

Introduction to UNIX



AfNOG X
May 2009
Cairo, Egypt

Boot Camp Summary Time Table

| Topic ----- | Length ----- |
|---|-----------------|
| Morning ----- | |
| Introduction to Unix - Kernel, shell, etc... | 0.5 |
| Introduction to commands - (command [-flags --flags] [params ...]) - practice commands | 0.5 |
| Introduction to the Unix File System/Hierarchy - hierarchy, containers, relative vs. absolute - practice | 0.5 |
| Privileges - User, Group, Other - root user - Octal/Numeric vs. Symbolic mode - Some special cases - Inherited privileges - practice (chmod, chown, etc.) | 0.5 |

Afternoon -----

| | |
|---|---|
| Editing files (configuration files) - Introduction to vi - practice - Virtual terminals - Using the mouse daemon buffer - practice w/ config files (rc.conf, other) - Use multiple terminal windows in GUI w/ cut and paste. | 1 |
| Introduction to TCP/IP - Basic IP/network - Netmask - Default vs static routes - Local network - What is IPv6 - What is SSH and use it to connect | 1 |

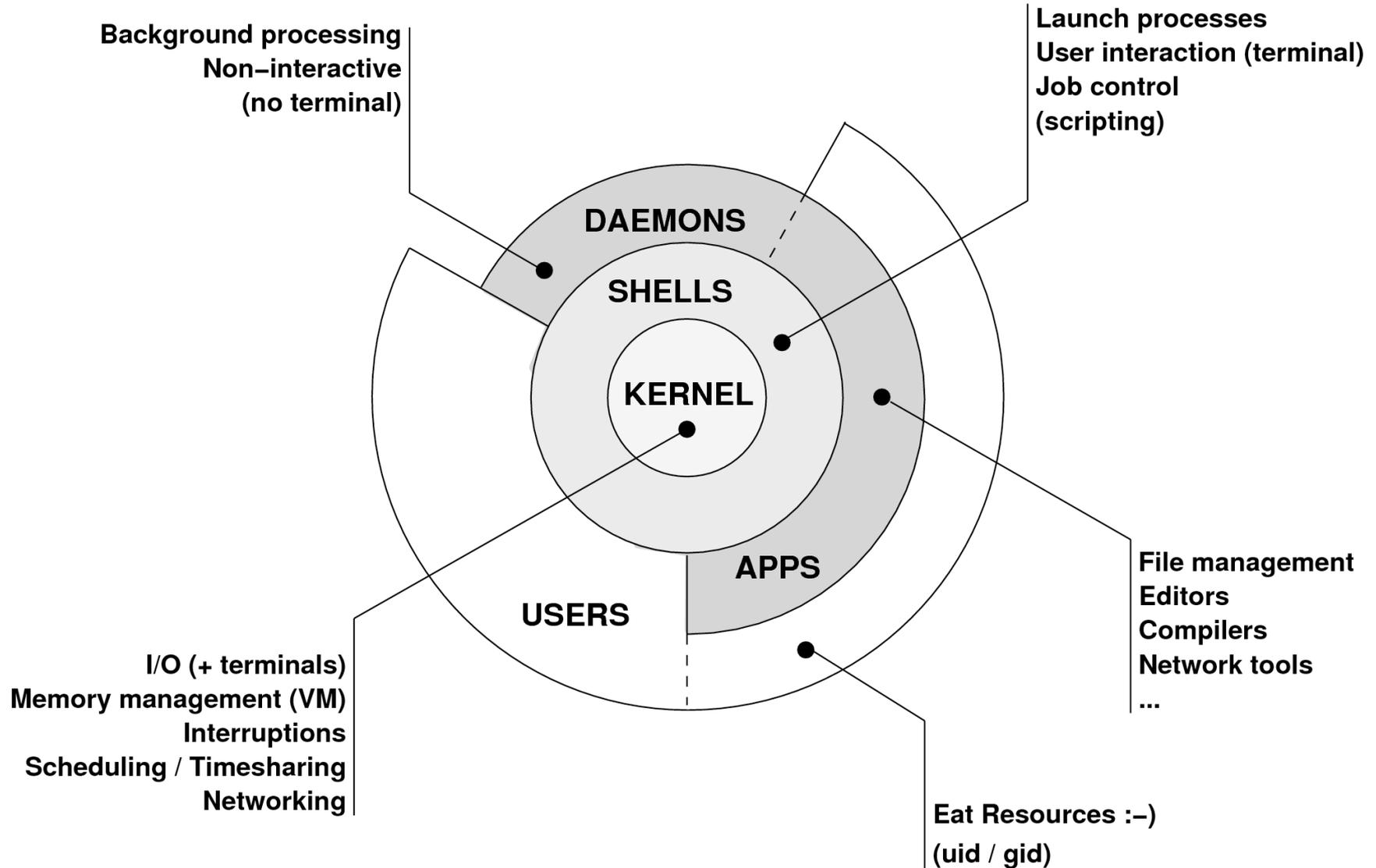
Why use UNIX?

