Linux



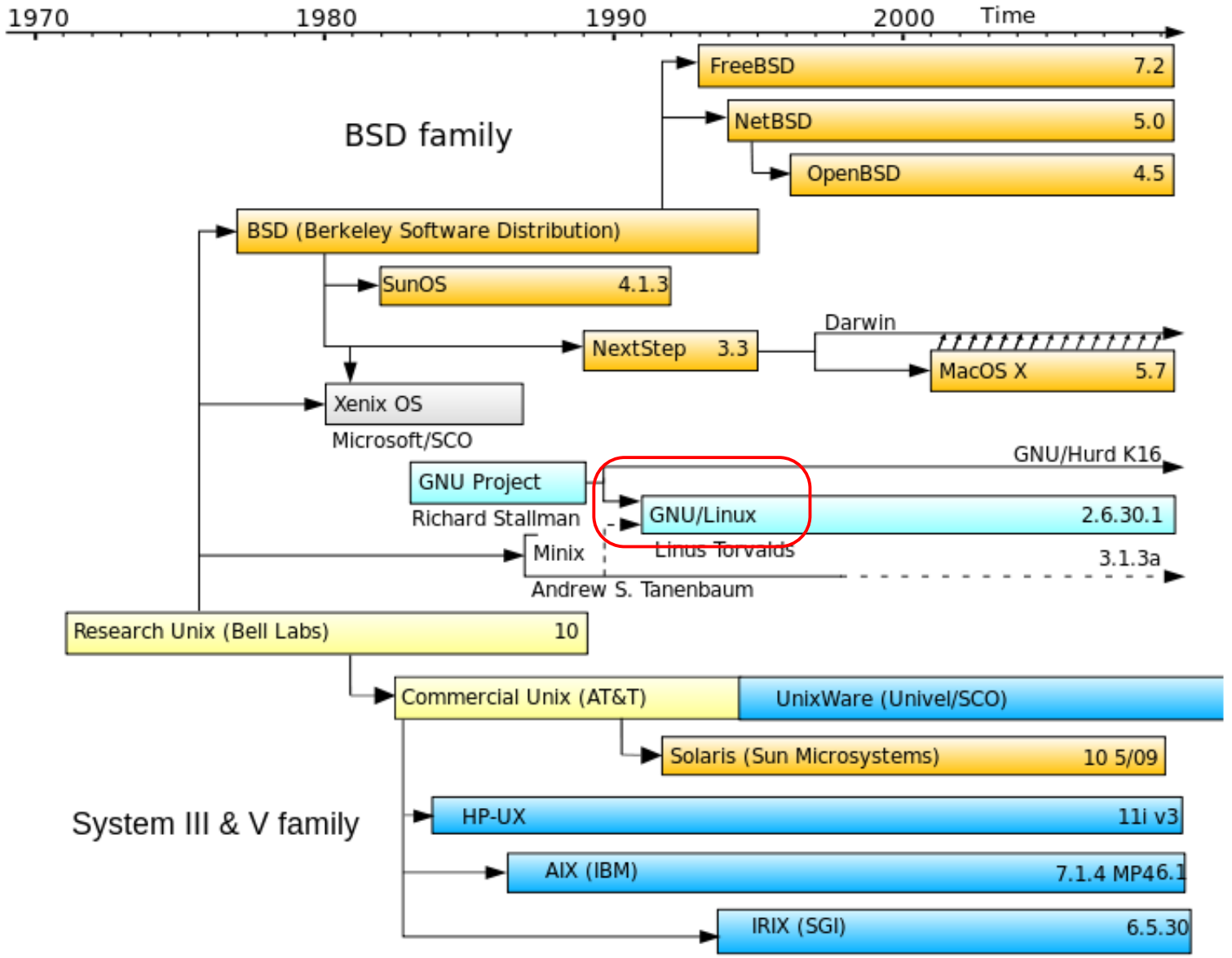
Some statistics

91% of the Top 500 Supercomputers run Linux

60%+ of web servers run Linux

50%+ of new phones sold in 2011 run Linux

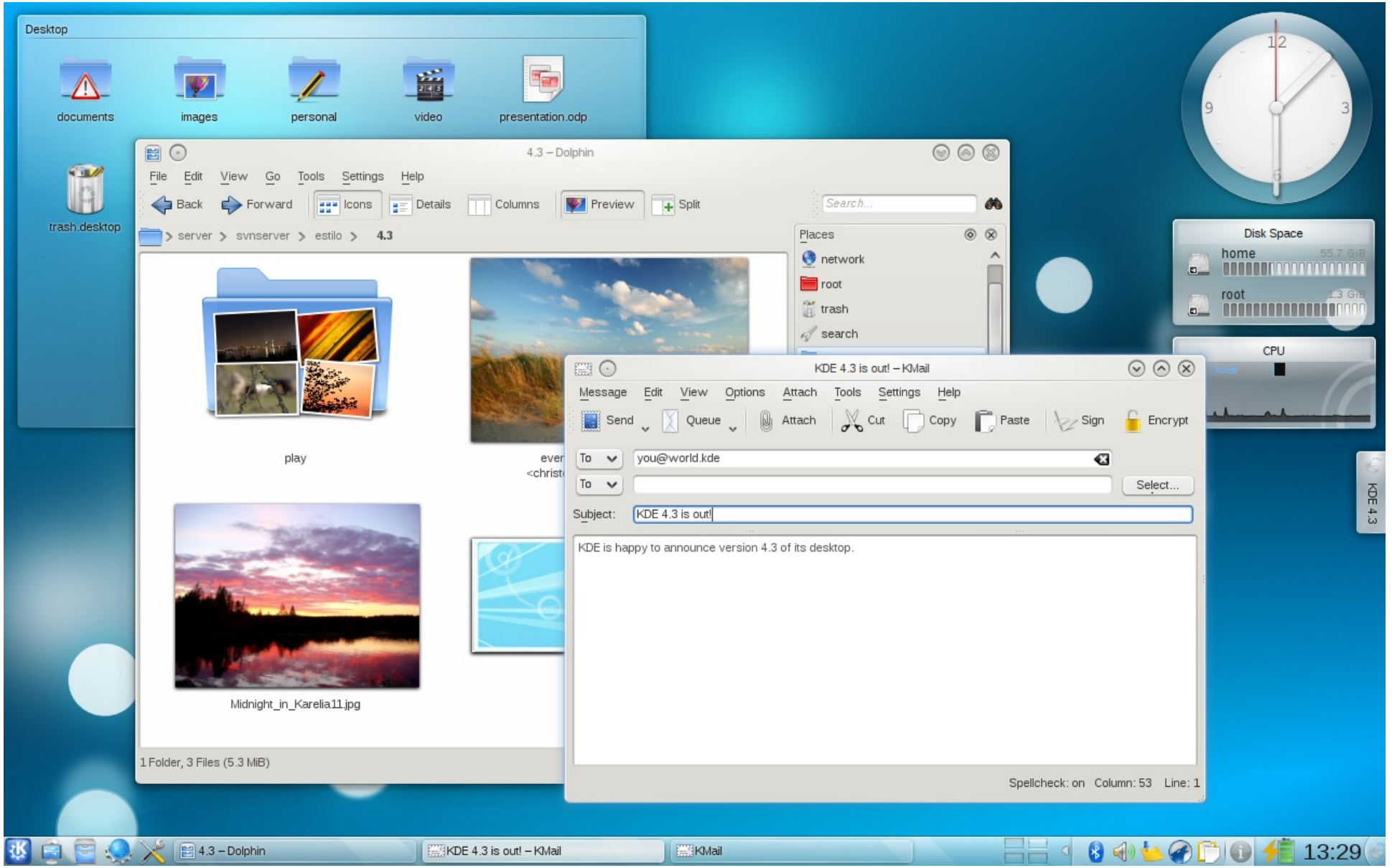
History


1970
1980
1990
2000
Time
FreeBSD
7.2
NetBSD
5.0
OpenBSD
4.5
BSD family
BSD (Berkeley Software Distribution)
SunOS
4.1.3
Darwin
NextStep
3.3
MacOS X
5.7
Xenix OS
Microsoft/SCO
GNU/Hurd K16
GNU Project
Richard Stallman
GNU/Linux
2.6.30.1
Linus Torvalds
Minix
3.1.3a
Andrew S. Tanenbaum
Research Unix (Bell Labs)
10
Commercial Unix (AT&T)
UnixWare (Univel/SCO)
Solaris (Sun Microsystems)
10 5/09
System III & V family
HP-UX
11i v3
AIX (IBM)
7.1.4 MP46.1
IRIX (SGI)
6.5.30

