31338

NETWORK SERVERS

STUDY NOTES

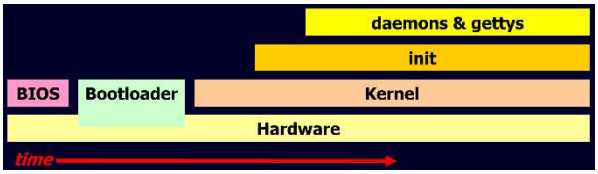
2019

TABLE OF CONTENTS

WEEK 1 – INTRODUCTION AND SYSTEM STARTUP

THE ROLE OF THE SYSTEM ADMINISTRATOR

- Installing & configuring servers
- Installing & configuring application software
- Creating & maintaining user accounts
- Backing up & restoring files
- Monitoring & tuning performance
- Configuring a secure system
- Using tools to monitor security
- Upgrades, security patches
- Fault finding / fixing
- Automating repetitive tasks
- Answering questions
- Negotiating & monitoring service agreements
- Meetings
- Initiating or updating policy documentation
- Purchasing new equipment
- Maintaining awareness of threats & patches



UNIX HISTORY

- Originally developed at Bell Labs
- SVR5: System V Release 5 one of the first commercial versions
- 3 main varients in 2012; IBM, HP-UX and Oracle Solaris
- Linux first release 5th Oct 1991 by Linus Torvalds

BIOS (BASIC INPUT/OUTPUT SYSTEM)

- Located on ROM chip, NVRAM for settings
- Function: Load bootstrap program into memory and start running it
- Allows configuration of the location where the bootstrap is located
- When new card is inserted into PC hardware it requires an IRQ and I/O address, modern devices share IRQ
- IRQ or Interrupt Request – Signal sent to CPU for it to suspend current activity and handle external events
- IRQs are triggered by hardware devices and are numbered 0-15
- Some devices communicate with CPU by storing data/control info in special areas of memory
- DMA (Direct Memory Addressing) – Hardware devices bypass CPU and write directly into system RAM
- RTC (Real-Time Clock) – Battery powered, maintains time when PC powered off, set through BIOS screens
- Linux uses internal clock, syste,m timer initialised from RTC at boot, RTC set from timer value at shutdown
- `date` = shows current linux time. `hwclock` = interfaces with RTC
- Modern BIOS is set to use LBA (Logical Block Addressing) mode to configure hard disk parameters

BOOTLOADER

- Function: Find the Kernel and run it
- May support multiple Kernels
- Located in the MBR (Master Boot Record) first sector of a given hard drive
- Main Bootloaders in Linux PCs:
  - LILO – Linux Loader, discontinued in Dec 2015, multiple kernels and options, change “image=” to kernel filename in /etc/lilo.conf, `lilo` as root installs LILO into the MBR
  - GRUB – Grand Unified Boot Loader, current bootloader v2 2014, configured in /boot/grub/grub.conf, “paragraph” each bootable image, add one and adjust “default=”, written to MBR when installed, unlike LILO, reads config file at boot, does not need to be re-written, `grub-install /dev/hda`
- Windows Bootloader is BOOTMGR, looks for active partition, reads BCD (Boot configuration database) from \boot directory containing config parameters, executes Windows Loader (winload.exe) or (winresume.exe) which loads drivers and transfers control to the kernel

INIT

- First process ever run in a UNIX system PID (Process ID) of 1
- Sys V init processes instructions from /etc/inittab, starts daemons + gettys
  - Daemons – Background processes that perform system tasks
  - Gettys – Processes that monitor a terminal for activity (logins)
- CentOS – Uses "upstart" instead of Sys V
  - Sys V limitations – Synchronous, blocks future tasks until current completed, tasks defined in advance
  - "upstart" developed as event based replacement, was renamed "init"
- Linux has 7 runlevels – Defines the state of the machine after boot, only one level is executed on startup