- **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
 - UNIX and its applications are not blameless though

Initial topics:

- UNIX birds-eye overview
- Partitioning
- FreeBSD installation

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

Shell

- 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. But distributed under GPL (more restrictive than BSD licence)

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 the base system
 - editors, compilers, system admin tools
- Lots more stuff available to install too
 - packages / ports

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/pwd.db* (fast indexed database)
- **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. *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".
- FreeBSD has its own partitioning system, because UNIX pre-dates the PC.
- FreeBSD recognises MBR partitions, but calls them "slices" to avoid ambiguity.

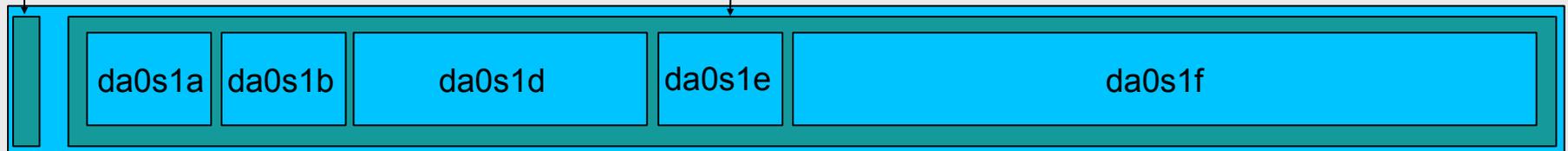
FreeBSD partitions

- Partitions (usually) sit within a slice.
- Partitions called a,b,c,d,e,f,g,h.
- CANNOT use 'c'
 - for historical reasons, partition 'c' refers to the entire slice
- By convention, 'a' is root partition and 'b' is swap partition.
- 'swap' is optional, but used to extend capacity of your system RAM.

Simple partitioning: /dev/da0 (20GB)

MBR

Single slice /dev/da0s1



/ *swap* */var* */tmp* */usr*

| | | | |
|-------------------|-------------------------------|---------------------|---------------------------|
| <code>/</code> | <code>(root partition)</code> | <code>da0s1a</code> | <code>512 MB</code> |
| | <code>swap partition</code> | <code>da0s1b</code> | <code>~ 2 x RAM</code> |
| <code>/var</code> | | <code>da0s1d</code> | <code>4-8 GB (+)</code> |
| <code>/tmp</code> | | <code>da0s1e</code> | <code>1-2 GB</code> |
| <code>/usr</code> | | <code>da0s1f</code> | <code>rest of disk</code> |

'Auto' Partition

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

- Small root partition
 - this will contain everything not in another partition
 - /boot for kernel, /bin, /sbin etc.
- A *swap partition* for virtual memory
- Small /tmp partition
 - so users creating temporary files can't fill up your root partition
- Small /var partition
- Rest of disk is /usr
 - Home directories are /usr/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
- So you might want to split into /usr and /u
 - Suggest 4-6GB for /usr, remainder for /u
- Everything in “/” is now more common due to RAID. Why? Valid?
- Some people use a ramdisk for /tmp

```
# /etc/fstab: 64MB ramdisk
md    /tmp    mfs    -s131072,rw,nosuid,nodev,noatime    0    0
```

Core directory refresher

| | |
|------|---|
| / | <i>(/boot, /bin, /sbin, /etc, maybe /tmp)</i> |
| /var | <i>(Log files, spool, maybe user mail)</i> |
| /usr | <i>(Installed software and home dirs)</i> |
| swap | <i>(Virtual memory)</i> |
| /tmp | <i>(May reside under "/")</i> |

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

Note...

- Slicing/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” \Leftrightarrow Backup

Summary: block devices

- IDE (ATAPI) disk drives
 - /dev/ad0
 - /dev/ad1 ...etc
- SCSI or SCSI-like disks (e.g. USB flash, SATA)
 - /dev/da0
 - /dev/da1 ...etc
- IDE (ATAPI) CD-ROM
 - /dev/acd0 ...etc
- Traditional floppy drive
 - /dev/fd0
- etc.

Summary

- Slices (scsi/sata)
 - /dev/da0s1
 - /dev/da0s2
 - /dev/da0s3
 - /dev/da0s4
- Defined in MBR
- Slices == Partitions in the Windows universe.
- BSD Partitions
 - /dev/da0s1a
 - /dev/da0s1b
 - /dev/da0s1d ...etc
 - /dev/da0s2a
 - /dev/da0s2b
 - /dev/da0s2d ...etc
- Conventions:
 - 'a' is /
 - 'b' is swap
 - 'c' cannot be used

Any questions?

?

Installing FreeBSD

- Surprisingly straightforward
- Boot from CD/DVD, runs "sysinstall"
- Slice your disk
 - Can delete existing slice(s)
 - Create a FreeBSD slice
- Partition
- Choose which parts of FreeBSD distribution you want, or choose "all"
- Install from choice of media
 - CD-ROM, DVD, FTP/PXE boot, etc.

Finding more information

- Our reference handout
- man pages
 - Use this when you know the name of the command
- www.freebsd.org
 - handbook, searchable website / mail archives
- "The Complete FreeBSD" (O'Reilly)
- comp.unix.shell FAQ
 - <http://www.faqs.org/faqs/by-newsgroup/comp/comp.unix.shell.html>
- STFW (Search The Friendly Web), or "GIYF"