Distributions and Platforms

COMP8440: FOSSD
Lecture 11

CC BY SA

Not just Linux

FOSS projects exist for a number of platforms:

- Microsoft Windows
- Apple MacOSX
- Various proprietary Unices
- PDAs, phones and mobile devices
- specialised O/S's (mainly real-time)

FOSS O/S's

- Linux
- FreeBSD, NetBSD, OpenBSD, others
- Minix
- FreeDOS
- OpenSolaris
- Darwin/OpenDarwin
- L4
- Plan 9

http://en.wikipedia.org/wiki/Comparison_of_open_source_operating_systems

http://en.wikipedia.org/wiki/File:Unix_history.en.svg

Linux Distributions

(strictly, Linux is just the “kernel”, rest is generally GNU and other utilities)

Can classify distributions by package management:

- Those using Red Hat Package Management (RPMs):
  - Red Hat (RHEL), CentOS, Fedora
  - SuSE, OpenSuSE
  - Mandrake/Mandriva
  - many others
- Those using Debian (.deb) packages:
  - Debian GNU/Linux
  - Ubuntu, Kubuntu, Xubuntu, Edubuntu ...
  - Knoppix
  - many others

http://upload.wikimedia.org/wikipedia/commons/8/8c/Gldt.svg

Other Linuxes

Many distros may not use package management at all:

- Slackware (oldest current distro)
- Gentoo
  - Uses “portage” system to build apps from source
- Various embedded distros which are custom built for purpose

Other Classifications

Can also classify distributions by level of commercial support:

- Redhat RHEL, SuSE etc.
- most everything else

Could also classify by:

- server vs. desktop
- stable vs. development
- hardware architecture (CPU) supported
- “liveboot” vs. traditional install
- strictness of adherence to FOSS principles

Devices

- FOSS is rapidly gaining in the very large “embedded devices” “market”
- Mainly embedded Linux using a uCLinux kernel (no hardware memory management)
- Routers
- STBs/PVRs/TVs (Tivo, Beyonwiz etc.)
- mobile phones (Motorola, others)
- PDAs/tablets/eBooks (Nokia, Sony, others)

Devices 2

Main advantages to manufacturer:

- no per-unit license fees to pay (!)
- less development time – faster to market
- lower-cost development – large community to “help”

Main disadvantages to manufacturer:

- need to provide source code – may cause licensing/patenting issues
- hobbyists may modify and “break” products – possible additional after-market support costs

Embedded GPL Violations

<some user> wrote:

Beyonwiz are obligated to release the full source code and all required tools to build the base Linux firmware that the unit is built on top of. They are already in violation of the GPL license by not explicitly making a written offer to do so.

<Beyonwiz manager/engineer> replied:

Thank you <some user> for your kind interest and cooperation always. As for the GPL, we are intending to open the related things later through our web site, however I think it will be not so much helpful for users to do somethig because our release codes will be quite partial and limited ones which has purely and only related to the GPL to do it (There are so many other codes we have created under Sigma Design's SDK too). So, we are carefully considering to release something useful application programs like Topfield TAP later after clearing most of bug list and wish lists requested by users and by ourself. Kindly let us work best to provide the most stable firmwares with many useful features at this moment under forum members great cooperation. We will always do best to satisfy our product users.

Searching for GPL code

- Many devices may contain GPL or other FOSS code, but may not make that apparent
- How to find out if a device has FOSS code?
- Search the web – see if someone else has already found out
- Look for hints in user and other interfaces
- Search the firmware, if available, for hints of known GPL'd software, or for license strings

Porting

A great challenge for a hardware oriented "hacker" is to port a FOSS O/S to a piece of otherwise proprietary hardware

- some BSD on an original Apple Airport
- HP/Compaq iPaq (Familiar Linux)
- replacing proprietary O/S on laptops (eg. my MacBook Air)

What issues are there in porting to a new platform?

Porting Issues

- getting access to documentation, preferably without signing a Non-Disclosure Agreement (NDA)
- developing novel debugging and testing techniques
- setting up a “toolchain”
- getting assistance from the hardware vendor
- getting a user base to test the port