TRADITIONAL SYS V

- Format: *`id:runlevels:action:process`* EG. 10:3:wait:/etc/rc.d/rc 3
  - *Id* – 1-4 characters which identify an entry in inittab
  - *Runlevels* – Lists the run levels for which the specified action should be taken
  - *Action* – Describes what action should be taken
  - *Process* – Specifies the process to be executed
- Scripts are in /etc/rc.d/rcN.d, where N = runlevel EG. /etc/rc.d/rc3.d contains scripts for runlevel 3
  - Snn scripts are run in order when entering runlevel EG. system startup, changing to a new runlevel
  - Knn scripts are run in order when leaving runlevel EG. shutdown/reboot, changing to a new runlevel
- **Chkconfig** – CLI tool to manage startup scripts, **Ntsysv** – Text-based graphics, friendlier, less flexible
  - Check default runlevel `grep :initdefault: /etc/inittab`
  - Check current runlevel (shows previous and current) `runlevel`
  - Change runlevel `telinit 4` or `init 4`
  - Re-read /etc/inittab file `telinit q`
- Shutdown command
  - `shutdown now`
  - `shutdown –h` #shut down, then halt
  - `shutdown –P` #Power off after shut
  - `shutdown –r now` #Reboot after shut
  - `shutdown +15` #shut in 15 minutes
  - `shutdown –c` #Cancel pending shut
- Other commands: `halt`, `reboot`, `poweroff`

LOG FILES

- System logs help diagnose problems that may occur during booting
- Kernel ring buffer – Kernel and module messages
  - dmesg | less #dmesg: write kernel messages to standard output
  - (after boot, may be in /var/log/dmesg or /var/log/boot.log)
- Main system log file – /var/log/messages Contents configured through syslog daemon – more later
- Other log files in /var/log

WINDOWS STARTUP PROCESS

- The session manager process (SMSS.exe) equivalent to Unix init

WEEK 2 – INSTALLATION AND CONFIGURATION

DOCUMENTATION

- Man pages – View with `man` command – `man 5 passwd` searches for *passwd* in section 5
  - 1. General commands (normal users)
  - 2. System calls (programmers)
  - 3. C library functions (programmers)
  - 4. Special files (usually /dev devices)
  - 5. File formats and conventions
  - 6. Games
  - 7. Miscellanea
  - 8. System admin commands and daemons
- Info documentation – Stored in /usr/share/info – Supports hierarchical structure with hyperlinks
  - Dedicated info reader tool EG. Info ls to view ls documentation
- Package documentation – Installed with software packages – Often text files, HTML, PDF in /usr/share/doc

CONFIGURING THE LINUX MANUAL

- Config file is /etc/man.config
  - MANPATH – Determines search path for manual pages
  - MANSECT – Default order and manual sections to search if not specified
  - Mappings for how to uncompress compressed man page files EG. .gz and .bz2

DOCUMENTATION ON THE INTERNET

- The Linux Documentation Project (www.tldp.org)
  - How To + FAQs – Short/Medium Length
  - Guides – Longer documents (Book-length)
- Program/package web pages and vendor web pages
- Web searches, web forums, Usenet newsgroups and mailing lists
- Windows – Microsoft website and various developers' websites

HARDWARE CONFIGURATION

- ISA Bus (and PnP) – Industry Standard Architecture – Old way of connecting peripherals, required configuring IRQ, IO Address and DMA manually by setting jumpers. Now obsolete
- PCI Bus – Peripheral Component Interconnect – Modern way of connecting peripherals, wider bus (32/64 bits), faster clock rate (33.33Mhz), PnP built in. Linux `lspci` (List PCI devices) `setpci`
- RS-232 Serial Ports – On motherboard, use IRQ and memory address, has UART chip (Universal Asynchronous Receiver Transmitter), 16550A+ do buffering ~115.2Kbps, 16450 ~9.6Kbps, enabled in BIOS
  - Linux `setserial /dev/ttyS0` reports info and configures device
- USB – Universal Serial Bus – USB1.1 12Mbps, USB2.0 480Mbps, Typically EHCI + either UHCI or OCHI drivers
  - UHCI = Universal Host Controller Interface (USB 1.x, Intel/VIA chipset)
  - OHCI = Open Host Controller Interface (USB 1.x, non-Intel/non-VIA)
  - EHCI = Enhanced Host Controller Interface (USB 2.0)
  - Udev used to manage USBs. Runs a daemon and listens (via netlink socket) to uevents the kernel sends if a new device is initialized or is removed from the system. /etc/udev. Previously usbmgr then hotplug
- Sound cards – Managed through BIOS, two sound systems for Linux: OSS = Open Sound System (pre-2.6.x kernels) and ALSA = Advanced Linux Sound Architecture (current) config in /etc/alsa
  - Linux `alsact1` – To control settings for the ALSA soundcard drivers, `alsamixer` – Volume/mixer
- Video Cards
- IDE/ATA/ATAPI Disks – Integrated Drive Electronics – AT Attachment (Packet) Interface – Controller logic built into driver, less work for BIOS. Older PCs have two IDE controllers, supporting 4 devices. Newer PCs have serial ATA (SATA) drives. Typically appear with SCSI device file names /dev/sda, /sdb, /sdc, /sdd
- SCSI Disks – Small Computer System Interface – Separate bus for peripherals, disks and tapes, many varieties SCSI1-Ultrawide, usually needs a SCSI host adapter in system BIOS. Each device has a unique ID, Original 8bit bus, IDs0-7 7=Highest Priority. Newer 16bit bus IDs0-15 priority 7-0, then 15-8
  - Linux cat /proc/scsi/scsi – To list SCSI devices on the system. Disk devices /dev/sda Tapes /dev/st0

