Linux, Raspberry Pi and Amateur Radio

Mark G8PHM, 9/2/2015

What is Linux?

...An operating system! :-)

So… what's an operating system? Once upon a time, it meant the software which enabled a user to read, write and manipulate files stored on floppy drives for micro-computers.

Now, it refers to the software which provides

- the computer user(s) with mechanisms to run any and all applications
- those applications with standard ways of accessing all connected devices, such as screen, keyboard, storage, network, and USB dongles

Some of those applications need to be tools such as assemblers, compilers, debuggers and binary editors, so that programmers can write the applications.

Most Linux code is written by volunteers, largely the code is of very high quality, but not all projects remain live, and not all projects have large development teams, so like proprietary software, it's something of a mixed bag.

Unlike proprietary software, all Open-Source can be fixed by someone with the right capability.

The most important operating systems in use in domestic and small commercial environments are:

Windows NT, Apple OSX and Linux.

Windows NT series (up to 10)

- Fully proprietary - copyright Microsoft
- Source code generally not available
- Most versions are binary compatible
- Designed for graphical UI (mouse clicks)
- Very limited text shell interface
- Very poor security record, essential to run with virus-guards and similar
- Very popular, including with virus-writers!
- Current versions have no relationship with DOS-based versions (Win 3.0 to Millenium)
- Capable of running with proper separation of ‘ring 0’ superuser and normal users, but historical applications support opposes this
- Many open-source applications and environments have been ported to Windows NT
- Very broad hardware support, most hardware provided with Windows drivers for ‘plugging into’ the kernel
- Negligible applications and library management causes extremely complex interdependencies and unreliable operating; reboots often required
- Windows NT has always been charged for

Apple OS-X (10)

- Proprietary graphical shell - copyright Apple
- Source code generally not available
- Most versions are binary compatible
- Designed for graphical UI (mouse clicks)
- Kernel is Mach/Unix kernel
- Very powerful text shell interface
- Very good security record, observes unix security model
- OS9 and earlier apps can sometimes be run on OSX through compatibility layer, but performance is limited
- Always runs with proper separation of ‘ring 0’ superuser and normal users
- Many open-source packages and environments have been ported to OSX, including the X-server by Apple themselves
- Detailed library management, applications generally work and interact properly
- Very limited hardware support, essentially to what Apple approve
- Apple OSX has generally been chargeable, but recent versions have been offered as free-of-charge updates

GNU/Linux

- Open source kernel, Linux, and open-source libraries and shells
- GNU licensing means:
  - All source code always available on request
  - Proprietary code not generally permitted
  - Any user can compile a full system from source-code, using standard tools
- There is generally no binary compatibility
- Linux provided as pre-compiled distributions, several thousand applications
- Very powerful text shell interface, and can be run completely ‘headless’
- Very good security record, observes unix security model, no known effective viruses and no virus-guard required
- Always runs with proper separation of ‘ring 0’ superuser and normal users
- Very powerful graphical interfaces are available, including Gnome and KDE, both of which offer similar experience to OSX or Windows
- Detailed library management, applications generally work and interact properly, dependencies are automatically satisfied
- Hardware support mixed, generally excellent for older hardware, but can be limited on more modern devices
- Linux is generally free of charge

Other operating systems

- SunOS/ Solaris - form of Unix - only available for Sun Hardware
- VMS - Vax mini-computer operating system, exceptionally secure and stable
- Minix - Open-source unix-like operating system, designed for training and education

- CPM - Control Programme for Microcomputers - originally 8080 but available for other chips including 8086. Still possible to boot on some PCs
- QDOS - Quick and Dirty Operating System - simple development of CPM
- MS-DOS - Microsoft copy of QDOS, very popular in 80s and 90s, used with Windows 2,3.0,3.1, 3.11 series
- DR-DOS - Digital Research DOS, derived from CPM-86, supports most MS-DOS applications, generally better memory and device management than MS-DOS
- FreeDOS - Open source copy of DOS system calls, with real-time disk drivers - fully functional and still under development
- Psion OS/Symbian - single user palm OS, used by Nokia for smartphones
- RiscOS - Acorn OS for ARM processors, largely replaced by Linux
- Android - Proprietary top-end wrapped around GNU/Linux core

Linux is great for...

