Thoughtful Solutions Ltd
http://very.thoughtful-solutions.info/

Solaris Operator Training

Abstract

Solaris Operator Training

creative commons

COMMONS DEED

Attribution-ShareAlike 2.0 United Kingdom

You are free:

- to copy, distribute, display, and perform the work
- to make derivative works
- to make commercial use of the work

Under the following conditions:

Attribution. You must give the original author credit.

Share Alike. If you alter, transform, or build upon this work, you may distribute the resulting work only under a licence identical to this one.

- For any reuse or distribution, you must make clear to others the licence terms of this work.
- Any of these conditions can be waived if you get permission from the copyright holder.

Your fair use and other rights are in no way affected by the above.

This is a human-readable summary of the Legal Code (the full licence).

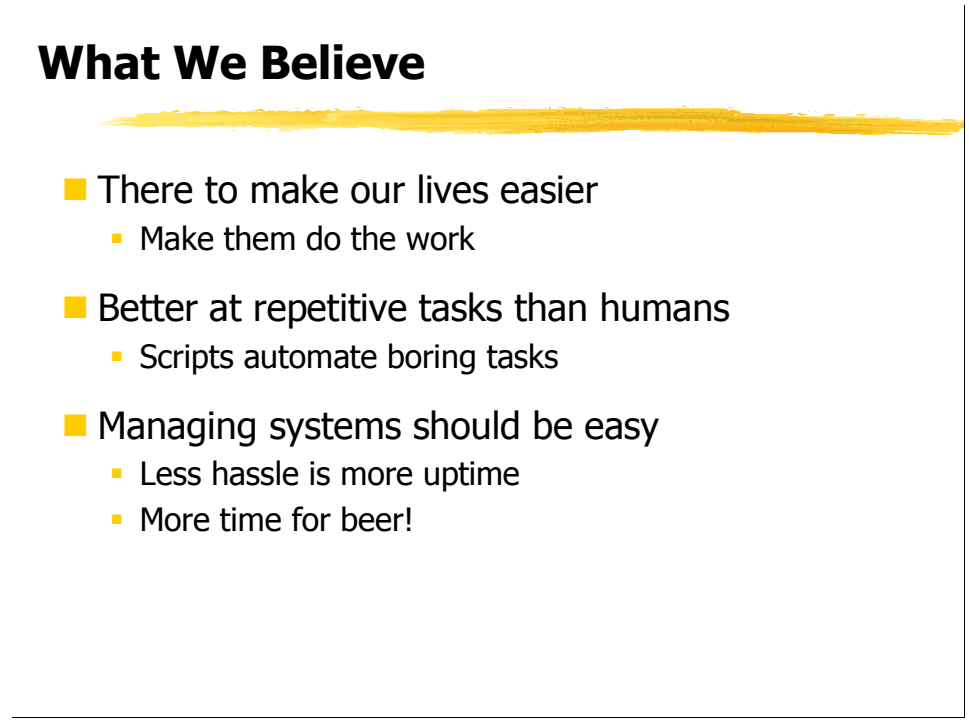
Disclaimer

Systems Administration Philosophy

"We will encourage you to develop the three great virtues of a programmer: *laziness, impatience, and hubris*."

Larry Wall

What We Believe



Laziness
The quality that makes you go to great effort to reduce overall energy expenditure. It makes you write labor-saving programs that other people will find useful, and document what you wrote so you don't have to answer so many questions about it.
Hence, the first great virtue of a programmer or admin
Impatience
The anger you feel when the computer is being lazy.
This makes you write programs that don't just react to your needs, but actually anticipate them. Or at least pretend to.
Hence, the second great virtue of a programmer or admin.
Hubris
Excessive pride, the sort of thing Zeus zaps you for.
Also the quality that makes you write (and maintain) programs that other people won't want to say bad things about.
Hence, the third great virtue of a programmer or admin.

What We Do

- Automation
  - Scripts to install OS
  - Scripts to install applications
  - Scripts to maintain system
- Naming conventions
  - Separation of OS, applications and data
- Centralised management
  - DNS
  - LDAP

Why We Do it

- Consistency! Consistency! Consistency!
  - Hand installed operating systems and applications are never the same twice
  - Scripted builds and application installs will always be the same as the scripts
  - Scripted builds should be self documenting
  - Scripts can parse log files for problems far better than humans

Consistency is the last refuge
of the unimaginative

Oscar Wilde

Automation

- Automation provides repeatability
- Repeatability provides
  - Consistency
  - Faster recovery

One of my primary objects is to form the tools so the tools themselves shall fashion the work and give to every part its just proportion.

Eli Whitney

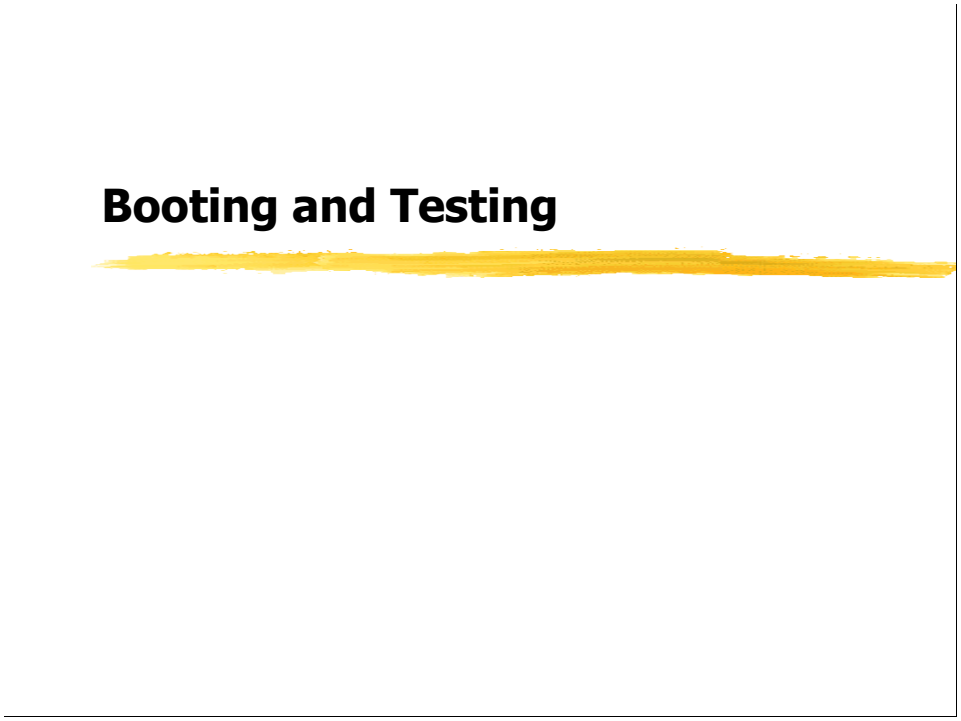
Naming Conventions

- Separate ownership and so, responsibility
  - Leave original copies of OS config files behind
    - Always know base config
    - /etc/hosts, /etc/hosts.Solaris9
  - Installed applications clearly separated from OS
    - Own directory, users, groups etc.
    - e.g. /domain/Oracle/10.1/etc…
  - Versions separated
    - /domain/Oracle/current -> /domain/Oracle/10.1

A place for everything and everything in its place.
Samuel Smiles

Centralised Management

- DNS and LDAP provide centralised management
  - Keep data in one place
  - Make changes in one place
  - Use replication to remove single point of failure
- DNS
  - Multiple machine names
  - Service starting by machine (SRV records)
- LDAP
  - Access controls available everywhere, consistently


Booting and Testing

Open Boot PROM

- Every Sun system has a Boot-PROM
  - Can be accessed prior to starting Solaris
  - **OR** while the OS is running
  - Enter the PROM by pressing `STOP-A` on the keyboard
  - On a serial console, send a long break
- Once you are in the Boot-PROM mode, an OK-Prompt is displayed.
- If Solaris was running, it is "frozen"
  - Type `go` at the `ok` prompt to restart the OS

If the system has LightOut Management, use "#." at the console to enter the LOM, then type "break" to access the OBP.

