Introduction to Linux

PacNOG5
June 2009
Papeete, French Polynesia

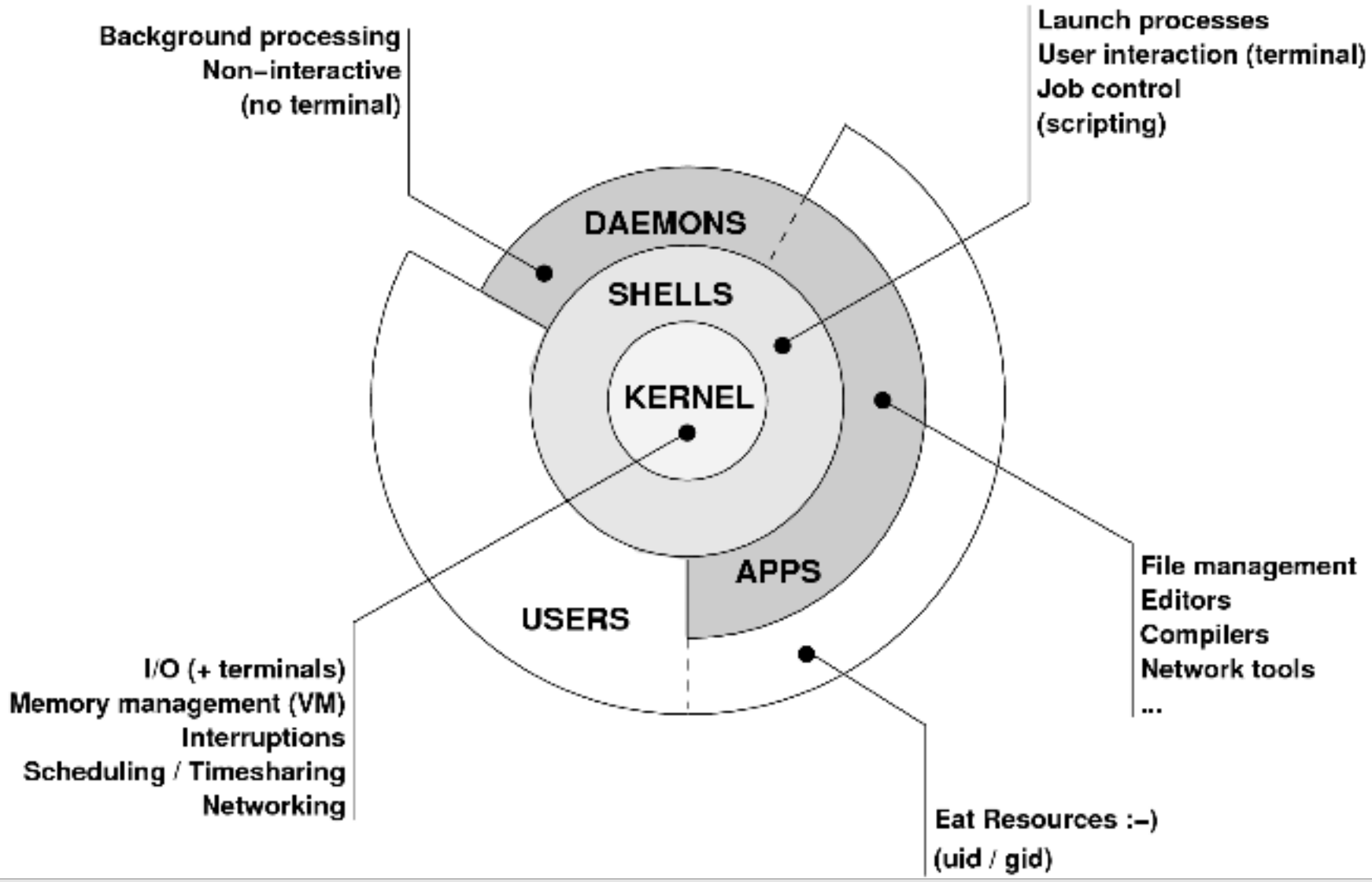
Why use Linux?

- Scalability and reliability
  - has been around for many years
  - works well under heavy load
- Flexibility
  - emphasises small, interchangeable components
- Manageability
  - remote logins rather than GUI
  - scripting
- Security
  - Due to modular design has a reasonable security model
  - Linux and its applications are not blameless though

Initial topics:

- Linux birds-eye overview
- Partitioning
- Ubuntu'isms

The UNIX system



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 (multiprocessing)
- Networking stacks - esp. TCP/IP
- Enforces security model

Shells

- Command line interface for executing programs
  - DOS/Windows equivalent: command.com or command.exe
- 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.

