

Today:

- Introduction to Unix/Linux (80 minutes)
- Peer Evaluation Criteria (20 minutes)

Introduction to Unix (especially Linux)

Overview

- Some history
- Basics of Linux for this class
 - Two approaches
- Demonstrations with both approaches

When did Unix appear?

When did Unix appear?

Late 1960s, early 70s

http://www.unix.org/what_is_unix/history_timeline.html

When did Unix appear?

Late 1960s, early 70s

http://www.unix.org/what_is_unix/history_timeline.html

When did Windows appear?

When did Unix appear?

Late 1960s, early 70s

http://www.unix.org/what_is_unix/history_timeline.html

When did Windows appear?

First shipped in 1985

<http://windows.microsoft.com/en-us/windows/history>

When did Unix appear?

Late 1960s, early 70s

http://www.unix.org/what_is_unix/history_timeline.html

When did Windows appear?

First shipped in 1985

<http://windows.microsoft.com/en-us/windows/history>

When did the first Apple computer appear?

When did Unix appear?

Late 1960s, early 70s

http://www.unix.org/what_is_unix/history_timeline.html

When did Windows appear?

First shipped in 1985

<http://windows.microsoft.com/en-us/windows/history>

When did the first Apple computer appear?

Apple II publicly announced in 1977

https://en.wikipedia.org/wiki/History_of_Apple_Inc.

So Unix is, roughly, the oldest widely used operating system.

What about Linux? (And why do I refer to it, usually, as GNU/Linux?)

Unix is proprietary; GNU is a project to create a free version of Unix. **When did that start? (And what does GNU stand for?)**

So Unix is, roughly, the oldest widely used operating system.

What about Linux? (And why do I refer to it, usually, as GNU/Linux?)

Unix is proprietary; GNU is a project to create a free version of Unix. **When did that start? (And what does GNU stand for?)**

Announced in 1983: “GNU’s Not Unix”

<http://www.gnu.org/gnu/gnu-history.en.html>

OK, so **what about Linux?**

OK, so **what about Linux?**

The GNU project implemented many parts of the Unix operating system, but not the *kernel*. A Finnish college student made a first attempt at that... **who and when?**

OK, so **what about Linux?**

The GNU project implemented many parts of the Unix operating system, but not the *kernel*. A Finnish college student made a first attempt at that... **who and when?**

Linux Torvalds started the project in 1991 and released it as free software in 1992. Technically, Linux is the kernel, and the other services are GNU. Thus, GNU/Linux.

https://en.wikipedia.org/wiki/History_of_Linux

Wait... what's a kernel? For that matter, what's an operating system?

Wait... what's a kernel? For that matter, what's an operating system?

From the reading:

An operating system (OS) is a **resource manager**. It takes the form of a set of software routines that allow users and application programs to access system resources (e.g. the CPU, memory, disks, modems, printers, network cards, etc.) in a **safe, efficient** and **abstract** way.

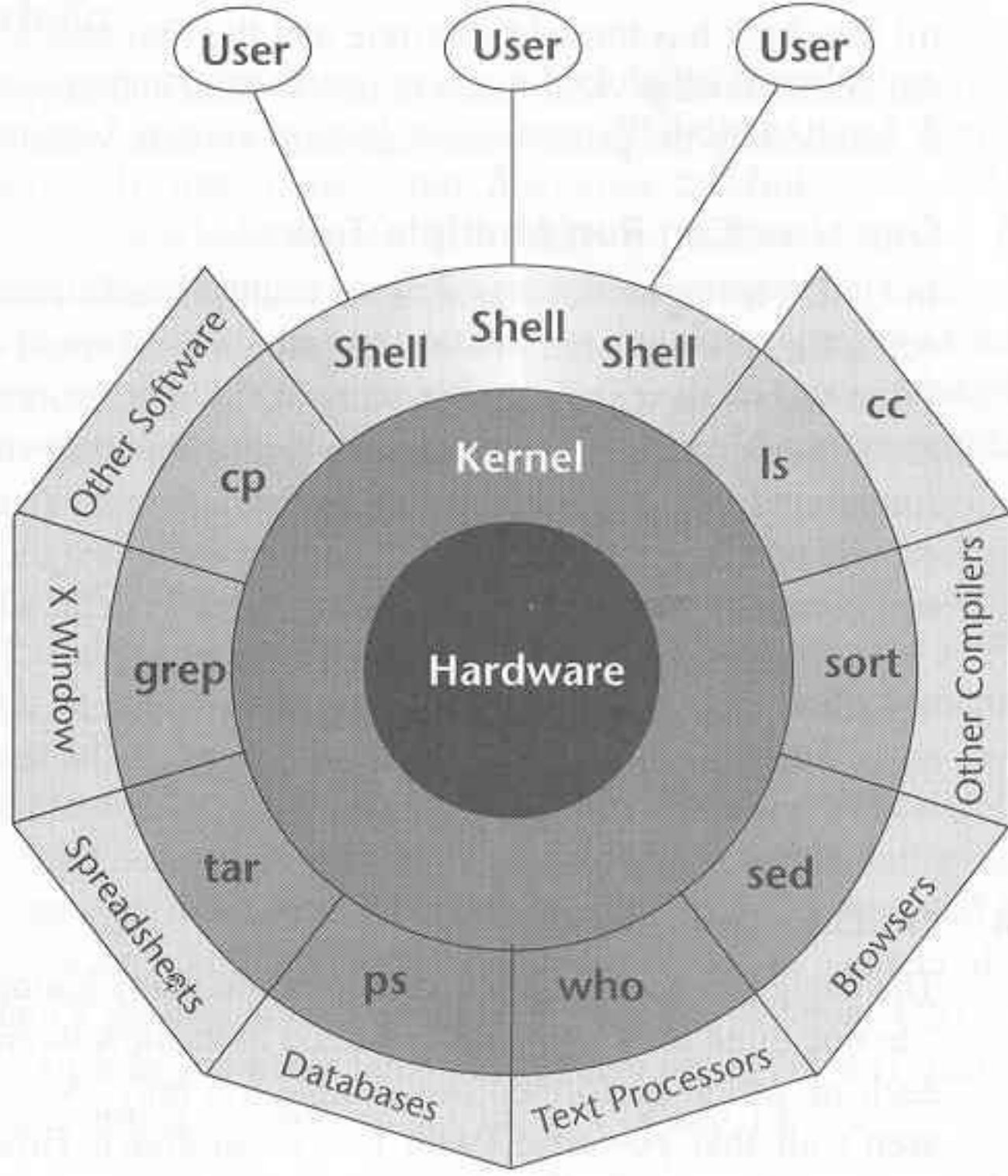
<http://www.doc.ic.ac.uk/~wjk/UnixIntro/Lecture1.html>