Testing at the Console

- PROM can be accessed via
  - a directly connected keyboard and monitor
  - a separately connected terminal
- If the system does not detect a keyboard while booting, input and output are automatically redirected to the first serial interface
  - A terminal or another computer is often attached to this port

Accessing the console

- Linux
  - Serial communication applications such as kermit or minicom
- Windows
  - Hyperterminal is a serial communication application bundled with Windows
- Terminal server
  - Terminal servers are pieces of hardware that take the output from serial devices and present them to the network

Rebooting a system

- Prior to rebooting, systems should be shut down in an appropriate manner
- Several alternative commands exist
- In a shell, either locally or remotely, use
  - `shutdown`
  - `reboot`
  - `telinit <runlevel>`
  - `poweroff`

Rebooting the System - cont.

- If a terminal is not accessible, locally or remotely, the machine will need to be rebooted uncleanly.
- If a console is available, use the `sync` command at the `OK` prompt
  - Syncs all file systems, panics the OS and reboots
  - Creates large OS core dump files in `/var/crash/<hostname>`
- Otherwise, the machine must be power cycled
  - For machines with Lights-Out Management, this may be possible remotely

Testing the System

- The following commands are useful tests at the PROM `ok` prompt
  - `banner`
  - `probe`
  - `test`
  - `printenv`
  - `devalias`

Testing the System - banner

- `banner` displays the following information:
  - machine type
  - number, type, and speed of CPUs
  - Boot PROM version
  - quantity of installed memory
  - Serial number
  - MAC-Address and Host-ID

```
ok banner
Sun Netra X1 (UltraSPARC-IIe 400MHz), No Keyboard
OpenBoot 4.0, 640 MB memory installed, Serial #50666290.
Ethernet address 0:3:ba:5:1b:32, Host ID: 83051b32.
```

Testing the System - probe

- The `probe` command shows attached devices on a particular bus
  - `probe-scsi[-all]` Show attached SCSI devices
  - `probe-ide[-all]` Show attached IDE devices

```
ok probe-ide
  Device 0  ( Primary Master )
          ATA Model: ST320413A

  Device 1  ( Primary Slave )
         Not Present

  Device 2  ( Secondary Master )
         Not Present

  Device 3  ( Secondary Slave )
         Not Present
```

Testing the System - test

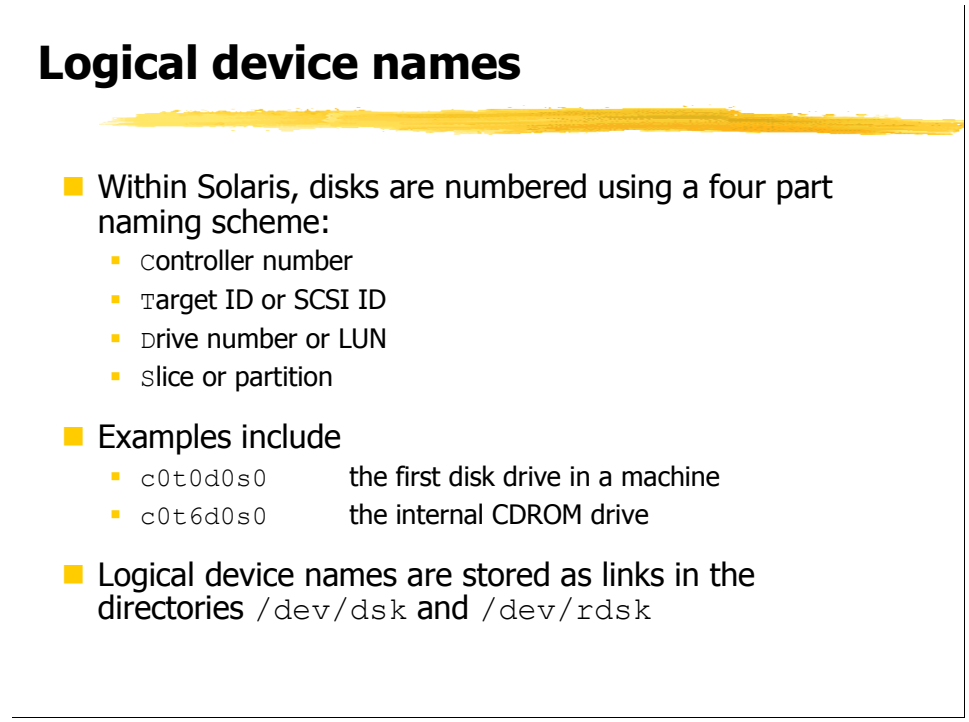
- The test command can be used to run self tests on particular devices
  - `test scsi`
  - `test ide`
  - `test net`
- The `test-all` command will run tests on all devices with a self-test method

Testing the System - printenv

- The Open Boot PROM contains a number of configuration variables that are used to change the way a system behaves
  - e.g. default boot device, serial port configuration
- The `printenv` command displays all available variables and their current and default settings
- `setenv <variable> <parameter>` is used to set a specific configuration variable
- The `set-defaults` command will reset all configuration back to factory defaults

Administering Disks

Logical device names



Disks are both block and character devices. Block for normal access. Character for laying down the initial formatting.

Logical device names - cont.

- New devices are generally not available to the OS until there is a link to them
- The `/dev/dsk` and `/dev/rdsk` directories can be repopulated either
  - while booting (use `boot -r` at the OBP)
  - **OR** during Solaris OS operation with `devfsadm`
- After updating, logical links should exist for all devices

devfsadm

- devfsadm maintains the `/dev` and `/devices` namespaces.
- By default, it attempts to load every driver in the system and attach to all possible device instances
  - i.e. locate all devices attached to the system
- devfsadm then creates device special files in /devices and logical links in /dev.
- devfsadmd is a daemon responsible for handling
  - reconfiguration boot processing
  - updating /dev and /devices in response to dynamic reconfiguration event notifications from the kernel

Partitions

- Partitions are logical divisions within physical disks
  - A single range of contiguous blocks
  - A physical disk may have up to eight partitions
- In Solaris, partitions are also known as slices
  - They are indicated by the letter `s` in the OS device naming scheme
- Partitions may be formatted in a variety of ways
  - Unix UFS
  - Veritas' VxFS
  - A paritition may also be left raw
    - e.g for kernel swap space

Partitions - cont.

- The `format` command is used to create and administer partitions
- A list of partitions can be seen using the `format` command
  - Enter the `partition` menu and use the `print` option
- The default partitioning scheme used by the Solaris 9 install creates three partition
  - `/` the root partition, ca. 1GB
  - `swap` a raw swap partition
  - `/export/home` a partition for user data
- This is not an ideal way of laying out the disk

Partitioning Schemes

- Create a separate `/var` partition
  - Avoids hangs from full root partitions
    - If the root file system is completely filled, an error will occur
    - Errors are written to `/var/adm/messages`
    - If the `/var` directory is on the root partition, the error cannot be written to the log and another error is generated
    - This will continue forever, and the machine will no longer respond to any other requests

Partitioning Schemes - cont.

- Install applications in separate locations to the OS
  - Separation makes it easier to determine where any problems occur
- Separate applications and data
  - Simplifies backups as only variable data needs backing up
  - Store as much as possible with read only permissions
    - Reduces possibility of accidental changes or removal
- Use version numbered directories and links to current version
  - Makes deployment of new versions easier and rollback to old versions faster

Crash Dumps

- Crash dumps are initially written to the swap partition
  - Thus, the swap partition should be at least as big as the physical memory in the machine
- During a subsequent boot, they are read back and compressed into a file in `/var/crash/<hostname>`
  - The `/var` partition should be big enough to deal with crash dump files

Crash Dumps - cont.