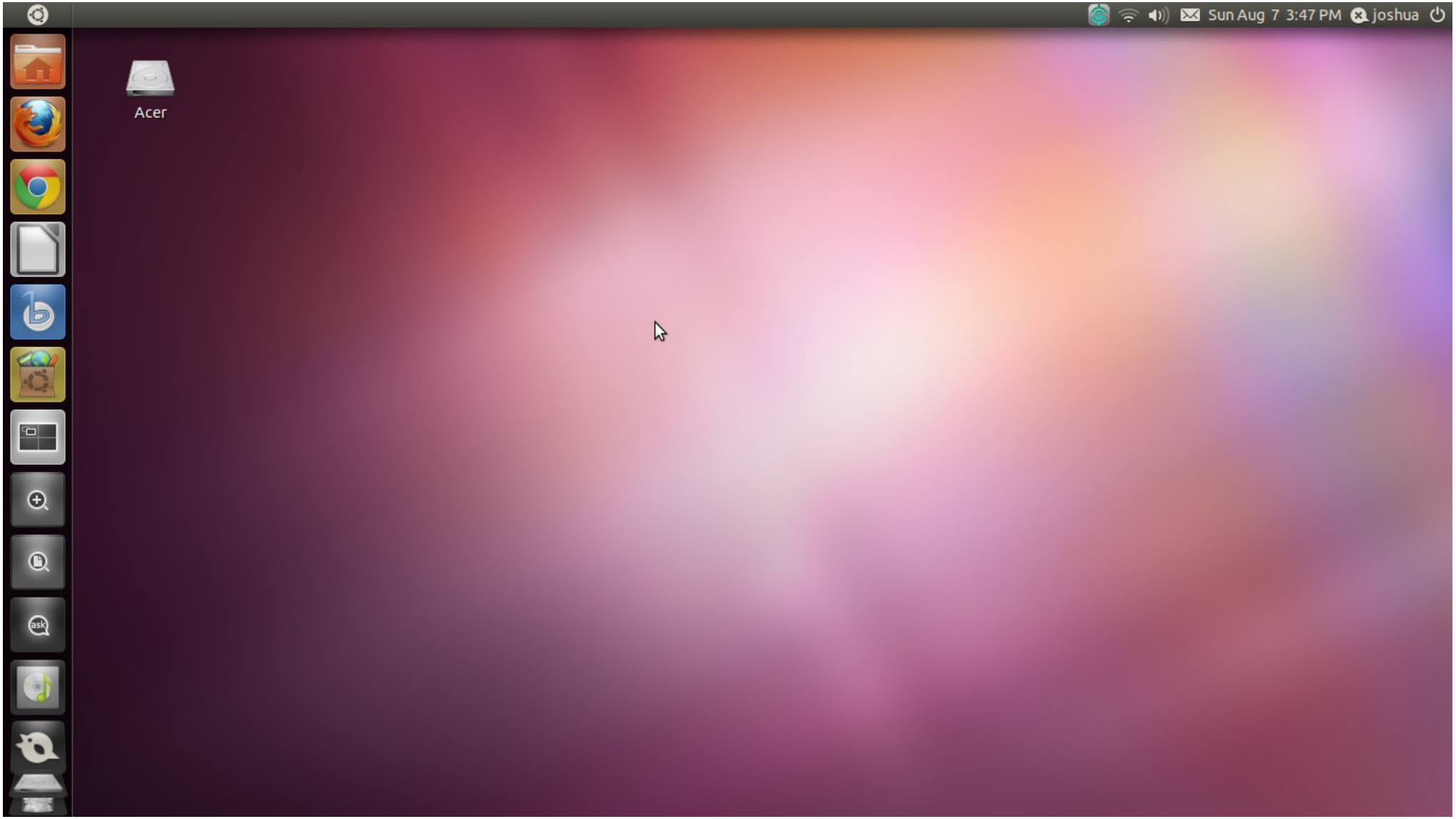
Differences vs Windows:
A users perspective

- Linux is free and highly configurable.
- Unlike Windows there are different desktop environment. We'll be using GNOME 2.
- Desktop environments are optional in Linux.
- The directory structure is different and is **case-sensitive**.
- Software is usually installed from a package manager.
- Windows software will not run (well) on Linux.