- Older hardware - can breathe life into old PCs and laptops
- Very tight security requirements - security is excellent and specialist distributions offer fully encrypted storage, secure browsing, access to secure email networks and more
- Low cost installations - Linux is generally free of charge
- Remote or headless running - Linux doesn’t need a GUI to run, it doesn’t even need a mouse or keyboard to be connected
- Very high-performance servers and super-computers - Linux is very lightweight and configurable; powers Google and many super-computers
- Fixing computers using LiveCDs, including Windows installs in some cases
- Multi-boot environments, such as Linux/Windows, or Linux/OSX/Windows
- Wide range of hardware architectures, including
    - x86, x86-64, Arm (Raspberry Pi, most mobiles), Mips (used in many small routers and similar), PowerPC (older macs, some smaller devices), Sun Sparc, IBM S/390 (mainframes)

Linux is not best for...

- Windows applications or Apple applications -
  - WINE emulator can run some Windows binaries, but is difficult to install, and compatibility is limited
  - DOSEMU can run many DOS binaries, but again usability and compatibility is limited.
- Randomly installing!
  - Be very careful! Many installations will wipe all data on hard drives, so you can lose everything - do not install unless you are confident that you have everything backed up.

Linux Distributions.

- Linux is generally available to download and install from the web,
  - run from a liveCD,
  - or boot from a USB stick or similar.
- There are several distributions available, which fall into 3 main categories, based on the method used to track dependencies
  - Debian based
    - Debian itself, Ubuntu, Kubuntu...
  - Red Hat based
    - Fedora, Pidora, SuSE...
  - Source based
    - Gentoo, Linux From Scratch
- I personally prefer Debian-based, but all have their advantages.
- The above are all available free of charge, with source code, and all have stellar security records.

Package Management.

- Distributions such as Debian, Pidora, Ubuntu all have methods for
  - installing software for users - editors, web browsers, specialist apps
  - installing updates for bug fixes and security fixes
  - performing full distribution upgrades
- Debian distributions have both GUI and Console based management
  - Synaptic - debian GUI manager
  - Muon - KDE package manager
  - Aptitude - menu-based text manager for Debian
  - Apt-get - command-line based manager for Debian
- All managers provide the user with
  - control over the in-built package management system - reporting, package selections
  - capability to install packages, generally digitally signed, from
    - repositories on the internet
    - local files,
    - CDs and flash storage
    - **Comprehensive database of all installed files - every package can be completely removed from a linux system.**

Documentation and help

- Help for linux comes many sources, including the following:
  - How-to guides - Linux Documentation Project
    - /usr/share/doc/HOWTO/…
  - man pages - built-in help from the command line, generally brief reminders of key functions
  - package readme files and other material
    - /usr/share/doc/<package name>/…
  - Linux Gazette - free online magazine hosted by the LDP
  - Commercially published books, particularly O'Reilly books
  - Commercial magazines such as Linux Format, Linux User and Developer, available from WHSmith
  - Raspberry Pi magazines - also from WHSmith
  - The internet - try Google!
  - Usenet - comp.os.linux.<various> - freely available newsgroups, just install a reader

What about my standard packages?

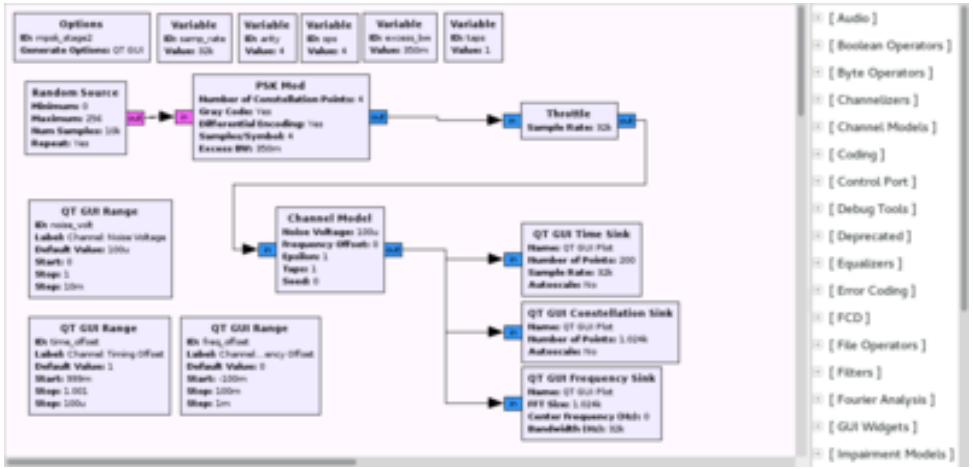
- Web browsing - Firefox, Google Chrome, Opera
- Email - Mozilla Thunderbird (from the Firefox team), Google mail (online)
- Document preparation - Libre Office - reads most Microsoft Word formats, Google Office
- Spreadsheets - Libre Office , Google Office
- Presentations - Libre Office , Google Office
- Instant Messaging - Pidgin, Kopete [support most types of IM in one package]
- Video/Audio players - mplayer - most video formats, Amarok, Audacious for audio
- Audio editing - Audacity
- Video editing - Blender - many others
- Audio mixer/recorder - Ardour