- The crash dump directory should be checked regularly and old files removed
  - This is a good task to automate with a script!
- On systems with many megabytes of memory, recovering crash dumps from swap can be very slow
  - A dedicated crash dump partition can be allocated using the `dumpadm` command
  - Recovery from dedicated dump partitions is performed in the background

Removable devices

- Removable devices include CD-ROM, floppy disk and USB pen drives
- Solaris handles mounting these using the Volume Manager
  - Floppy disks are not automatically detected
    - Use `volcheck` to detect a floppy disk when inserted
- Volume Manager may become confused if mounted media is removed without using the `eject` command
  - Restart the volume management service

Removable devices - CD-ROM

- CD-ROMs are automatically detected and mounted by the Volume Manager
  - Once mounted, the hardware eject on the drive is disabled
  - A mounted CD-ROM can be ejected using the `eject` command
  - If a process is using the mounted directory, the eject will fail
    - To find processes using a directory, use

      `fuser -c` ***`path`***

    - To kill processes using a directory, use

      `fuser -k` ***`path`***

Removable devices - CD-ROM cont.

- If Volume Manager is not being used, then as the super user, CD-ROMs must be
  - mounted manually

```
mount -F hsfs /dev/dsk/c0t6d0s0 /tmp/cdrom
```

  - unmounted manually

```
unmount /tmp/cdrom
```

  - ejected explicitly

```
eject /dev/dsk/c0t6d0s0
```

Removable devices- USB Pen Drives

- Most newer Sun systems have USB ports.
- Pen drives can be recognized and mounted by the Volume Manager
  - Create a directory `/rmdisk` with the same ownership and permission attributes as `/cdrom`
  - After a FAT file system memory stick is inserted into the USB slot, its content should be readable as `/rmdisk/<name>`
  - If this is not the case, look at freshly created links in `/dev/dsk/...` pointing to usb paths and try a manual mount:
    - `mkdir /tmp/usbstick`
    - `mount -F pcfs /dev/dsk/c1t0d0s2:c /tmp/usbstick`

Removable devices - Tapes

- Tape drives are not under the control of Volume Manager
- They are handled with commands like `tar`
- Typically, tape drives are connected to the system as external SCSI devices
  - Connection and configuration can be performed while Solaris is running
  - Run `devfsadm` to create links in the `/dev/rmt` directory
  - `0` describes the first tape drive found, `1` describes the second tape drive, etc.

Removable devices - Tapes cont.

- Specific tape drives can be accessed using commands such as

```
tar tvf /dev/rmt/0
```

- Note: Tapes written with SGI systems often can only be read using a different blocksize:

```
tar tvfb 512 /dev/rmt/0
```

- In some rare cases an additional byte swap has to be performed:

```
dd if=/dev/rmt/0 conv=swab | tar tvfb 512 -
```

Cloning a Disk

- It can be useful to clone disks, i.e. produce an exact copy of a disk to be used in another machine
- Hardware of the target should be identical to the source
  - The same hardware architecture must be used
  - The target disk must be the same size or bigger than the source
  - If the target disk is bigger than the source, space will be wasted

Cloning with internal disks

- This method is most appropriate to systems with easily swappable internal disks
  - Insert a target disk
  - Boot to single user mode
  - Configure disks with `devfsadm`
  - Use `format` to find the names of the disks
  - Make sure you understand which is the source and which is the target as reversing the process will delete all data
  - Use `dd` to make a 1:1 copy of the disk
    - `dd if=`*`source`* `of=`*`target`* `bs=128k`
    - where the source and target are `/dev/rdsk/<partition>` paths
  - Check the root filesystem on the target using `fsck`

Cloning with a network

- For machines where the internal disks are harder to get at, it is possible to clone a disk across a network
  - This is a slower method and requires two machines
- On the source machine
  - Add the IP address of the target to `/.rhosts`
- On the target machine
  - Log in at the command line
  - Stop as many processes causing disk I/O as possible
  - Edit the `/etc/inet/hosts` file to include the name and IP addresses of the source and target machines
  - Bring up the network interface
  - Check that you can run commands remotely on the source machine, e.g. `rsh sourcehost ls`

Cloning with a network - cont

- The data is copied using `dd` and `rsh`

```
rsh sourcehost "dd if=/dev/rdsk/sourcedisk \
       ibs=128k obs=8k" \
| dd of=/dev/rdsk/targetdisk ibs=8k obs=128k
```

  - The data is copied from the source disk to STDOUT of the target machine and the block size is changed
  - This is then used as STDIN to `dd` on the target machine, which reconverts the block size and writes it to disk
- To complete the process, reboot the target machine from CDROM
  - Run a filesystem check (`fsck`) on the root partition

RAID

- RAID stands for redundant array of independent (or inexpensive) disks
- The benefits of RAID are
  - increased data integrity
  - fault-tolerance
  - performance

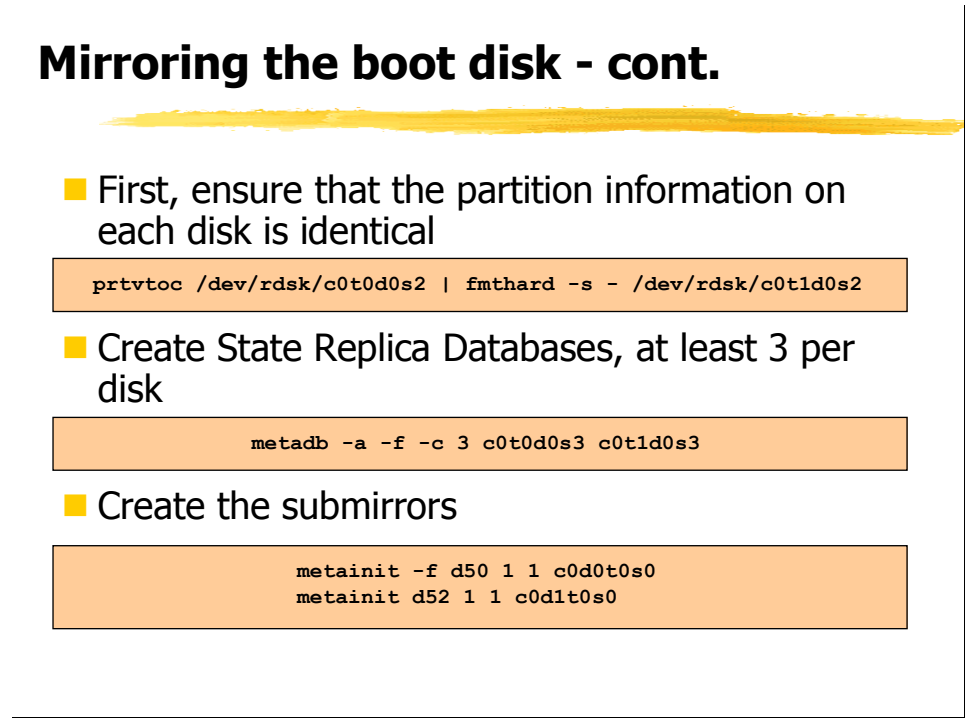
Mirroring and RAID 5

- Mirroring is a form of RAID and is sometimes referred to as RAID 1
  - Mirroring requires two disks and keeps a copy of all data written to the first disk on the second disk
- RAID 5 is another commonly used form
  - RAID 5 requires at least three disks, and stripes the data at the block level over the drives, keeping parity information on all drives
  - RAID 5 is usually used over at least five disks, where one is kept as a "hot spare"

Mirroring the boot disk

- In a machine with two drives, it is possible to mirror the boot disk
  - This provides a failover location from which the machine can be booted should an error occur on the usual boot disk
- To use boot disk mirroring, both disks must be partitioned with
  - a backup partition in slice 2
  - a small partition for metadata

Mirroring the boot disk - cont.



prtvtoc /dev/rdsk/c0t0d0s2 | fmthard -s - /dev/rdsk/c0t1d0s2
prtvtoc reads the virtual table of contents from the first disk
fmthard -s - takes the output from prtvtoc and uses it to create the vtoc on the second disk

metadb -a -f -c 3 c0t0d0s3 c0d1d0s3
metadb created the state replica databases
-a updates /kernel/drv/md.conf and /etc/lvm/mddb.cf with name of new meta device
-c 3 creates three replica files
-f forces creation of new state replica databases
then the slices to be used

metainit -f d50 1 1 c0d0t0s0
metainit d52 1 1 c0d1t0s0
metainit configures meta devices
-f used to force command to act on mounted filesystems or swap partions
d## meta device name (device is /dev/md/dsk/d##)
1 1 specifies that the meta device is a simple concatenation using 1 stripe and 1 partition

Mirroring the boot disk - cont.

