UNIX™/Linux Overview

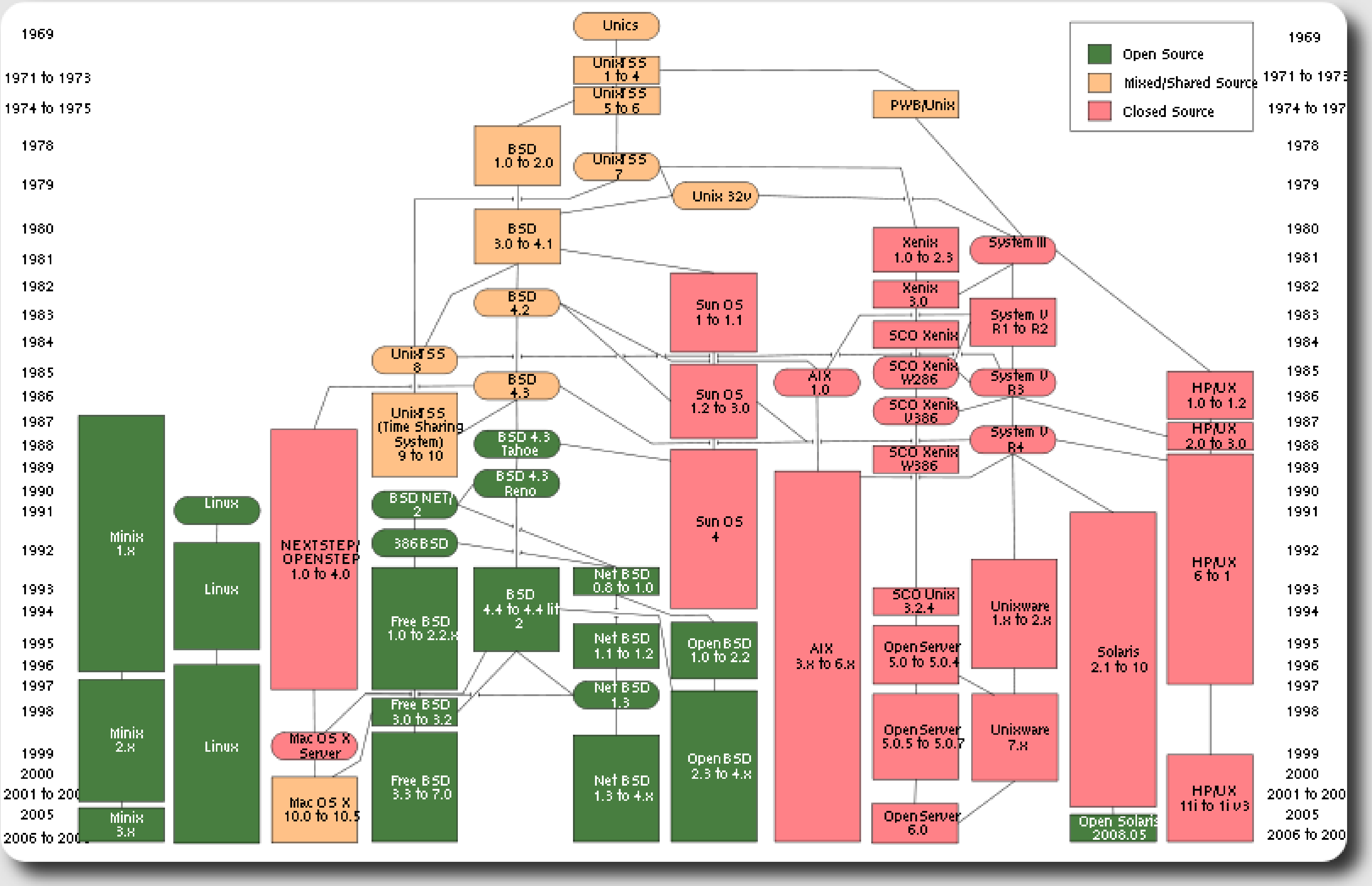
Unix/IP Preparation Course
July 19, 2009
Eugene, Oregon, USA

