Booting and Shutting Down

- **Bootstrapping**
  - The computer must pull itself up
  - Automatic and manual booting
  - **Steps in the boot process**
    - Loading and initialization of the kernel
    - Device detection and configuration
    - Creation of spontaneous system processes
    - Execution of system startup scripts
    - Multi-user operation

- **Kernel initialization**
  - Pathname is vendor dependent
    - Traditional /vmunix
  - Two steps loading
    - ROM loads a small boot program into memory from disk
    - This program then arranges for the kernel to be loaded
  - Memory for internal data structure is allocated

- **Hardware configuration**
  - Check what hardware present
  - Locate and initialize each device as specified
  - Probe the bus for devices and ask the appropriate drivers for information
    - Reboot maybe needed for new devices connected later.
Booting and Shutting Down

- **System processes**
  - **Spontaneous processes**
    - Vary from system to system
    - On system V-ish machines:
      - `sched` – process 0
      - `init` – process 1
      - various memory and kernel process handlers.
    - Not real processes, but part of kernel
  - *Once the spontaneous processes have been created, the kernel's role in bootstrapping is complete.*

- **Single user**
  - Password may or may not be required to enter into root shell
  - The root partition `/` is mounted
    - `/usr` maybe mounted too
  - You can mount other partition manually
  - You can change the read-only mount to be write-mode

- **Execution of startup scripts**
  - Normal shell scripts
  - Process `init` run them according some rules

- **Multi-user operation**
  - `Init` spawns getty processes
  - `Getty` allows user login
Booting and Shutting Down

- Booting in single-user mode
  - Solaris
    - From boot prom (by press stop + a), type
      `boot -s`
    - Other boot commands
      - `boot /path_to_kernel`
      - `boot -s`
      - `boot -r`
      - `boot -a /etc/system.bak`
      - `probe-scsi`
  - HP-UX
    - During prompt
      - Type "boot pri isl"
    - Then type "hpux -iS /stand/vmunix"

Booting PC

- More complex than ...
- Firmware in machine designed for Unix knows
  - How to use the device connected to the machine
  - How to talk to the network on a basic level
  - How to understand disk based file system
- On PC, initial boot code is BIOS - extremely simplistic compare to the firmware
  - Set the boot priority, try cdrom, then disk, ...
  - Load the first 512 bytes of the disk - Master Boot Record (MBR)
  - MBR load a secondary boot program ("boot loader") from a disk partition
    - Lilo (Linux Loader)
    - Grub (Grand Unified Bootloader)
Grub

- Powerful boot loader
  - Is designed to address the complexity of booting a personal computer
  - A wide variety of free OS + proprietary OS

- How to boot OS
  - Loading an OS directly
  - Chain-loading

- Naming convention
  - The first hard disk partition 1: (hd0,1)
  - (hd0,0)/vmlinuz

Loading OS from Grub

- Grub can boot any multiboot-compliant OS in the steps directly:
  - Set root device with command root
  - Load the kernel image with command kernel
    - Append parameters to kernel after the filename
  - Load modules with module
  - Run the command with boot

- Example:
  - title GNU/Linux
  - Kernel (hd1,0) /vmlinuz root=/dev/hdb1
**Grub’s chain-lading**

- Boot unsupported operating system
  - The boot load is in the boot sector of the partition where the OS is installed
  - Set root device with rootnoverify
  - Set the active flag in the partition with makeactive
  - Load the boot loader with chainloader
  - Run boot

- **Example:**
  - Title Windows NT
  - Root (hd0,0)
  - Make active
  - Chainloader +1

---

**Make your system robust**
Network boot

- Situation
  - No local disk
  - Installation

- Net BOOT ROM - PXE ROM
  - Boot the machine
  - Set up the network card (talk with DHCP)
  - Download a second stage boot image - pxegrub
  - Pxegrub will download configuration from tftp server
  - Then download the OS image

Protecting

- Set password to start interactive operation
  - password -md5 $1$............

- For each item
  - Use lock
    - Title Boot DOS
    - Lock
    - Rootnoverfy (hd0,1)
    - Makeactive
    - Chainload +1
  - Use password in the place of lock
Grub’s user interface

- Menu interface
  - Choose preconfigured
  - Edit the configuration
- Command line
  - flexible. - do what ever you need
- Get into the interface
  - Interrupt before the timeout
  - timeout sec

Exercise

- Take a look at your grub configuration file
  - /etc/grub.conf
  - what is the root device
- Get into the grub menu interface
- Get into the grub command line
Startup scripts

- System V style
  - /etc/init.d
  - /etc/rc0.d /etc/rc1.d ...
- Startup scripts perform
  - Setting the name of the computer
  - Setting the time zone
  - Checking the disks with fsck
  - Mounting the system's disk
  - Removing old files from /tmp directory
  - Configuring the network interfaces
  - Starting up daemons and network services

- Run levels:
  - Level 0: system is completely shut down
  - Level 1 or 5: single-user mode
  - Level 2 -5: multiuser levels
    - 2 or 3 are normal
    - 4 or 5 rarely used
  - Level 6: reboot level
- What to do for each level
  - Defined in /etc/inittab
    - Entry format:
      id:runlevels:action:process
    - Default level
    - Typically SAs don't need to mess up with it
**Startup scripts**