- Initialise the first half of the mirror

```
metainit d54 -m d50
```

- Enable booting from either mirror

```
metaroot d54
```

- Reboot the system
- Attach the second disk to the mirror
  - This will cause the mirror to sync from the first disk to the second

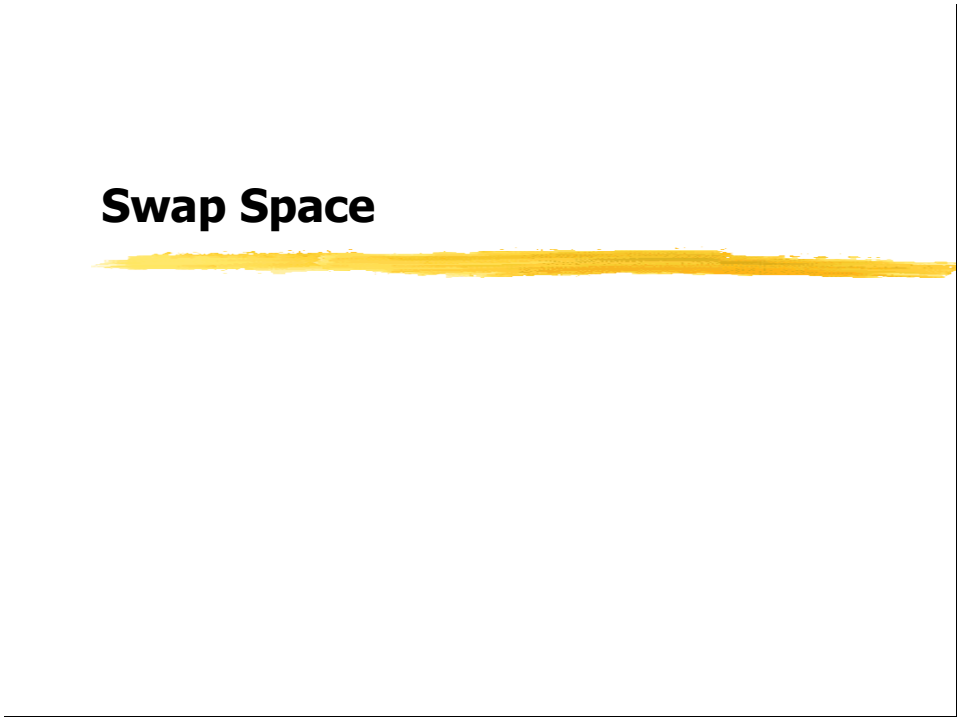
```
metattach d54 d52
```

metainit d54 -m d50
This creates a one way mirror of the meta device d50 as meta device d54

metaroot d54
Modifies `/etc/vfstab` and `/etc/system` to allow booting from either disk in the mirror

Mirroring the boot disk - cont.

- To avoid error messages during emergency boot, repeat the process to mirror the swap slice
- It should now be possible to boot the system from either disk at the OBP


Swap Space

Defining Swap Space

- A swap file system can use
  - a raw partition like `/dev/dsk/c0t0d0s1`
  - plain files like `/usr/local/swapfile.1`
- Multiple swap file systems can be defined
  - These are logically combined by mounting them to the `/tmp` mount point
  - This means as long as physical memory is available, writing files to `/tmp` will write to and use up memory

Permanent Configuration

- Swap space can be permanently defined in `/etc/vfstab`

```
trinity $ cat /etc/vfstab
#device         device          mount           FS      fsck    mount   mount
#to mount       to fsck         point           type    pass    at boot options
#
fd      -       /dev/fd fd      -       no      -
/proc   -       /proc   proc    -       no      -
/dev/dsk/c0t0d0s0       /dev/rdsk/c0t0d0s0      /       ufs     1       no      -
/dev/dsk/c0t0d0s1       -       -       swap    -       no      -
/usr/local/swapfile.1   -       -       swap    -       no      -
swap    -       /tmp    tmpfs   -       yes     -
```

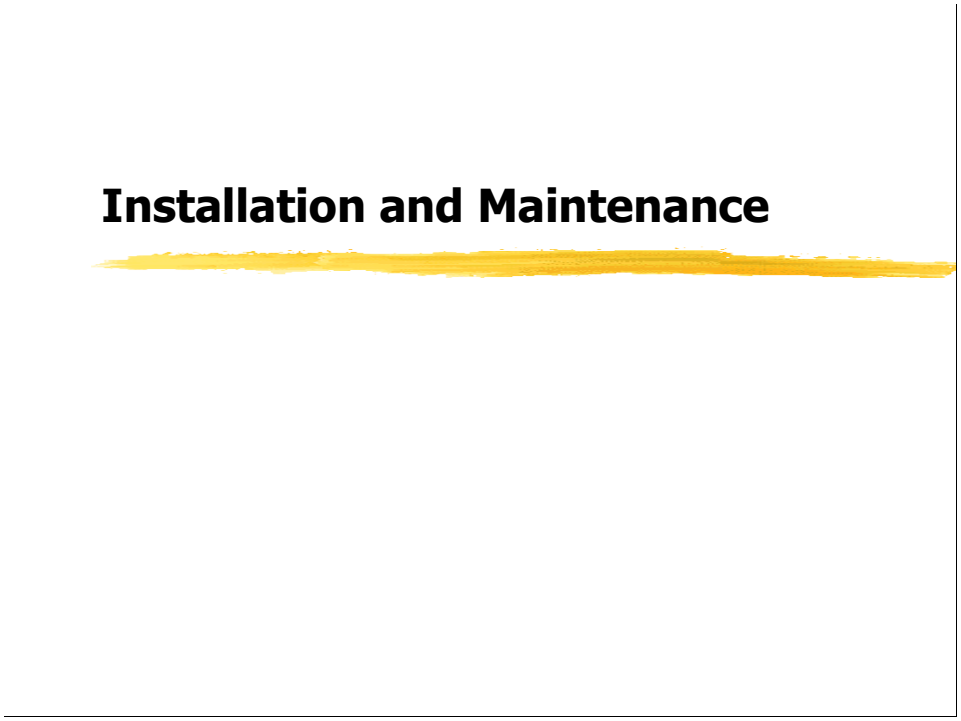
Temporary Configuration

- Plain files can be used to add swap space during standard Solaris OS operations:

```
trinity# mkfile 100m /usr/local/swapfile.1
trinity# swap -a /usr/local/swapfile.1
trinity# swap -l
```

- This addition is temporary, and will be removed at the next reboot
- To remove temporary swap space without a reboot, use

```
trinity# swap -d /usr/local/swapfile.1
```


Installation and Maintenance

Installation From CD

- Solaris can be installed interactively from CDROM
- Advantages
  - Often requires no advance setup
- Disadvantages
  - This method is slow and labour intensive
  - During installation, it is necessary to enter details about the system
  - Every machine installed this way is likely to be slightly different
  - Not all Sun machines ship with CDROM drives
  - CDROM drives are slow

Installation from CD - cont

- Insert the CD in the drive, and, at the OBP, type `boot cdrom`
- Follow the on screen instructions
- After an installation, root can only log in at the system console
  - To enable remote login via telnet
    - edit `/etc/default/login`
      Comment out the line
      `CONSOLE=/dev/console`
  - To enable remote login via ssh
    - edit `/etc/ssh/sshd_config`
    - Change the line
      `PermitRootLogin=no`
      to `PermitRootLogin=yes`

Installation from CD - cont.