hervey@nsrc.org

Network Startup Resource Center

UNIVERSITY OF OREGON

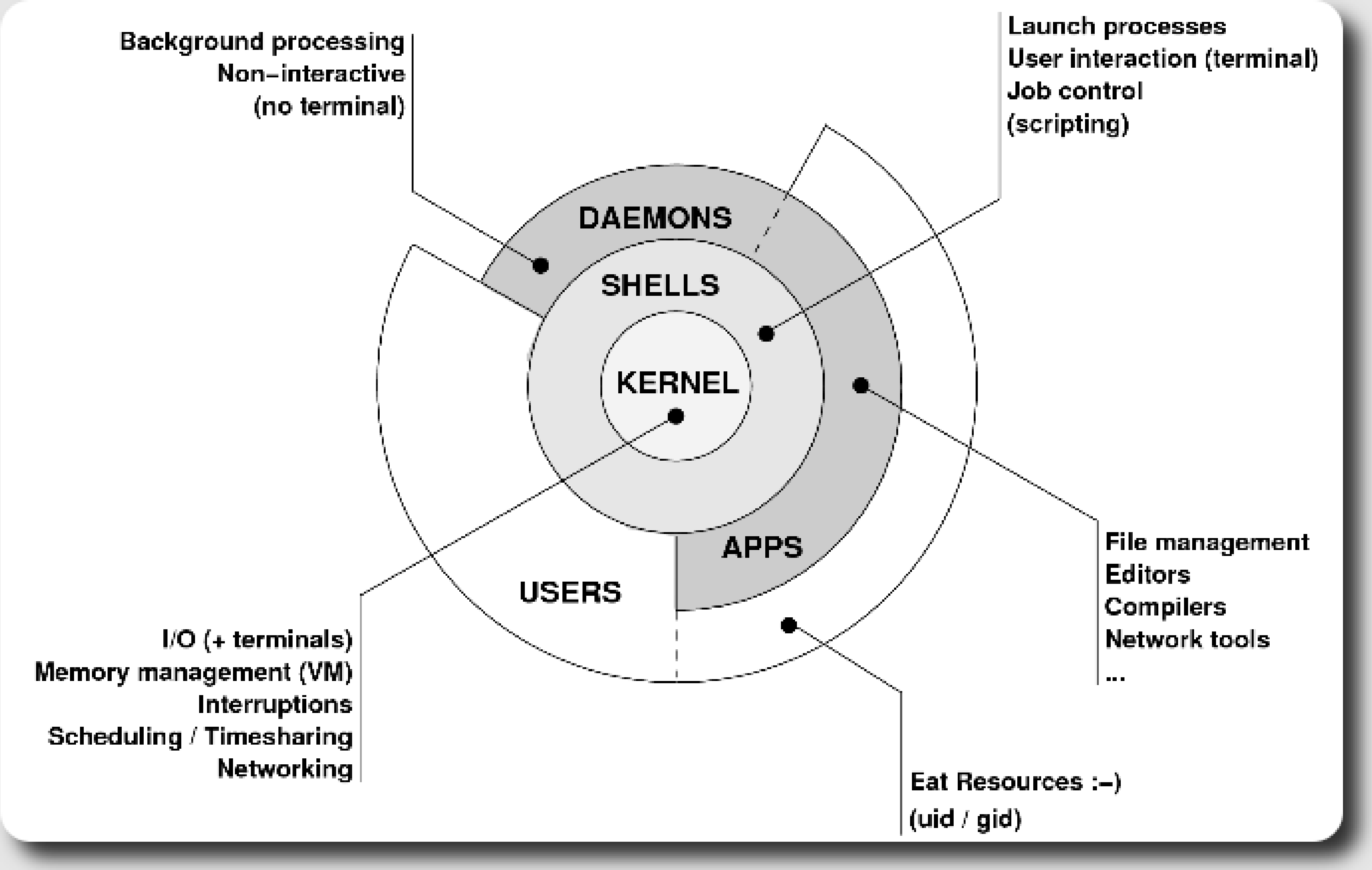
History


1969
1971 to 1973
1974 to 1975
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001 to 200
2005
2006 to 200
Unics
Unix TSS 1 to 4
Unix TSS 5 to 6
PWB/Unix
🟩 Open Source
🟧 Mixed/Shared Source
🟥 Closed Source
BSD 1.0 to 2.0
Unix TSS 7
Unix 32v
BSD 3.0 to 4.1
Xenix 1.0 to 2.3
System III
Sun OS 1 to 1.1
BSD 4.2
Xenix 3.0
System V R1 to R2
SCO Xenix
Unix TSS 8
BSD 4.3
AIX 1.0
SCO Xenix W286
System V R3
HP/UX 1.0 to 1.2
Sun OS 1.2 to 3.0
Unix TSS (Time Sharing System) 9 to 10
SCO Xenix V386
BSD 4.3 Tahoe
System V R4
HP/UX 2.0 to 3.0
SCO Xenix W386
Minix 1.x
NEXTSTEP/ OPENSTEP 1.0 to 4.0
BSD 4.3 Reno
Sun OS 4
Linux
BSD NET/ 2
386BSD
AIX 3.x to 6.x
HP/UX 6 to 1
Solaris 2.1 to 10
Linux
Net BSD 0.8 to 1.0
BSD 4.4 to 4.4 lit 2
SCO Unix 3.2.4
Unixware 1.x to 2.x
Free BSD 1.0 to 2.2.x
Open BSD 1.0 to 2.2
Net BSD 1.1 to 1.2
Open Server 5.0 to 5.0.4
Net BSD 1.3
Free BSD 3.0 to 3.2
Unixware 7.x
Minix 2.x
Linux
Mac OS X Server
Open BSD 2.3 to 4.x
Open Server 5.0.5 to 5.0.7
Free BSD 3.3 to 7.0
Net BSD 1.3 to 4.x
HP/UX 11i to 11i v3
Mac OS X 10.0 to 10.5
Open Server 6.0
Open Solaris 2008.05
Minix 3.x
1969
1971 to 197
1974 to 197
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001 to 200
2005
2006 to 200

Unix vs. Linux

- Are they the same?
  - Yes, at least in terms of operating system interfaces
  - Linux was developed independently from Unix
  - Unix is much older (1969 vs. 1991)
- Scalability and reliability
  - Both scale very well and work well under heavy load
  - (this is an understatement 🙂)
- Flexibility
  - Both emphasize small, interchangeable components
- Manageability
  - Remote logins rather than GUI
  - Scripting is integral
- Security
  - Due to modular design has a reasonable security model
  - Linux and its applications are not without blame

The Unix System



Background processing
Non–interactive
(no terminal)

Launch processes
User interaction (terminal)
Job control
(scripting)

File management
Editors
Compilers
Network tools
...

