Desktop Linux Technologies

System Administration DeCal
Week 10
Spring 2010

Final Projects

- Start doing 'em!
- Ask me questions!
- Presentations in 2 weeks- you don't present, you don't pass.
- If you need anything, come and ask me **today** after lab.

Last Time

- Talked about init, or how to start the system
- Used in most variants of Linux/Unix, though being phased out by newer technologies (eg Upstart)
- What else do you need to start the system?

“Desktop Experience”

- How do we define the “desktop experience” in Windows or Mac OS X?
- Generally considered to be things like a desktop, windowing system, audio system, graphics system, etc.
- Also includes things like icons, menus, taskbars, etc (collectively the GUI)

“Desktop Experience”

- Windowing system (X11)
- Window manager (the desktop)
- Drag-and-Drop
- Event notification (D-bus)
- Plug and play devices (udev)
- Easy network/user/system management GUI

X11-Windows

- Drives your video card, supports your mouse, supports each window
- The backbone of a Linux desktop
- Client-server networked model lets you use it over a network, like VNC
- Pros: Very, very flexible
- Cons: Network use requires a lot of bandwidth!

X11 tricks

- X can be forwarded over SSH (e.g. running Emacs from home)
- Lying on bed, computer on desk: set DISPLAY variable during ssh session to start new programs on your desk machine
- It’s great for being lazy with Unix!

Window manager

- OK, you have your windowing system. Now what?
- You can’t drag windows around, resize them, hide them, bring them forward or shift them backward.
- A window manager is needed to handle all this.
- examples: Metacity (GNOME), kwin (KDE), xfwm4 (Xfce)

The Earliest Window Manager: twm

Device Enumeration

- Remember what we learned about /dev
- Each file represents the input/output of a particular device on the system
- So to represent all the possible devices you could have, you need an enormous /dev folder - even if they aren’t plugged in
- That’s only the tip of the iceberg...

Device Enumeration

- Problem: A new Linux device driver comes out, complete with new minor/major device numbers. You then need to create entries in /dev yourself. Major pain.
- Problem: You have two USB keys. Insert key #1 first and then #2. They are “sda” and “sdb”. One day, you insert them in the wrong order and #1 = “sdb”, #2 = “sda”. You format the wrong key and lose data.

udev

- Works with the Linux kernel directly
- Creates only those device nodes that are needed at any given point
- Can assign unique identifiers to certain devices based on ID (avoid the sda/sdb problem)
- Can do certain things like tell the desktop system when you insert a USB key