- The following may cause issues
  - Name service
    - If name services have not yet been set up on the network, choose NONE to avoid hangs
  - Installation cluster
    - Choosing the "Entire Plus OEM Software" cluster ensures everything is installed
    - For dedicated servers, unnecessary packages may be removed afterwards to minimize security risks.
  - Disk partitioning
    - The default installation partitioning system does not include a separate `/var` partition
    - Use "custom," not "default," when prompted for an installation method to change this

Network Installation

- Solaris can also be installed across a network
- Advantages
  - Faster
  - Installation can be automated
  - Post-installation configuration can be scripted
  - One machine can be easily cloned to many others using Flash archives
- Disadvantages
  - Requires working network infrastructure
  - Requires a Jumpstart server to be configured

Creating a Jumpstart server

- Jumpstart installations can be performed from any system that provides NFS and DHCP
  - Server need not be the same architecture as the target platform
- Mount the installation CD and run the install script from the Tools directory

```
trinity$ cd /cdrom/cdrom0/s0/Solaris9/Tools
trinity$ setup_install_server -b /export/install/boot
```

- Once the files are copied from the CD, export the directory over NFS

```
trinity$ share -F nfs -o ro,anon=0 /export/install
```

Adding Jumpstart Clients

- Add clients using `add_install_client`
  - By default, the client script uses R/ARP to assign network addresses

```
trintiy$ cd /export/install/boot/Solaris8/Tools
trinity$ add_install_client -d -c trinity:/export/install/admin \
-s trinity:/export/install/boot -p trinity:/export/install/admin \
morpheus sun4u
```

  - `-d` specify as a DHCP client
  - `-s` location of Solaris boot image
  - `-c` location of `sysidcfg` file
  - `-p` location of `rules.ok`

Automating Jumpstart

- To fully automate a Jumpstart, three files are required
  - `sysidcfg`
  - `rules.ok`
  - `profile`
- Two other files may be referenced
  - `begin` script
  - `finish` script

Automating Jumpstart: sysidcfg

- The `sysidcfg` file identifies the system
  - Using DHCP removes the need for each machine to have a separate file

```
name_service=DNS {
        domain_name=matrix.com
        name_server=10.10.0.100
        search=matrix.com }
network_interface=PRIMARY {
        dhcp
        protocol_ipv6=no }
root_password=zoJMzm9KSEWF6
security_policy=NONE
system_locale=en_GB
terminal=xterm
timezone=GB
timeserver=localhost
```

Automating Jumpstart: rules

- The `rules` file is used to allow different configurations to be given to different machines based on a number of simple tests e.g. architecture, hostname, network
- Simplest `rules` file has one configuration for all machines

```
any - - Profiles/basic.profile -
```

- During installation, the `rules.ok` file is used
  - This is generated from the rules file using the `check` script
    - checks the syntax of both the rules file and the profile
    - strips white space and comments
    - produces checksummed rules.ok file

Automating Jumpstart: profiles

- A profile is a text file that defines how to install Solaris on a system
- The profile defines elements of the installation, for example, the software group to install
- Profiles consists of profile keywords and values
- Profiles must contain the following:
  - The `install_type` profile keyword as the first entry
  - One keyword per line

Automating Jumpstart: profiles - cont.

```
install_type      initial_install
system_type       standalone
cluster           SUNWCXall
partitioning      explicit
filesys           rootdisk.s0 2048  /
filesys           rootdisk.s1 2048  /var
filesys           rootdisk.s3 1024  swap
filesys           rootdisk.s7 free  /export/home
locale            en_GB.ISO8859-1
```

Automating Jumpstarts: finish script

- Finish scripts are user-defined Bourne shell scripts
- Perform tasks after the Solaris is installed on a system, but before the system reboots
- Useful for adding packages, and doing configuration tasks
  - e.g. allowing ssh access for root, configuring CDE access via vnc and inetd

Jumpstart: network infrastructure

- Physically, this need only be a crossover cable
- The following software is required
  - DHCP server provides IP addresses to clients
  - NFS server serves installation data over network
  - TFTP server provides initial kernel
- The following can also be useful
  - DNS server provides name resolution

Faster Jumpstarts - flash archives

- An alternative method of Jumpstarting installs the operating system from a Flash archive, also known as a flar
- A flar is a compressed archive containing a copy of a Solaris installation

Creating flash archives

- Flash archives are created from a master host that has been installed and configured as required
  - Install Solaris on the master host
  - Add patches and install software packages
  - Configure the system as required
  - Create a flash archive

```
trinity$ cd /data
trinity$ flar create -c -n b100s -x /data ./b100s.flar
```

    - The archive b100s.flar is compressed and named b100s
    - -x excludes the /data directory
  - Copy the flar file to the jumpstart server
  - Create a profile to use the flash archive

Jumpstart: Installation

- Once the Jumpstart server has been configured, all other clients can be installed from it
- At the OBP, boot the machine from the network using dhcp
  - Use the `install` flag to show this is an installation

```
ok boot net:dhcp - install
```

Packages

- Sun and its third-party vendors deliver products in a form called a software package
- Packages are subdirectories or tar/compress/zip compressed subdirectories

Packages: Naming

- Packages are named as follows:
  - The first four uppercase letters correspond to the U.S. stock ticker symbol (or a fantasy symbol) of the origin company, e.g. SUNW, SFW
  - The following lowercase letters describe the function of the package
    - SUNWbash is the package containing the Bourne again shell
    - SFWvnc is the package containing VNC freeware

Packages: Managing

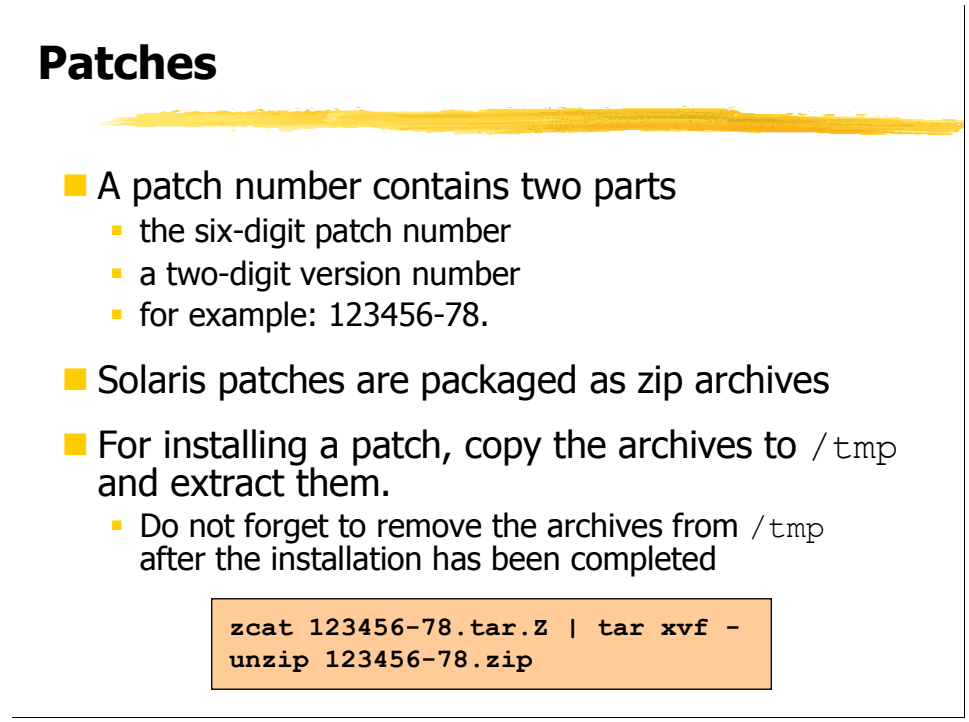
- Package are installed using `pkgadd` with the directory storing the package contents:
  `pkgadd -d /cdrom/cdrom0`
- `pkginfo` prints information on installed packages
- `pkgrm <Package-Name>` removes a package.
- `pkgchk -v <Package-name>` lists all files and directories associated to the package given.