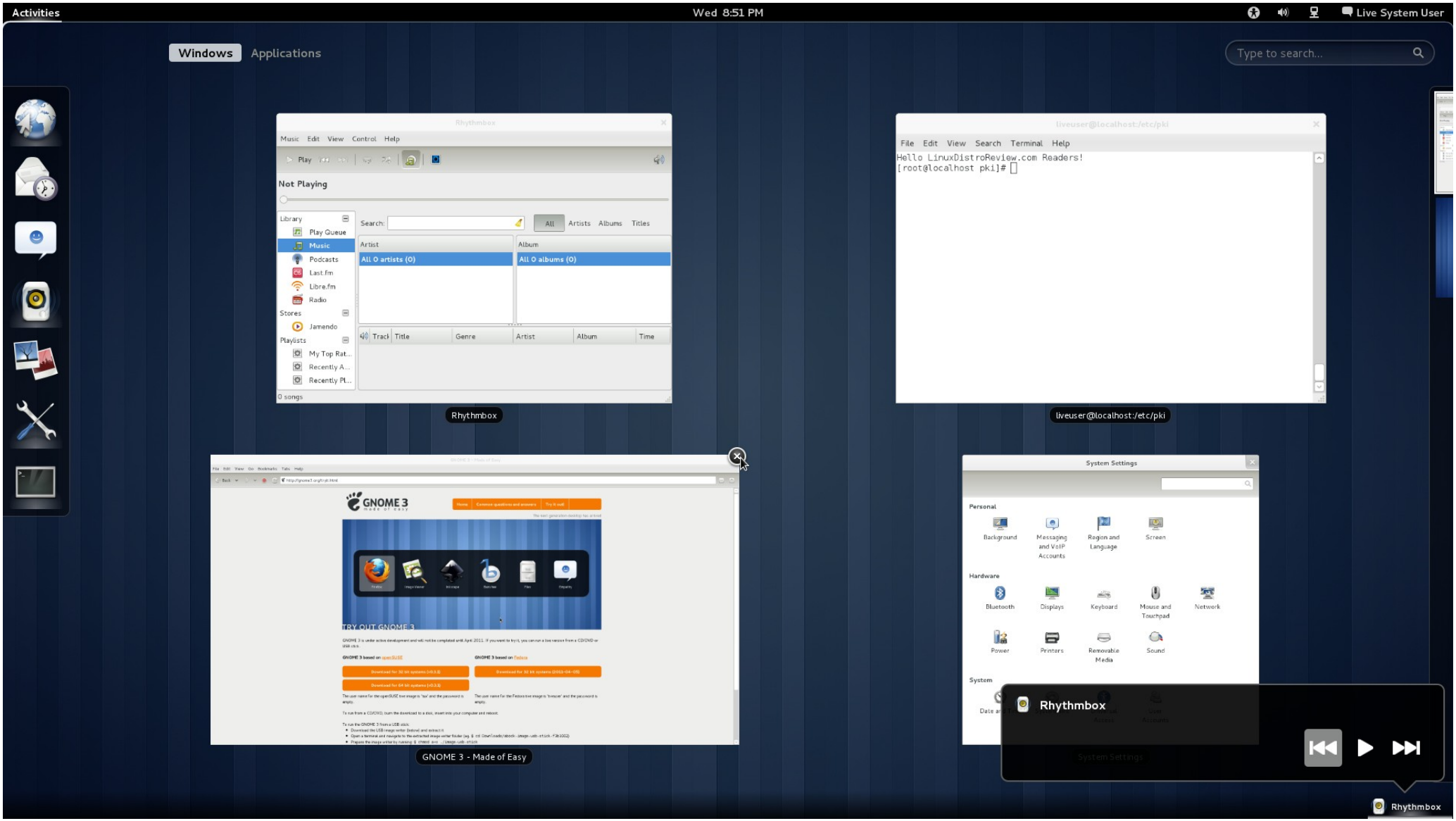
Desktop Environments: KDE 4



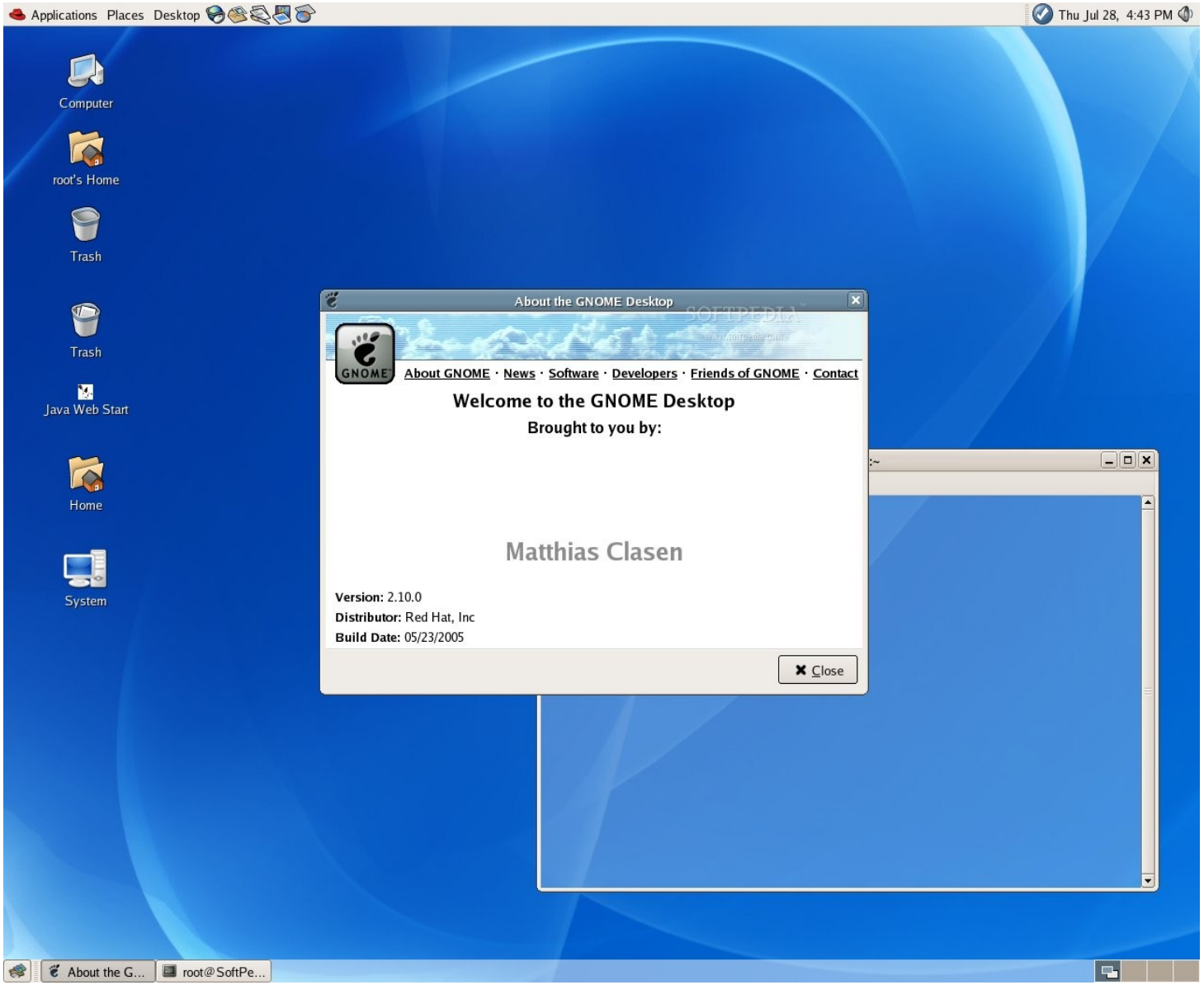
Desktop Environments: UNITY



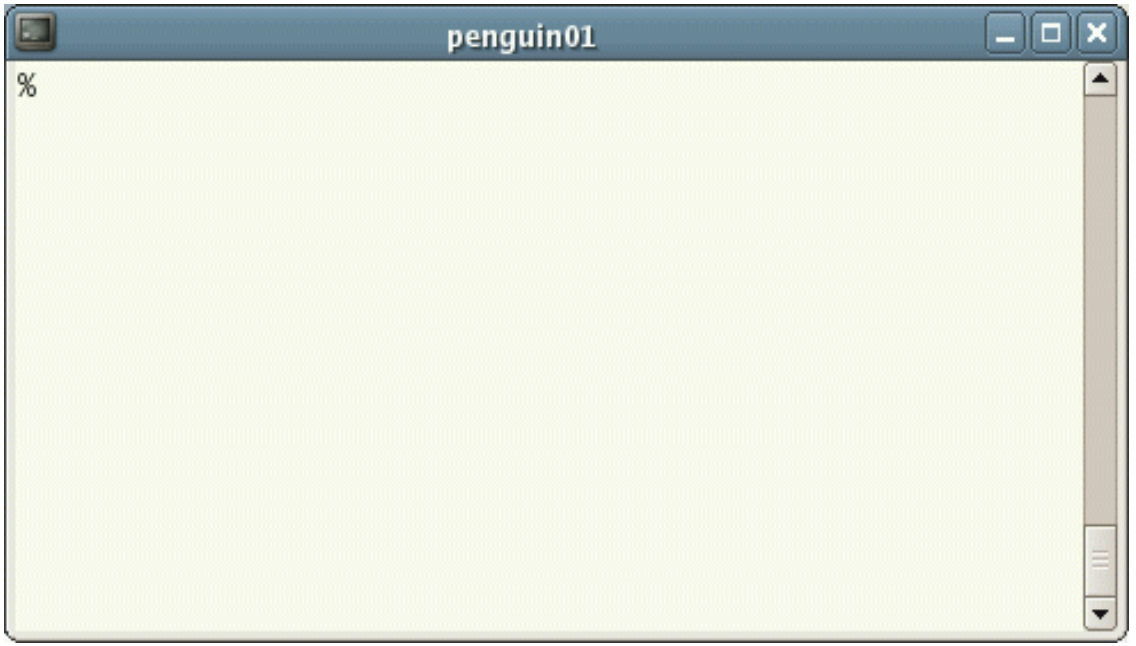
Desktop Environments: GNOME 3



Desktop Environments: GNOME 2



Shells and Command Line

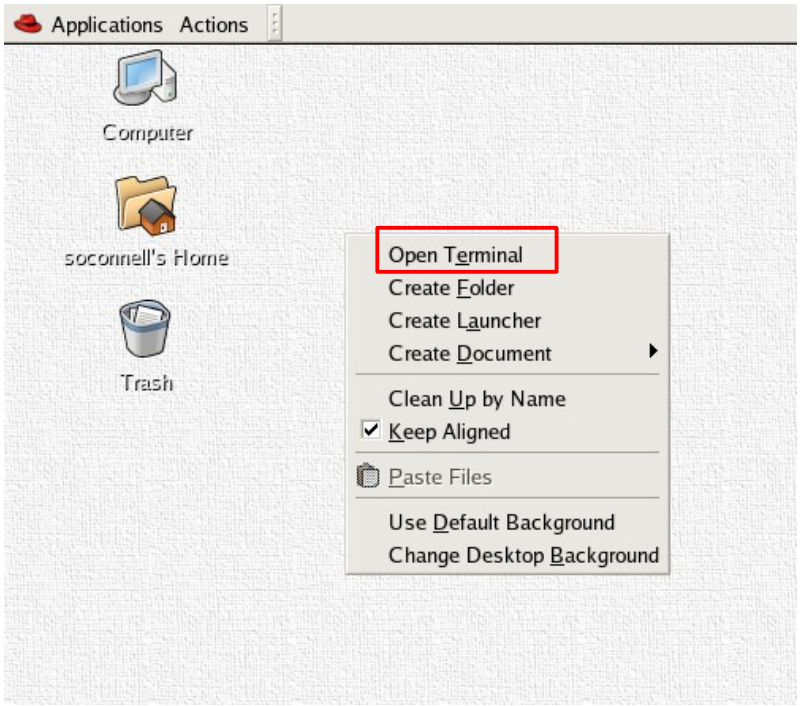


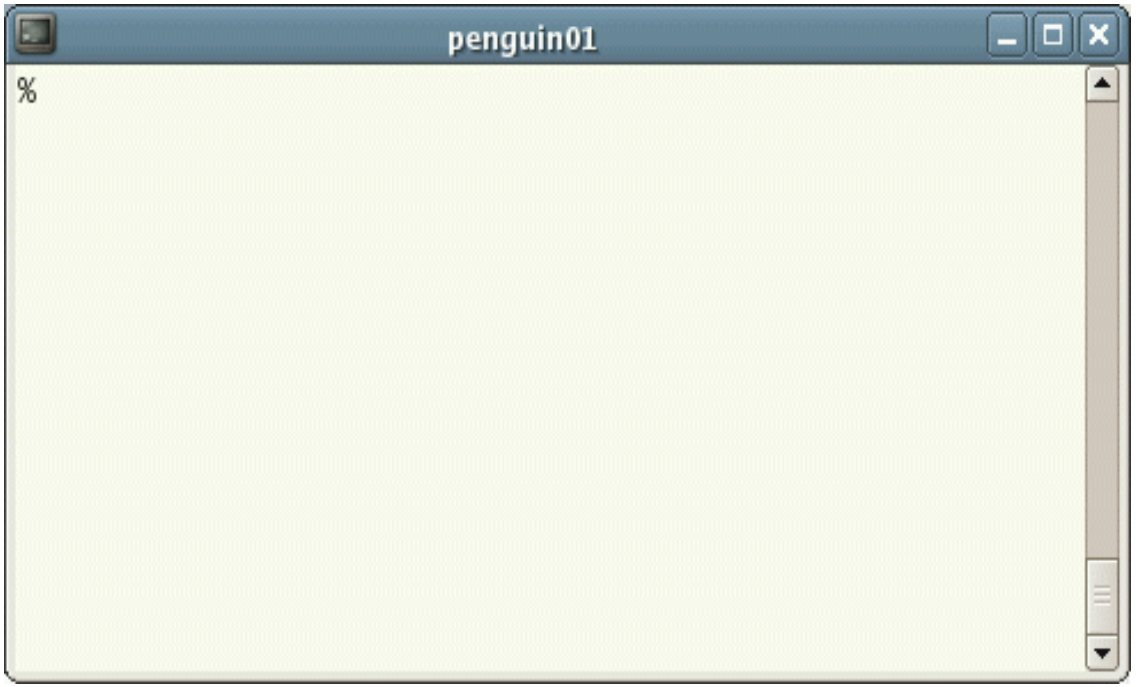
The shell allow you to type commands at the prompt
Commands can print to the terminal
You get control back when command finishes

Basic Commands

COMMAND	MEANING	NOTE
ls	List folder contents	Doesn't show hidden files
ls -al	List all contents (inc. hidden)	Displays in organised list
mkdir <folder>	Makes a directiory called <folder>	*Ex:* mkdir lab1
cd <folder>	Changes directory to <folder>	*Ex:* cd lab1