Amateur Radio Packages for Debian

- Several packages, all free, including:
  - Chirp - set up and programme many modern rigs, including Baofengs
  - Antenna modelling software, Nec, Nec2 and visualisation with Antenavis
  - APRS support, including gateways
  - AX25 support (if anyone still uses it!)
  - fldigi - supports many digitial modes, including BPSK
  - smith chart calculators
  - CW trainers
  - various Logging software packages
  - Rig control software
  - various SDRs
  - SSTV support
  - weak signal mode support
  - Path analysis software (terrain analysis)
  - Gnu Radio Project - the ultimate software radio - see next slide!
- Sites such as QRZ.COM, RSGB and so on all available using Firefox of course.

Gnu Radio Project - the ultimate SDR?

- http://en.wikipedia.org/wiki/GNU_Radio
- Graphical interface to a wide range of signal processing capabilities, like this:



- Includes, FFTs, scopes, waterfall charts and so on.
- Can take data from SDR USB chips.
- Not for the faint of heart!

Raspberry Pi

http://www.raspberrypi.org/help/faqs/

Single board computer:

Ethernet for network, USB ports (more on later models), audio out (USB dongle for audio i/o), PAL and HDMI video ports.

SD card slot for storage of operating system and data

Relatively inexpensive, around £25 - £45 for the boards, or typically < £100 for a full kit with PSU, keyboard, mouse, board, SD card with OS installed.

Generally runs Linux, the free operating system!

The seminal website for Amateurs is g4wnc.com

Arduino vs Pi

Arduino is a superb single-board micro-controller which uses an Atmel core.

Arduino typically doesn’t run an operating system, rather, the software is written to run directly on the controller.

Arduino would normally be programmed from another machine, such as a Linux, Mac or Windows machine.

Arduino is ideal for time-critical controllers where a non-real time operating system would cause problems.

Pi is a relatively low-power personal computer, designed to run an operating system, typically linux. It’s based on Arm processors, similar to those used in high end smartphones and tablets.

You could probably programme Arduino from Pi, but probably not the other way around!

Use my Pi - side by side demo

Pi can be used locally with a keyboard, mouse and monitor.

An HDMI monitor is best as the resolution is excellent. Analogue video can be connected from the phono socket to a television.

There are HDMI to VGA adaptors, but be sure to choose one known to support Pi.

Any standard keyboard and mouse with USB connectors is find.

Earlier Pis come with only 2 USB sockets, so a powered USB Hub is very useful, any type will suffice.

Pi has ethernet as standard, but you can add wifi with ‘Wi-Pi’ available from Farnell/Element 14. The farnell site has further instructions.

Accessing Pi 1/4

Remote Pi operation can use either command line (CLI) with ssh. Linux and Mac come with terminal programmes and ssh built-in, but you must have a network.

For Windows, you'll need to install 'putty' from www.putty.org, it's free of charge.

Or you can use a remote graphical destop; most Pi builds will support VNC.

For Windows, you can install TightVNC viewer from www.tightvnc.com

Both SSH access and VNC access will need to be set-up on the Pi. To enable ssh, from the cli, type: `$>sudo raspi-config` and, go to the Advanced menu, select SSH and enable it.

For vnc, type `$>sudo apt-get install tightvncserver`

Accessing Pi 2/4

Now you merely need to find the Pi on the network. This means you need to know its IP address. Most networks use a server called ‘DHCP’, which automatically provides IP addresses on demand as machines connect. These addresses are randomly selected from a large pool, so are difficult to guess.

There are some clever tricks with ‘arp’ tables, such as the following from a Linux machine, but that assumes you know the ethernet mac address range of the network card:

```
$> arp -a | grep b8:27:eb | cut -d " " -f1
```

The simplest is to connect a monitor and keyboard to the Pi, find the address using `$>ip addr list` at the command line. This may change on reboot.

The next simplest approach is to fix the ip address manually. You must be careful, though, to avoid the address range your DHCP server.