Packages: Managing - cont.

- To find out which files belong to which installed packages, look in `/var/sadm/install/contents`
- To find out which files are present in packages on a Solaris CD-ROM, for example, look into the pkgmap files

```
grep file /cdrom/cdrom0/s0/Solaris_9/Product/*/pkgmap
```

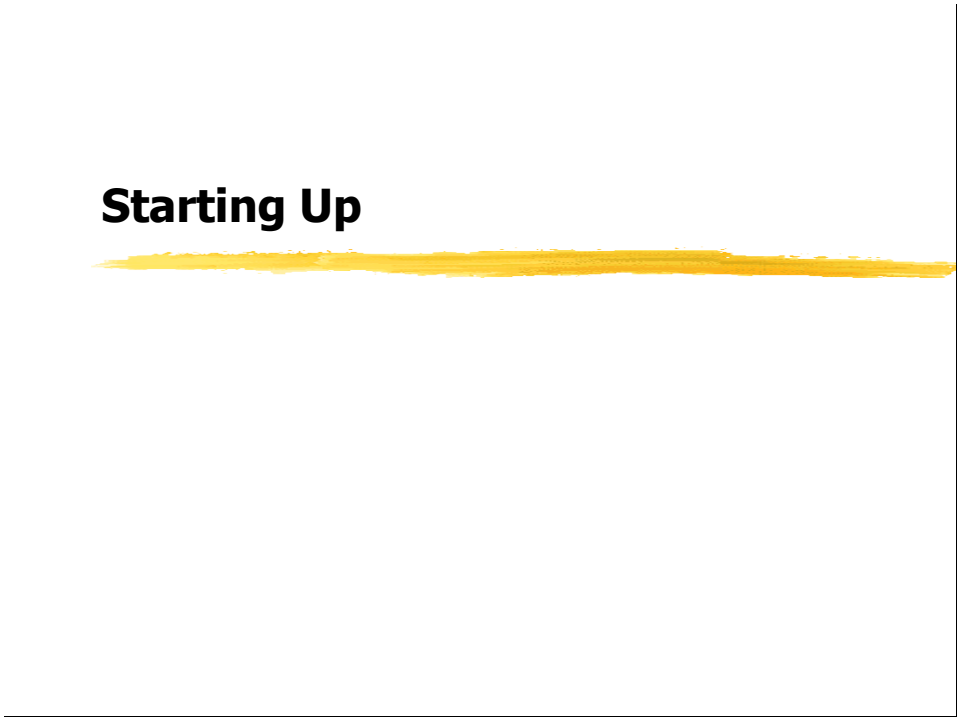
Patches



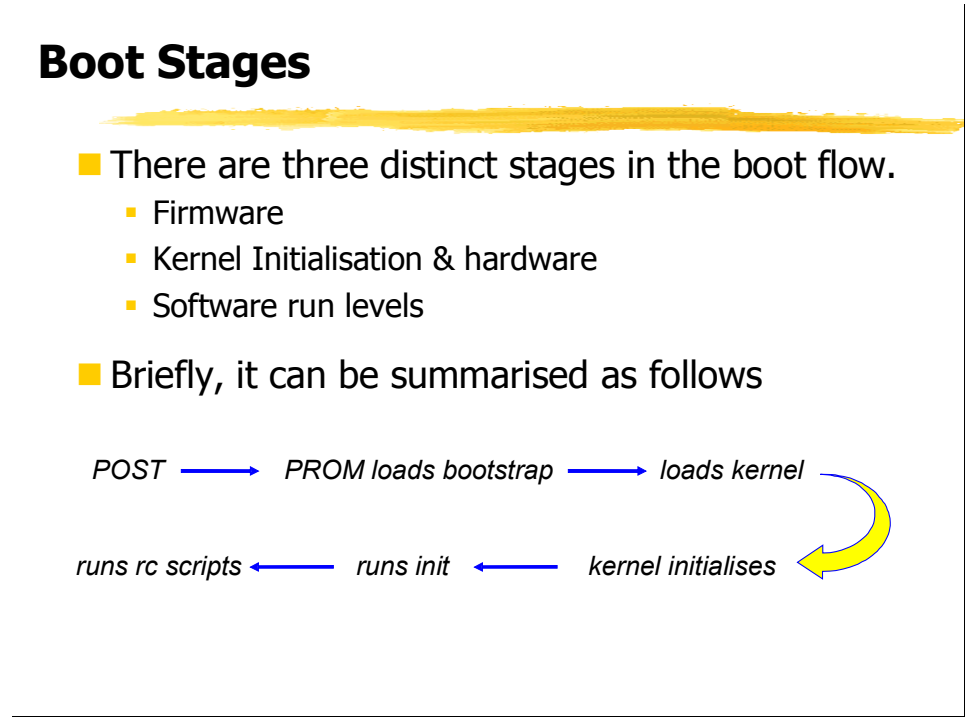
Starting with the Solaris 7 OS, the Installation CD comes with a pre-patched Solaris OS image. To upgrade previous quarterly updates to this image, use the "Maintenance Update.“

For example: Installing the "Maintenance Update 7" (which comes with the Solaris OS 8 02/02 version) onto the Solaris OS 8 07/01 version, will upgrade the Solaris OS patches (and only those -- that is, there are no functionality enhancements) to the status found on the 02/02 image. The upgrade installation writes a mark into the /etc/release file: "Solaris 8 07/01 Maintenance Update 7.“

The "Recommended Patch Cluster" is a different mechanism, since you can have a different combination of patches and patch versions in the "Recommended Patch Cluster" every month. This cluster can be downloaded from the BigAdmin Portal (http://www.sun.com/bigadmin) and should be installed after every Solaris OS installation.

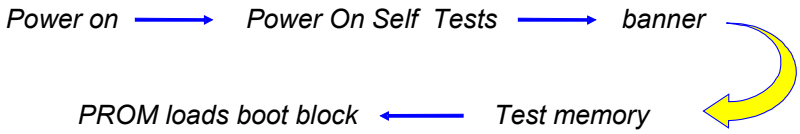

Starting Up

Boot Stages



The power on self tests are used when a system is powered on or reset from the PROM level. They check out the CPU and the MLB and the tests can vary from system to system. By default the POST output is not displayed to the monitor, however it can be viewed by connecting an ASCII terminal to the serial port.
User commands are used to control the operation of the PROM. They can change PROM variables such as the boot device and security modes.
The PROM contains an entire programming language and operating system known as forth. Developed in 1978, this high-level language can fit into an 8KByte chip.