mv <file> <destination>	Moves a file to a new location. You can also specify a new filename.	*Ex:* mv res.txt lab1/ *Ex:* mv res.txt res2.txt
cp <file1> <file2>	Copies a file. You can also set location.	*Ex:* cp one.txt lab2/two.txt
pwd	Displays what directory you are in	
rm <file>	Removes/Deletes the selected file	There is no undo delete!
gedit <file>	Opens the selected file in a text editor	Output also seen in terminal
gedit <file> &	Opens the file in the background	Can still use the terminal
wget	Downloads a file	*Ex: w*get http://google.ie/index.html

1. Opening the Terminal





Right Click on the Desktop and click “Open Terminal”

2. First Command

Your first command:

whoami

This will print out your current username.

3. Print your current Directory

It's important to know your full path. The '**p**rint **w**orking **d**irectory' command will tell you.

pwd

You should be in your home directory. e.g.

/home/JSmith

Note: Differences vs Windows II

Linux

- /
- /home/john/
- /bin

Windows 7

- C:\
- C:\Users\John\
- C:\Program Files\

4. Moving around

To move to the root of the partition

cd /

To view the contents of the folder

ls

Note: Commands and Arguments

```
ls
ls  -l
ls  -a  -l
ls  -al  --color
```

```
ls  -h  or  ls  --help
```

- Commands can take arguments that modify their behaviour
- Most functions will print out help

Note: More on the file structure

/dev/
A virtual directory where your devices are listed

/etc/
Global settings, configuration files

/home/
Non-root users' homes folders

/lib/
Shared libraries (perl, python, C, etc.) and kernel modules.

/mnt/
Mounting cdroms, floppy disk drives, USB memory sticks, etc

/proc/
Virtual folder contains information about your system

/root/
Home directory for the root user

/sbin/
System programs are installed

/tmp/
Files for temporary use.

/usr/
This contains various programs, non-daemon program settings and program resources.

/var/
Log files

5. Make a folder

First return to your home directory

cd ~

then

mkdir *newfolder*

- This will create a new folder called “newfolder”.
- Now move into the folder:

cd newfolder

- Verify you’re in it

pwd

Note: Shell features

The command-line interpreter (or shell) you are using has a lot of built in feature.