UNIX KERNEL AND MODULES

- Current kernels are monolithic but support loadable modules if/when needed
- `Uname -a` – Print system information -a for all
- `Lsmod` – Prints the contents of the /proc/modules file – Shows which kernel modules are currently loaded
- `Modinfo` <modulename> – Shows module info

LOADING AND REMOVING KERNEL MODULES

- `insmod` /lib/modules/2.6.21-1.3228.fc7/kernel/fs/fat/fat.ko – To load a module
  - Requires module filename and doesn't check dependencies
- `modprobe fat` – To add and remove modules from Linux Kernel (Preferred)
  - Requires module name (not filename) and loads any modules needed as dependencies
  - Can use -v -n options to preview what will happen -n: dry-run, does everything but insert or delete
- Removing modules: `rmmod fat` – Will not remove module if currently in use, unless forced

MODULE OPTIONS AND DEPENDENCIES

- Sometimes need to pass options to modules EG. IRQ settings, video card resolution, etc.
- Config files for modules options in directory: /etc/modprobe.d/
- Some modules are dependent on others – Rebuild module dependency database with: `depmod`
  - Updates /lib/modules/‹kernel-version›/modules.dep

CREATING A CUSTOM KERNEL

- Base kernel usually comes from Linux distribution EG. RedHat, Fedora, Debian, SUSE, Slackware, etc
- Compile your own to:
  - Optimise kernel for your hardware config
  - Include support for obscure hardware
  - Support non-standard opt. EG. Huge RAM
- Kernel version numbering
  - Pre-2.6.x: Second number *even* = Stable, *odd* = Development
  - 2.6.x and later: Four digits, no special development numbers

OBTAINING THE KERNEL SOURCE

- Distribution-independent source: www.kernel.org
- Distribution-specific packages: EG. Fedora Core RPM (RedHat Package Manager) files: kernel-2.6.22.1-41.fc7.i686.rpm (pre-compiled) kernel-2.6.22.1-41.fc7.src.rpm (source)
- After download, can verify authenticity using gpg
- Kernel source traditionally in /usr/src/linux

CONFIGURING THE KERNEL BUILD

- Adapt old configuration – Copy .config file from old kernel source dir and run: `make oldconfig`
- Text-mode configuration – `make config` – Starts a character-based questions and answer session
- Text-mode menu configuration – `make menuconfig` – Starts terminal-oriented cursor configuration tool
- GUI configuration – `make xconfig` – Starts a X based configuration tool

COMPILING AND INSTALLING THE KERNEL

- To compile kernel and modules: `make` (modules for 2.6.x onwards) `make modules` (For pre-2.6.x)
- To install kernel: cp /usr/src/linux/arch/i386/boot/vmlinuz/boot/vmlinuz-2.6.22.1-41.fc7
  - vmlinuz may instead be vmlinux or bzImage on different systems
- To install modules: `make modules_install`
- Configure boot loader to boot new kernel EG. edit /boot/grub/grub.conf file and add new kernel section

WINDOWS SERVER 2008

- Multiple versions of Windows Server 2008 exist – Each defined to meet needs of a certain market segment
  - Standard Edition – Small-med business, provide file and print services, internet connectivity and central management of network resources, function as domain controller, member/standalone server