Accessing Pi 3/4

Fixing the address manually requires editing the config file using nano editor as follows:

```
$> sudo nano /etc/network/interfaces
```

Put a hash at the start of this line to prevent the DHCP client from running:

```
# iface eth0 inet dhcp
```

The add this stanza, but you must have the correct network, gateway, broadcast and ‘address’ quads:

```
iface eth0 inet static
     address 192.168.1.99
     netmask 255.255.255.0
     network 192.168.1.0
     broadcast 192.168.1.255
     gateway 192.168.1.1
```

Type `ctrl-x` to save the changes, then reboot `$> sudo shutdown -r now`

Accessing Pi 4/4

Alternatively, you can use a full-blow network scanner, the best is probably nmap, available from nmap.org and runs on Linux, Mac and Windows.

Follow the instructions for Mac or Windows to use the ‘executable installer’.

For Debian Linux, $sudo apt-get install nmap will do the trick. The following command line on linux or mac will find any computer ‘listening’ or port 22, the ssh port:

```
$>sudo nmap -T5 -n -p22 --open --min-parallelism 100 192.168.1.0/24
```

Simple ssh to each found address until you find the pi - easy!

`$> ssh -lpi 192.168.1.pi` note that the password by default is `raspberry`

Amateur Radio Pi Applications

Include:

- PSK31 terminal
- RTL radio server
- WSPR beacon (another day)
- Security Camera (another day)
- Repeater controller (another day)

PSK31 terminal

- You'll need a sound card. The Daffodil US01 card is presently £5.05 from Amazon.
- You'll need to install fldigi with the following command:
  - `$>sudo apt-get install fldigi`
- start it:
  - `$> fldigi`
    - and follow the instructions.
- You'll need audio cabling from your rig to and fro the sound card ,and you'll need to set the rig to 'vox' for transmitting.
- Check the ALC level, and adjust it for none. Do not use more than 25% of your transmit power!

RTL Dongle Server

- The dongle, based on the RTL2832U chipset - R820T2 ~£10
- UHF or BNC to MCX adaptor lead, ~ £3
- then install the drivers on Pi using the following procedure
  - `apt-get install cmake`
  - `apt-get install libusb-dev`
  - `git clone git://git.osmocom/org/rtl-sdr.git`
  - `cd rtl-sdr/`
  - `mkdir build`
  - `cd build`
  - `sudo cmake ../ -DINSTALL_UDEV_RULES=ON -DDETACH_KERNEL_DRIVER=ON`
  - `make`
  - `sudo make install`
  - `sudo ldconfig`
- next, reboot Pi, plug in your dongle, and then run the server:
  - `rtl_tcp -a ‘ip address’`

So what's my ip..?

- You can get around this with a short shell script as follows:
  - `#!/bin/sh`
  - `ip add list | grep inet | grep -v 127.0.0 | cut -d " " -f 6 | cut -d "/" -f 1`
- Type this in using an editor such as nano (or vi for the brave!)
- Save it as piip, from Nano:
  - `ctrl-o`
  - `piip`
  - `<enter>`
  - `ctrl-x`
- Then make it executeable:
  - `$chmod a+x piip`
- Now you can run the server like this:
  - `do /usr/local/bin/rtl_tcp -a $(piip); done`
- This will keep automatically restart the server should it crash, with the proper IP address

Using the server from Mac or Linux

- Install the gqrx sdr package
  - This will also work well with Funcube
  - It's based on Gnu Radio
- Configure i/o devices
  - `Device: other`
  - `Device string: rtl_tcp=192.168.1.89:1234 [ use your proper IP address for your pi]`
  - `Sample Rate:  1200000 [ pick from the list ]`
- Now press the 'on' button on the top left
  - `There are various controls including RF and AF gain, AGC settings, mute (squelch), DC cancel and others`

Using the server from Windows

- Make sure you have very good virus protection enabled
- Install SDRSharp guide here: http://www.atouk.com/SDRSharpQuickStart.html
- Under the ‘front-end’ section in the radio panel, select rtl_tcp

Summing up

The development kit can be set up for an investment of around £50 to £100 depending on whether you already have such as mice, keyboards and power supplies, plus the cost of a monitor - again, re-use where you can.

Once you have invested in the development capability, an RTL-SDR receiver could be managed for under £30, with coverage of 24MHz to approximately 2GHz, or a PSK 31 terminal for a similar outlay, re-using an existing monitor.

Have fun with your Pi, remember, more than one is not expensive!