Firmware



- The output from the POST can be seen on an ASCII terminal attached to Serial Port A of the machine

POST output

POST output

```
ok setenv diag-switch? true
ok setenv auto-boot? false
ok reset
Res
LOM event: +3d+23h48m54s host reset
etting ...


Processor Speed = [Speed Jumper = 7] 400 MHz
Baud rate is 9600
8 Data bits, 1 stop bits, no parity (configured from lom)

Firmware CORE  Sun Microsystems, Inc.
@(#) core 1.0.18 2002/05/23 18:22
Software Power ON
Verifying NVRAM...Done
Bootmode is 0
MCR0 = 16a0b004
MCR1 = c0804000
MCR2 = fc0af00
MCR3 = 313
```

The tests carried out by POST are simple yes/no type tests. It checks the integrity of the CPU, hardware and any devices it can probe.

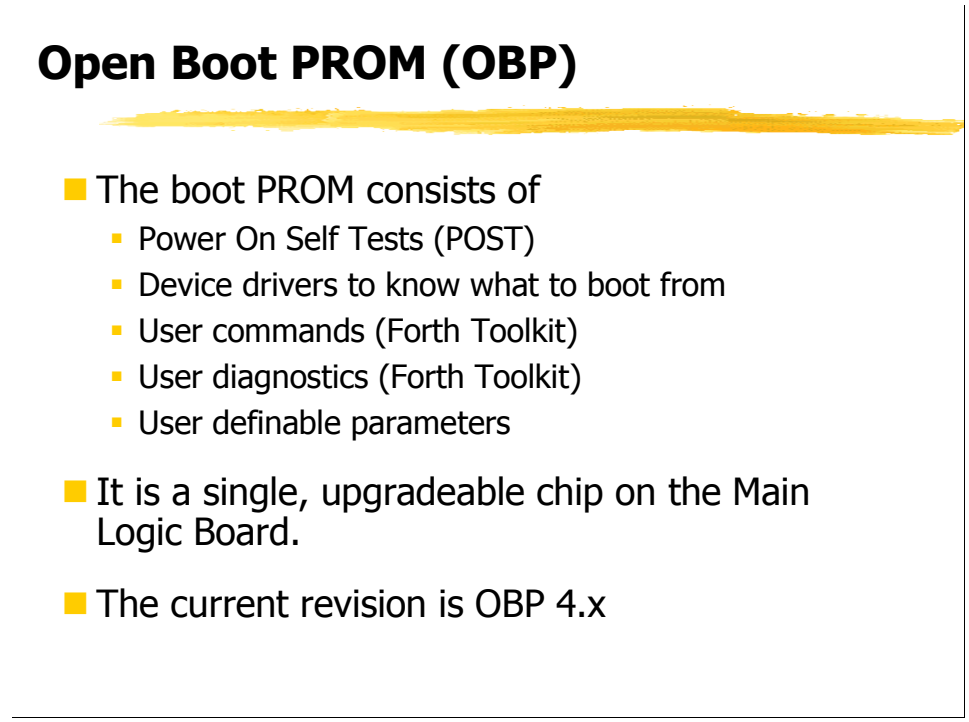
POST output

```
Ecache Size = 256 KB
Clearing E$ Tags Done
Clearing I/D TLBs Done
Probing memory
Done
MEMBASE=0x60000000
MEMSIZE=0x8000000
Clearing memory...Done
Turning ON MMUs Done
Copy ROM to RAM (154720 bytes) Done
Orig PC=0x1fff0007edc  New PC=0xf0f07f34
Processor Speed=400MHz
Looking for Dropin FVM ... found
Decompressing Client Done
Transferring control to Client...
Reset Control: BXIR:0 BPOR:0 SXIR:0 SPOR:1 POR:0
Probing upa at 1f,0 pci
Probing upa at 0,0 SUNW,UltraSPARC-IIe (256 Kb)
Loading Support Packages: kbd-translator
Loading onboard drivers:
Probing /pci@1f,0 Device 7  isa dma rtc power SUNW,lomh serial serial
   flashprom
```

POST output

```
Probing /pci@1f,0 Device 3  pmu i2c temperature dimm dimm i2c-nvram
   idprom motherboard-fru ppm beep fan-control
lomp
Probing Memory Bank #0   0 Megabytes
Probing Memory Bank #1   0 Megabytes
Probing Memory Bank #2 512 Megabytes
Probing Memory Bank #3 128 Megabytes
Probing /pci@1f,0 Device 7
Probing /pci@1f,0 Device 3
Probing /pci@1f,0 Device c  ethernet
Probing /pci@1f,0 Device 5  ethernet
Probing /pci@1f,0 Device a  usb
Probing /pci@1f,0 Device d  ide disk cdrom
todm5819 Sun Netra X1 (UltraSPARC-IIe 400MHz), No Keyboard
OpenBoot 4.0, 640 MB memory installed, Serial #50666290.
Ethernet address 0:3:ba:5:1b:32, Host ID: 83051b32.


Environment monitoring: disabled

ok
```

Open Boot PROM (OBP)



The power on self tests are used when a system is powered on or reset from the PROM level. They check out the CPU and the MLB and the tests can vary from system to system. By default the POST output is not displayed to the monitor, however it can be viewed by connecting an ASCII terminal to the serial port.
User commands are used to control the operation of the PROM. They can change PROM variables such as the boot device and security modes.
The PROM contains an entire programming language and operating system known as forth. Developed in 1978, this high-level language can fit into an 8KByte chip.

OBP: boot command

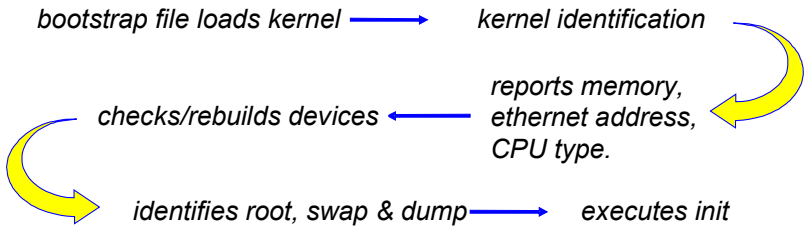
OBP: `boot` command

- To start the OS from the OBP `ok` prompt use

```
boot [device] [options]
```

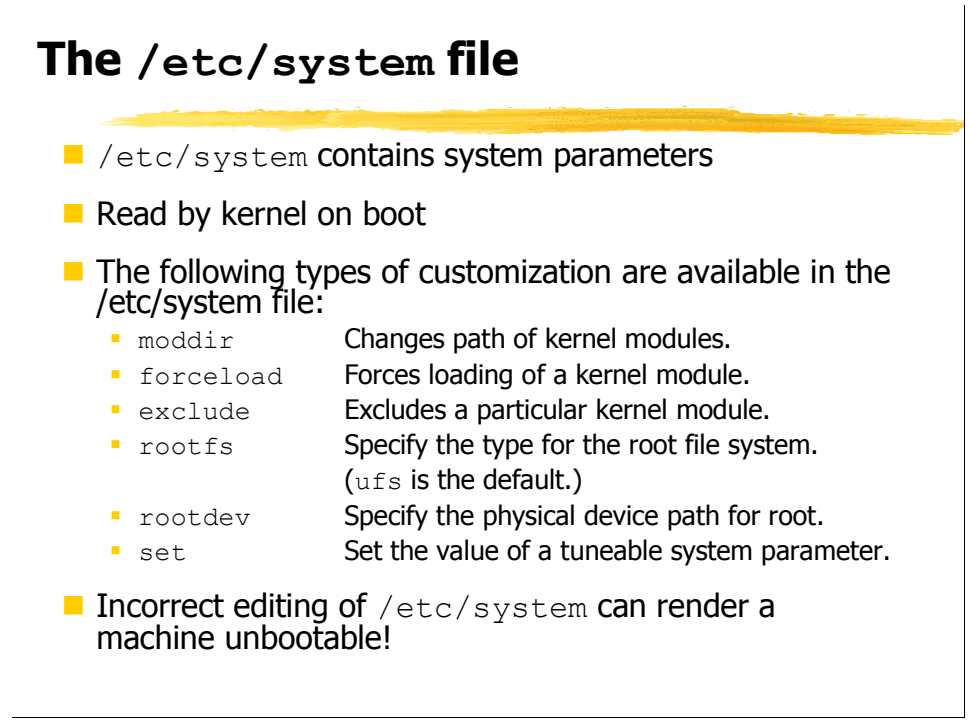
- Boot options include
  - `no options` boot from default device
  - `[cdrom]` boot from cdrom
  - `[net]` boot from network using ARP/RARP
    - `[:dhcp]` use DHCP to fetch network address
    - `[ - install]` perform a Jumpstart installation
  - `[-r]` reconfigure boot
  - `[-s]` boot to run level 1 (Single user)
  - `[-v]` verbose output
  - `[-a]` interactive boot

