Welcome and Intro

Hands-On UNIX System Administration DeCal

Lecture 1 — 23 January 2012

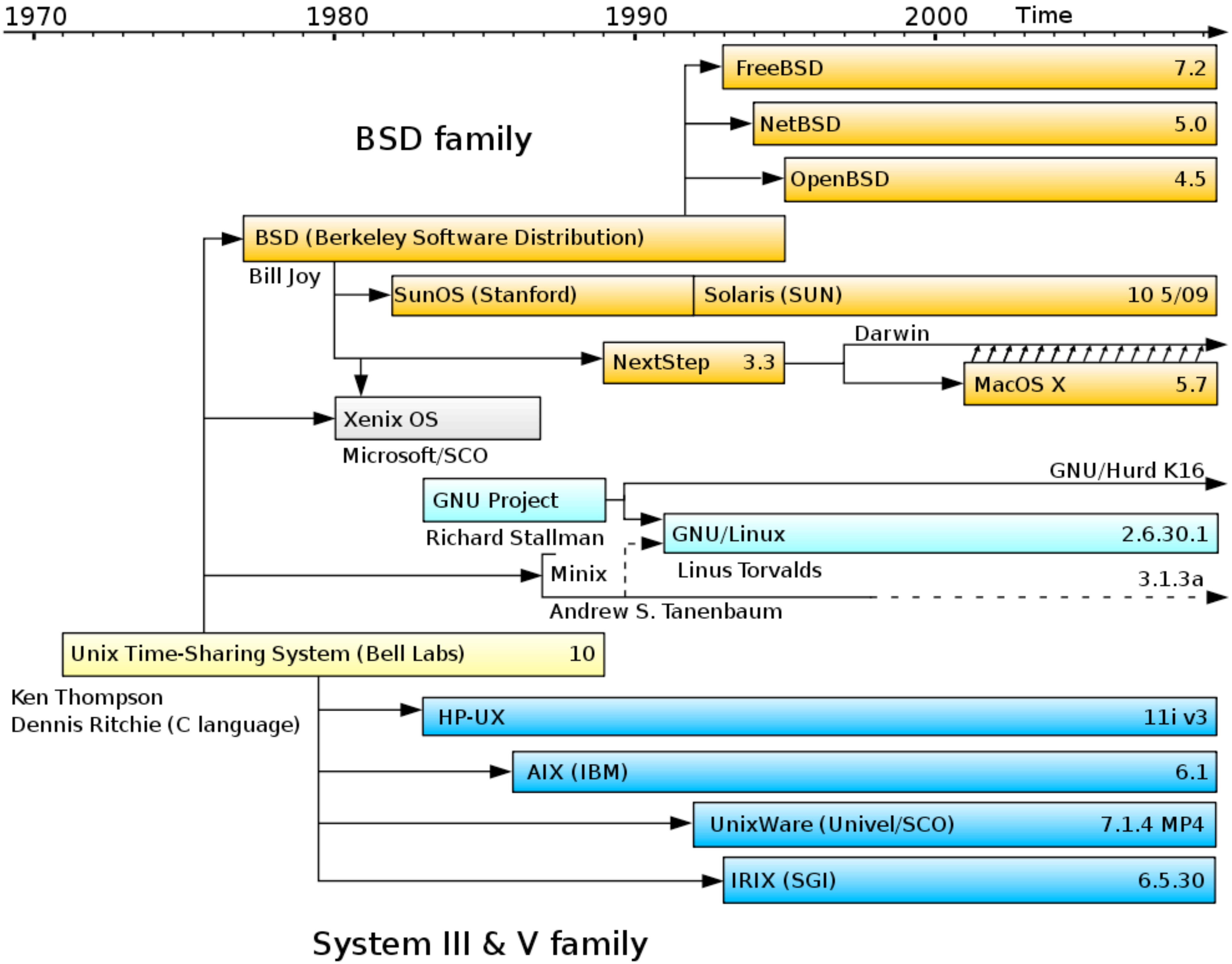
Found the right room?

- You’re in **CS 98/198-8**, the Hands-On UNIX System Administration DeCal.
- No experience with *nix required, but be ready to learn fast! (Don’t be shy — ask questions! We’re also using Piazza.)
- We’re diving in right after this mini intro.

Who are we?