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
  syslogd: takes log messages and writes them to files
  inetd: accepts incoming TCP/IP connections and starts programs for each one
  sshd: 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)
  - /etc/shadow
- Suitable security rules enforced
  - e.g. you cannot kill a process running as a different user, unless you are "*root*"

Any questions?

?

Standard PC boot sequence

1. Power to the Computer.
2. The *Basic Input/Output System* (BIOS) is read from a chip.
3. The BIOS locates a suitable boot source (e.g. hard drive, CD-ROM, network, USB).
4. Disks are divided into 512-byte blocks.
5. The very first block is the *Master Boot Record* (MBR).
6. The BIOS loads and runs the code in the MBR, which continues the bootup sequence.

Partitioning

- The MBR contains a table allowing the disk to be divided into partitions (4 max.).
- Beyond that, you can nominate one partition as an "extended partition" and then further subdivide it into "logical partitions".
- Windows wants to be in the first partition (start of the disk). Linux can boot from most any partition or drive (with modern BIOSes).

Linux partitions

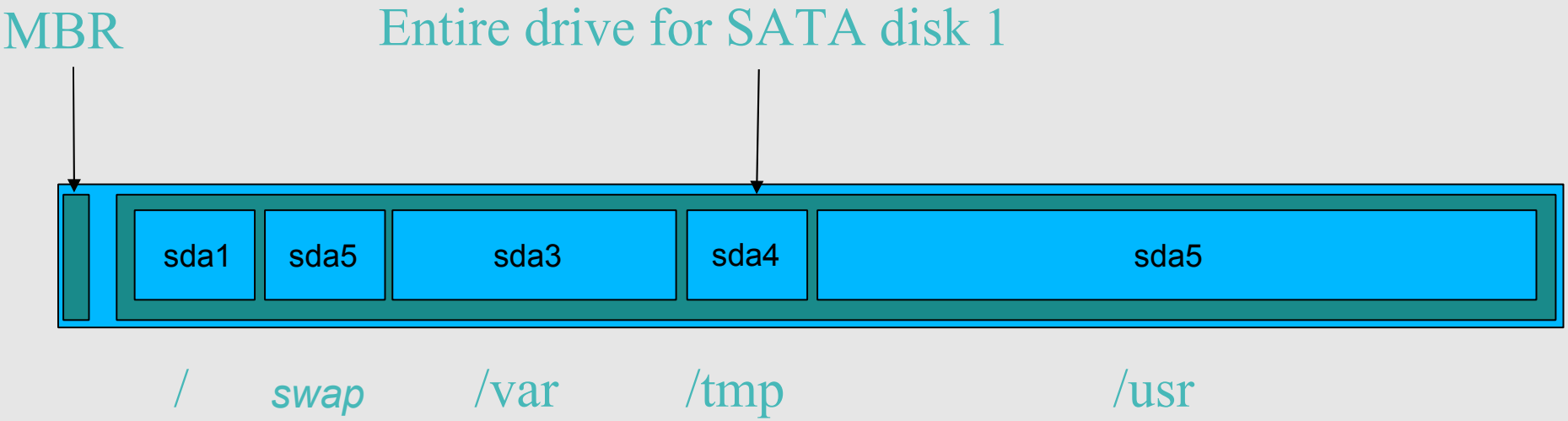
- Partitions referred to by device type, device, partition number - or:
- SATA disk 1 is /dev/sda
- Partition 1 on SATA disk 1 is /dev/sda1
- Partition 3 on SATA disk 2 is...?

  /dev/sdb3

- Partition 1 on IDE disk 2 is...?

  /dev/hdb1

Simple partitioning: /dev/sda (20GB)



/ (root partition)	sda1	512 MB
swap partition	sda5	~ 2 x RAM
/var	sda3	4-8 GB (+)
/tmp	sda4	1-2 GB
/usr	sda5	rest of disk

Core directory refresher

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

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

'Default' Partition

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

1. Small root partition
   - this will contain everything not in another partition
   - /boot for kernel, /bin, /sbin etc.
2. A *swap partition* for virtual memory
3. Rest of disk in “/”
   - Home directories are /home/<username>

Partitioning Issues

- **/var** may not be big enough
- **/usr** contains the OS, 3rd party software, and your own important data
  - If you reinstall from scratch and erase /usr, you will lose your own data
- Everything in “/” is now more common due to RAID. Why? Valid?
- What about /home?
- /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?

?

Ubuntu'isms

- 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?
  - 32,000+ packages
  - To get “apt-get install build-essential”

What's Different cont.

- The use of *root* is discouraged by default and *sudo* is used instead.
- You can do *apt-get dist-upgrade* to move between major and minor releases.
- Sources in /etc/apt/sources.list (how you install from cd/dvd).

Important Reads

man apt-get

man sources.list

Some people like aptitude. That's fine, but watch out for dependency issues!

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...