I/O (+ terminals)
Memory management (VM)
Interruptions
Scheduling / Timesharing
Networking

Eat Resources :–)
(uid / gid)

Kernel

The "core" of the operating system

Device drivers

communicate with your hardware

block devices, character devices, network devices, pseudo devices

Filesystems

organise block devices into files and directories

Memory management

Timeslicing (multitasking)

Networking stacks - esp. TCP/IP

Enforces security model

Shells

Command line interface for executing programs

DOS/Windows equivalent: command.com or command.exe

Also programming languages for scripting

DOS/Windows equivalent: batch files

Choice of similar but slightly different shells

sh: the "Bourne Shell". Standardised in POSIX

csh: the "C Shell". Not standard, but includes command history

bash: the "Bourne-Again Shell". Combines POSIX standard with command history.

Others: ksh, tcsh, zsh

User processes

The programs that you choose to run

Frequently-used programs tend to have short cryptic names

"**`ls`**" = list files

"**`cp`**" = copy file

"**`rm`**" = remove (delete) file

Lots of stuff included in most base systems

editors, compilers, system admin tools

Lots more stuff available to install too

Using the Debian/Ubuntu repositories

System processes

Programs that run in the background; also known as "daemons" ==>

Examples:

cron: executes programs at certain times of day

syslog*d*: takes log messages and writes them to files

inet*d*: accepts incoming TCP/IP connections and starts programs for each one

ssh*d*: accepts incoming logins

sendmail (other MTA daemon like Exim): accepts incoming mail

Security model

Numeric IDs

user id (uid 0 = "*root*", the superuser)

group id

supplementary groups

Mapped to names

/etc/passwd, /etc/group (plain text files)

Suitable security rules enforced

e.g. you cannot kill a process running as a different user, unless you are "*root*"

Any questions?

?

Core directory refresher

/	*(/boot, /bin, /sbin, /etc, maybe /tmp)*
/var	*(Log files, spool, maybe user mail)*
/usr	*(Installed software packages)*
/tmp	*(May reside under "/")*

Don't confuse the the "root account" (/root) with the "root" ("/") partition.

d

'Default' Partition

During an Ubuntu installation you can choose this option. It creates the following:

Root partition

this will contain everything not in another partition

/bin, /sbin, /usr etc.

user home directories under /home

A *swap partition* for virtual memory

/boot for kernel boot files

Partitioning Issues

/var may not be big enough

/usr contains OS utilites, third-party software

/home contains your own important data

If you reinstall from scratch and erase /home, you will lose your own data

Everything in “/” is now more common due to RAID. Why? Valid?

/tmp?

Others?

Note...

Partitioning is just a logical division

If your hard drive dies, most likely *everything* will be lost.

If you want data security, then you need to set up mirroring with a separate drive.

Another reason to keep your data on a separate partition, e.g. /u

Remember, "`rm -rf`" on a mirror works *very* well.

Or, as always "Data Security" <==> Backup

Any questions?

?

What's Different

Software management

dpkg

apt (this is what we'll use)

apt-cache

aptitude

synaptic

meta-packages

repositories

What's Different cont.

Startup scripts

In /etc/init.d/   (System V)

Upon install services run!

Controlling services

update-rc.d

sysvconfig

rcconf

rc-config

What's Different cont.

Make and GCC

- Not installed by default. Why?
- 30,000'ish packages
- To install:

```
apt-get install build-essential
```

What's Different cont.

The use of the *root* account is discouraged and the *sudo* program should be used to access root privileges from your own account instead.

You can do *apt-get dist-upgrade* to move between major and minor releases.

Package sources in /etc/apt/sources.list (how you install from cd/dvd or the network).

Important Reads

man apt-get

man sources.list

Some people like aptitude, partly for the full-screen interface

Meta Packages

Annoying to new users

Provide all packages for subsystems

Initial documentation

`https://help.ubuntu.com/community/MetaPackages`

Examples include:

build-essential (`libc, g++, gcc, make`)

ubuntu-desktop (xorg, gnome)

xserver-xorg-video-intel

There's More

But, hopefully enough to get us started...

Some Resources

www.ubuntu.com

ubuntuforums.org

www.debian.org

ubuntuguide.org

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

http://en.wikipedia.org/wiki/Ubuntu_(Linux_distribution)

GIYF (Google Is Your Friend)

Packages & Exercises

We'll reinforce some of these concepts using exercises...