- Jordan Salter — 4th year L&S CS^2 major. Has been seen meddling with OCF and CSUA servers.
- Dara Adib — 2nd year CivE major, City Planning minor. OCF general manager and former site manager.
- Felix Wong — 3rd year EECS major. OCF site manager and ResComp sysadmin.

What’s UNIX?



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

What's sysadminning?

WE TOOK THE HOSTAGES, SECURED THE BUILDING, AND CUT THE COMMUNICATION LINES LIKE YOU SAID.
EXCELLENT.
BUT THEN THIS GUY CLIMBED UP THE VENTILATION DUCTS AND WALKED ACROSS BROKEN GLASS, KILLING ANYONE WE SENT TO STOP HIM.
AND HE RESCUED THE HOSTAGES?
NO, HE IGNORED THEM. HE JUST RECONNECTED THE CABLES WE CUT, MUTTERING SOMETHING ABOUT "UPTIME".
SHIT, WE'RE DEALING WITH A *SYSADMIN*.

source: http://xkcd.com/705

Course logistics

- This is a **lecture/lab based course:** Each class has a lab assignment, due by the start of next week's meeting.
- There's a **final project!** You'll do it in groups of four — more about this later.
- **To earn a "P":** attend every class, attempt each lab (2 drops); do the final project.

Course logistics

- **We're in a bind.**

  This lab has 30 computers. We've already increased the class size to 40, but there are another 20 on the wait-list, a handful more auditing…

Course logistics

- Due to the limited number of computers in this room, **please bring a laptop** to work on if you can.

  (Don’t plan on treating lab assignments as homework — we won’t feel obligated to answer last-minute questions…)

In a nutshell

cd

mv

ls

cat

pwd

find

locate

bash

sed

awk

vi

less

ssh

cp

The shell

- The shell is your gateway to the system. It’s an ordinary program — `sh`, `bash`, `csh`, `ksh`, `tcsh`, `zsh`, `fish`…
- You can access and modify the filesystem, run programs, and administer the system.
- It makes your life easier with automation: shell scripting is a wonderful thing!

Basic commands

- `ls`: list directory contents
- `pwd`: print working directory
- `cd`: change working directory
- `mkdir`: make directory
- `rmdir`: remove directory

Basic commands

- `mv`: move files/directories
- `cp`: copy a file
- `rm`: remove a file
- `less` (`more`): scroll (“page”) through text
- `grep`: search text for a specified pattern

RTFM

- Programs have *command line options*. E.g., `ls -l` gives you more details than `ls`.
- How do you use these commands? There’s a manual — access it like this: `man find`.
- How do you find the right command? Search the manual with `apropos`.

Globbing

- Say you want to move the files “hw01.pdf” through “hw12.pdf” to the “hw” folder. Use a *wildcard*: `mv  hw*.pdf  hw/`
- “*” matches any string; “?” matches only one character (e.g., `lab??.pdf`).
- The Z shell (`zsh`) has souped-up globbing:

```
zmv hw*.pdf HW/*.pdf; ls **/*.pdf
```

The UNIX philosophy

"This is the Unix philosophy. Write programs that do one thing and do it well. Write programs to work together. Write programs to handle text streams, because that is a universal interface."

—Doug McIlroy

(inventor of Unix pipes)

Text streams

- Programs have three *standard streams*: stdin, stdout, stderr. By default, all three are hooked up to your terminal.
- You can *pipe* one command's stdout to another's stdin: `last | grep -c root`
- Or save stdout and stderr to files:

```
make >build-log 2>error-log
```

Text streams

- Feed a file to a program’s stdin:

```
analyze-logs < /var/log/mail.log
```

- Send garbage to the bit-bucket:

```
rm -rf / 2>/dev/null
```

- Append to, instead of overwriting, a file:

```
echo “installed STk” >> admin-log
```

Substitution

- Suppose you want to check the disk usage of all currently-logged-in users. Start by getting a list of their usernames:

```
w | grep pts | cut -f1 -d' '
```

- Then *substitute* the result into the disk quota command. There are two syntaxes for this:

```
quota -sv `w | grep pts | cut -f1 -d' '`
quota -sv $(w | grep pts | cut -f1 -d' ')
```

vi

- That wraps up our quick tour of the shell. You also need to know how to edit text files … introducing Emacs' arch-nemesis!

- vi is unlike anything you've ever seen before. It's a *modal editor*: every key on the keyboard is a command, and to enter text you must enter "insert mode."

- vimtutor is your friend — try it out ASAP.