safe access to a printer by allowing only one application program to send data directly to the printer at any one time.

efficient use of the CPU by suspending programs that are waiting for I/O operations to complete to make way for programs that can use the CPU more productively.

provides **abstractions** (such as files rather than disk locations) which isolate application programmers and users from the details of the underlying hardware.

Uh, so what about the kernel?



By the way, GNU/Linux is free software (vs. Unix, which is proprietary). What does that mean?

By the way, GNU/Linux is free software (vs. Unix, which is proprietary). What does that mean?

“Free software” means you’re free to do with it what you like; check out the [four freedoms](#).

(By the way, this is important to me: I co-authored a [book](#) about the importance of free software, and I’ll be speaking this year at [LibrePlanet](#), the annual free software conference.)

But who uses Unix, anyway?

But who uses Unix, anyway?

Android **uses the Linux kernel**

Mac OS is a **version of Unix**

Most web servers run **some kind of Unix**

98.8 of the 500 fastest supercomputers **use Linux**

Google developers use “<https://en.wikipedia.org/wiki/Goobuntu>
<https://en.wikipedia.org/wiki/Goobuntu>,” very similar to what
we’re using in class.

And some people (1-2% worldwide) use Linux as their
desktop/laptop operating system

OK, let's start talking about actually *using* Linux.

One *big* and immediately obvious difference between Unix and (say) Windows is the role of the GUI.

Windows (the name tells you) was designed to be fundamentally graphical: most user interaction is with graphic elements (windows, buttons, etc)

Unix, though, was designed and created before we had powerful-enough graphics displays, so the “guts” of Unix are at the *command line*.

Of course, since then, Unix has developed some different GUIs; Mac OS X is one, but more “pure” Unix systems use the **X Window System** (often just “X”)

This is a vastly customizable system with many components. Ubuntu 14.04 ships with the Unity window manager, but there are zillions of others, and each of them can be tweaked to death.

Similarly, Windows does have a “command prompt” but it’s much more **limited in power** than the Unix command line

Bottom line: X makes some things about working with Unix very convenient, but you really have to have basic familiarity with the command line in order to work effectively.

Two Ways to Linux

- 1) With a virtual machine installed on your computer.
- 2) Over the network, to the Linux lab in the WEB.
- 3) (Or maybe you are one of the few people who have Linux as their primary operating system.)

The Virtual Machine

- Full graphical experience, almost exactly like having Linux as your primary operating system.
- Requires a bit of setup (though not as much as installing Linux as primary OS); requires you to have a computer you “control”
- Very forgiving of experimentation—easy just to reinstall if you make a “mistake”
- Probably the best way to get a “proper” Linux experience

Over the Network

- Universally available from anywhere with a network connection
- Original/old-school Unix experience: text/command-line only; NO graphics
- Feels strange, but valuable to be able to work this way

Quick Tour of Virtual Machine

- Note very tight integration—mouse, keyboard, network, filesystem, etc.
- Internet browser
- Settings
- Search
- Software install
- Terminal
- Emacs

Quick Tour of Network Connection

- How to get an account
 - Ask at WEB desk (your account exists already...)
 - Be sure to get two sheets on connection info
- ssh
- Behold the infinite possibility of the command line
- If you haven't yet, you should do the exercises of the first two chapters of the [Introduction to Unix](#).

Software Development in Linux

- Editors
 - vi (or vim)
 - emacs
 - There are others, but...
- Chapter 6 of the [Introduction to Unix](#) gives you a basic overview of both
- They're both very commonly available, and it's worth learning (at least the basics of) how to use both. As the Introduction to Unix says,
 - “In practice most users tend to use both editors - vi to quickly edit short scripts and programs and emacs for more complex jobs that require reference to more than one file simultaneously.

Editor Tutorials

- For vi/vim, this interactive tutorial isn't bad
<http://www.openvim.com/>
- A lot of what you need in vi can fit on a one-page [quick reference sheet](#), though this [quick reference mug](#) solves two problems at once
- This online “[tour](#)” shows you what emacs looks like and can do. Really, the best way to get a sense is to run emacs and start the tutorial: Press `C t r l - h` followed by `t`—in Emacs notation, that's `C - h t`

Compiling and Running

The GNU project produces very good compilers. For C++, the command to run the compiler is `g++`

```
$ g++ hello.cc
```

```
$ a.out
```

```
Hello, world
```

```
$ g++ hello.cc -o hello
```

```
$ hello
```

```
Hello, world
```