- **The startup scripts called by inittab**
  - Individual script master copy stays in init.d, typically under /etc or /sbin
    - Understand argument start, stop, [restart]
  - Symbolic links are made to appropriate directories /etc/rc*.d or /sbin/rc*.d
    - Links start with S or K followed by a number and the name of the service
      - Example, on Fedora under /etc/rc3.d
        - K20nfs -> ../init.d/nfs
        - S12syslog -> ../init.d/syslog

- **When lower run level transitions to higher level:**
  - Init runs all the scripts start with **S**
    - in **ascending** numerical order
    - with argument **start**

- **When higher run level transitions to lower level:**
  - Init runs all the scripts start with **K** (for kill)
    - in **descending** numerical order
    - with argument **stop**

- **To start a daemon, we need to make a symbolic links.**
  - Example:
    - `#ln -s /etc/init.d/sshd /etc/rc2.d/S99sshd`
    - `#ln -s /etc/init.d/sshd /etc/rc2.d/K25sshd`
Startup scripts

- Solaris startup scripts
  - System-V style
    - /etc/init.d and /etc/rc*.d
  - Configuration files
    - /etc/default
      - init
      - Login
      - Console
      - Cron
      - ...
      - /etc/hostname.interface

- HP-UX startup scripts
  - System-V style
    - /sbin/init.d and /sbin/rc*.d
  - Configuration files
    - Under /sbin/rc.config.d
    - Match the master script name
    - Example:
      - /sbin/init.d/HpmOracle
      - /sbin/rc.config.d/HpmOracle
  - Output of startup scripts in /etc/rc.log
    - Excellent place to look for startup errors.
### Startup scripts

- Some commonly modified HP-UX config files in `/etc/rc.config.d`
  - SnmpMaster
  - auditing
  - netconf
  - nameservs
  - nfsconf
  - mailservcs
  - lp

- **Red Hat startup scripts**
  - `/etc/rc.local`
  - interactive confirmation mode
  - `/etc/sysconfig files`
    - Hwconf
    - Network, network-scripts
    - Sendmail
    - syslog

- **FreeBSD**
  - A single startup script `/etc/rc`
    - No concept of level
    - Read three files that specify configuration
      - `/etc/defaults/rc.conf`
      - `/etc/rc.conf`
      - `/etc/rc.conf.local`
  - Scripts `rc.something` will be run in a predefined order.
  - Script `rc.local`
Rebooting and shutting down

- The need to reboot/shutdown
  - Scheduled maintenance, such as OS upgrade
  - Hardware changes or additions
  - Running diagnostics
  - Performance tuning, such as changing static system configuration
  - Administrative tasks, such as testing new startup scripts, etc.
  - System not responding

- Clean system shutdown
  - All users are notified, with some reasonable advance warning.
  - All running processes are sent a signal to exit gracefully, provided the program has made provisions to do so.
  - All subsystems are shutdown gracefully.
  - All remaining users are logged off and remaining processes are killed.
  - File system integrity is maintained.
  - Depending on the type of shutdown, the system moves to single-user mode, the processor is halted, or the system is rebooted...
Rebooting and shutting down

- **Ways to shutdown or reboot**
  - Turn off power
  - Use shutdown command
  - Use halt or reboot commands
  - Sending init a TERM signal
  - Using telinit to change init’s run level
  - Killing init

- **Command shutdown**
  - Safest, most considerate, and most thorough way to
    - Initiate a halt
    - Reboot
    - Return to single-use mode
  - Arguments
    - **System V:**
      ```bash
      shutdown -g n [-i level] [-y]
      -g Wait time, default 60
      level usually
      0: if you intend to turn off the power
      1: administrative state
      2: single-user mode
      5 firmware
      6 reboot to initdefault state
      ```
Rebooting and shutting down

- Example. On Hp-Ux
  `shutdown -h now`
- Read “man shutdown” on your lab machine and try it out.

- Aborting a shutdown
  - During grace period, kill it
  - Otherwise, leave it

- Shutdown security
  - On Hp-UX
    - Only root is allowed by default
    - Configure `/etc/shutdown.allow`

---

Rebooting and shutting down

- When the system crashes
  - Hardware failure
    - Ex: CPU panic
  - Unrecoverable hardware errors
  - Power failures
  - Env problems: too hot.
  - Software problems
  - Resource problems

- What to do
  - Reboot
  - Collect the info
    - Check the lights
    - Check the console
    - Check the crash dump file or other logs if rebooted already

- When the system won’t boot
Rebooting and shutting down

- **Common failures**
  - **Bad or flaky hardware**
    - Check the obvious first
      - Is it turned on?
      - Is cable connection loose?
      - Is the SCSI chain terminated?
    - Try humoring the device
      - Power cycling
      - Call field service
  - **Device failure**
    - Call field service

- **Unreadable filesystems on working disks**
  - Bad rootfilesystem
    - Boot it up using other media (tape or cdrom)
    - Remaking the system restoring its files from backup.
    - The worst case is to reinstall OS and restore files
  - Example: HP-UX has the recovery utility

- **Damage to non-filesystem areas of a disk**
  - Damaged boot areas:
    - Boot block can be restored sometime.
    - Exactly follow the steps that vendor support provides
  - **Corrupted partition tables**
    - Recreate the partition provided you have complete and detailed records of how the partitions were setup.
Rebooting and shutting down

- **Incompatible hardware**
  - New device
    - System does not support the type of device
    - Need system reconfigure
      » Remove it first and then follow the vendor instruction to make the change on your system first.
  - Problems after an upgrade
    - Too old devices
    - New release bugs

- **System configuration**
  - Error in configuration files
  - Unbootable kernel