There are too many to describe fully.

Here are a few useful ones.

- *Tab completion*
  Start typing the name of a file or folder and press tab to complete it.
- Searching previous commands
  <Control>+R
- Go up one directory
  cd ..
- Return to previous folder
  cd -

5. Opening and Editing a file

gedit or GNOME editor is a text editor like notepad

gedit newfile.txt

This will open the file if it exists or give you the option to create it

- This will open the file. Notice that you can no longer use the terminal session.
- Close gedit or use <Control>+<C> to kill it
- Adding an ampersand will run the program in the background

gedit newfile.txt &

6. Other Useful commands

Will make a new file by redirecting the echo command

echo hello > newfile.txt

To print the file contents use concatenate

cat newfile.txt

- This will print the text ‘hello’ into a file, creating it first if it doesn’t exist
- Using `>>` will append text to a file

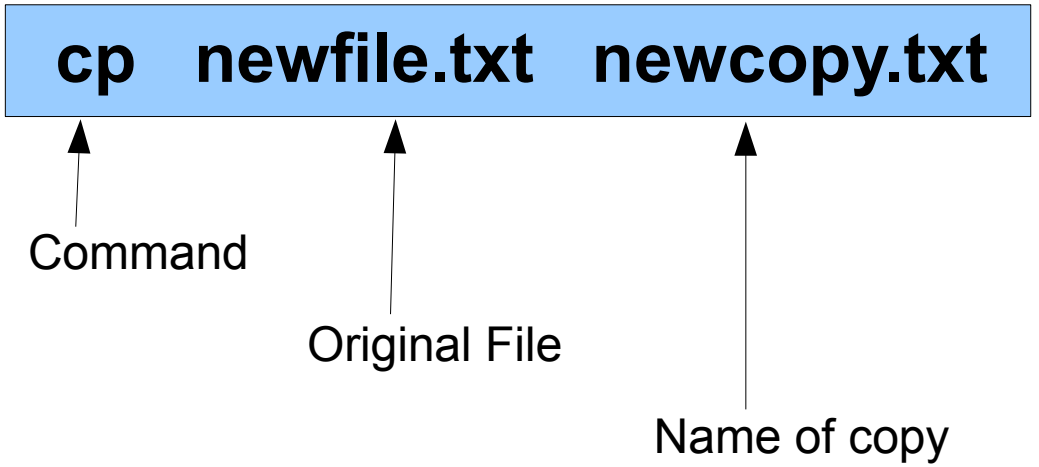
- **Hint:** You can read the manual for each command using:

  man echo (old way)

  info echo

7. Copying files

We will copy the new file with the following command



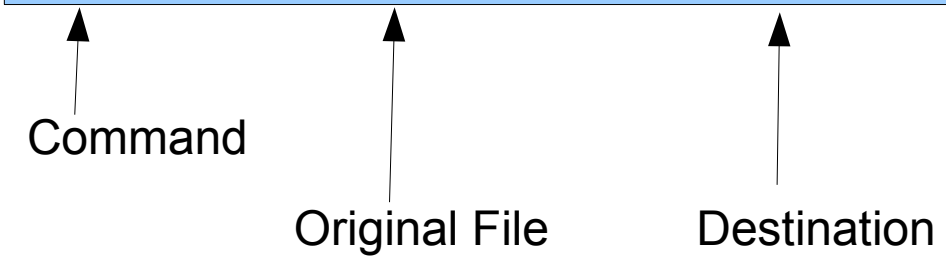
- List the folder contents before and after the command using **ls**
- Hint! To list files in a list use:

ls -l

8. Moving files

We will copy the new file with the following command

mv newcopy.txt moved.txt



Make a new folder

mkdir folder2

We can now move the file into it

mv moved.txt folder2/

9. Deleting files

We will use the **rem**ove command to delete a file

rm newfile.txt

There is no undo!

If you're unsure

Use rm -i

10. A simple script

Scripts are a powerful way of automating tasks

Open gedit and save the following as logscript.sh

```
#!/bin/bash
mkdir somelogs
dmesg > somelogs/dmesg.txt
echo 'Logs done'
```

- We now make the script executable

```
chmod  +x  logscript.sh
```

- Now we run it

```
./logscript.sh
```

Getting Linux

- VirtualBox http://www.virtualbox.org
- WUBI http://www.ubuntu.com
- Dual Boot

Thanks for your time