Kernel Initialisation



- The operating system is loaded from the boot PROM
- The kernel is the core of the operating system

The /etc/system file



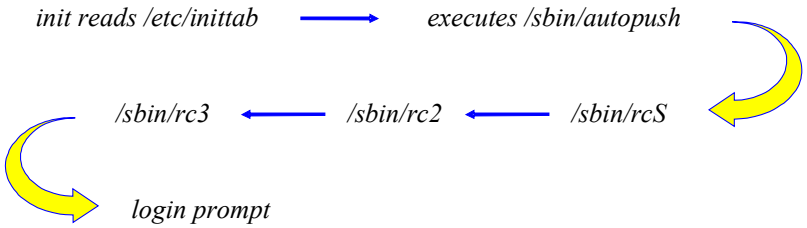
If the /etc/system file is edited, it is strongly recommended that a copy of the working file be made to a well-known location. In the event that the new /etc/system file renders the system unbootable, it might be possible to bring the system up with a boot -a command that specifies the old file. If this has not been done, the system may need to be booted from CD or network so that the file can be mounted and edited.

Boot output

Boot output

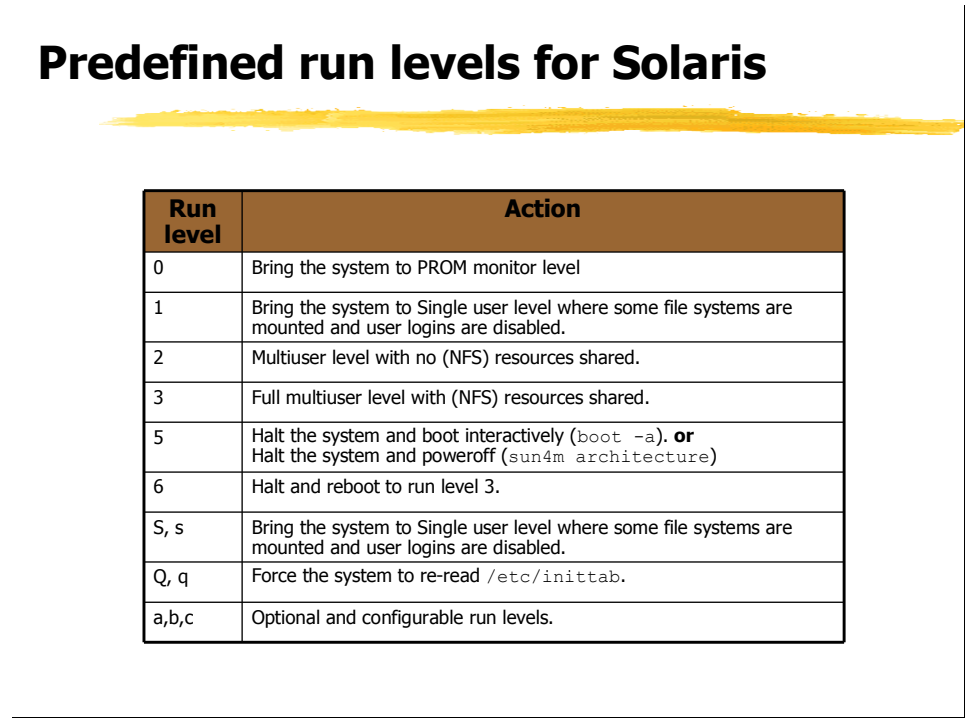
```
{ok} boot
Boot device: disk  File and args:
Loading ufs-file-system package 1.4 04 Aug 1995 13:02:54.
FCode UFS Reader 1.12 00/07/17 15:48:16.
Loading: /platform/SUNW,Ultra-Enterprise/ufsboot
Loading: /platform/sun4u/ufsboot
SunOS Release 5.9 Version Generic_112233-01 64-bit
Copyright 1983-2002 Sun Microsystems, Inc.  All rights reserved.
Use is subject to license terms.
configuring IPv4 interfaces: hme0.
starting DHCP on primary interface hme0
Hostname: trinity
The system is coming up.  Please wait.
starting rpc services: rpcbind done.
Setting default IPv4 interface for multicast: add net 224.0/4: gateway
10.100.0.100
syslog service starting.
syslogd: line 24: WARNING: loghost could not be resolved
volume management starting.
The system is ready.

trinity console login:
```

init and run levels



- Solaris works in one of a number of operational states called 'run levels'
- A 'run level' is a mode of operation in which certain resources are made available
  - a known set of processes are running
  - certain files are mounted and (possibly) shared
  - Solaris run levels are different to other UNIX-like operating systems

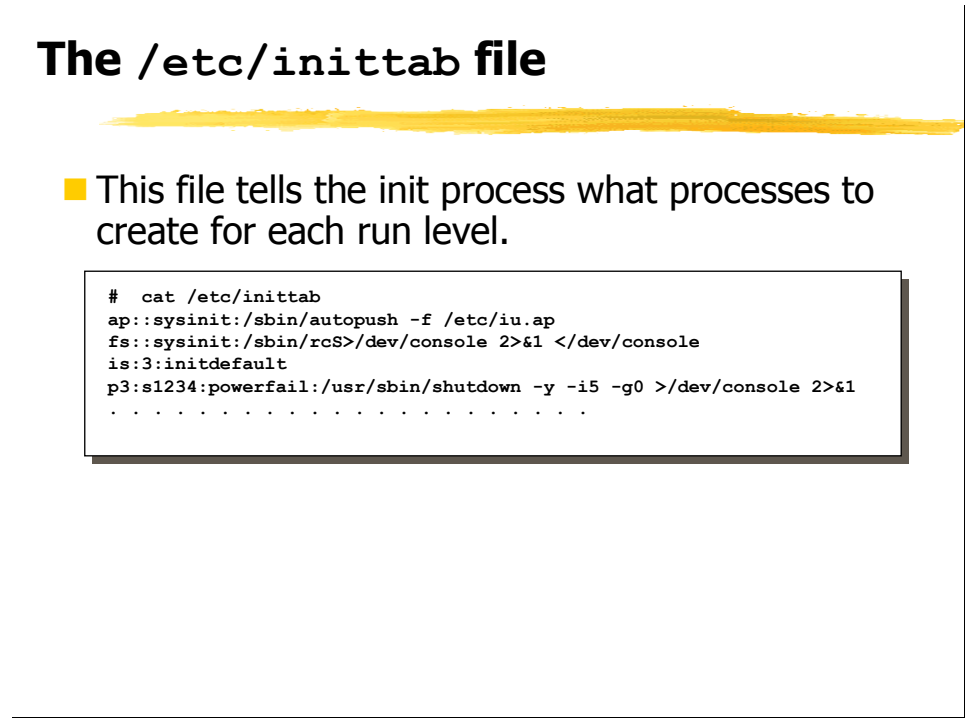
Predefined run levels for Solaris



Other UNIX-like OSs have different definitions
e.g. Linux

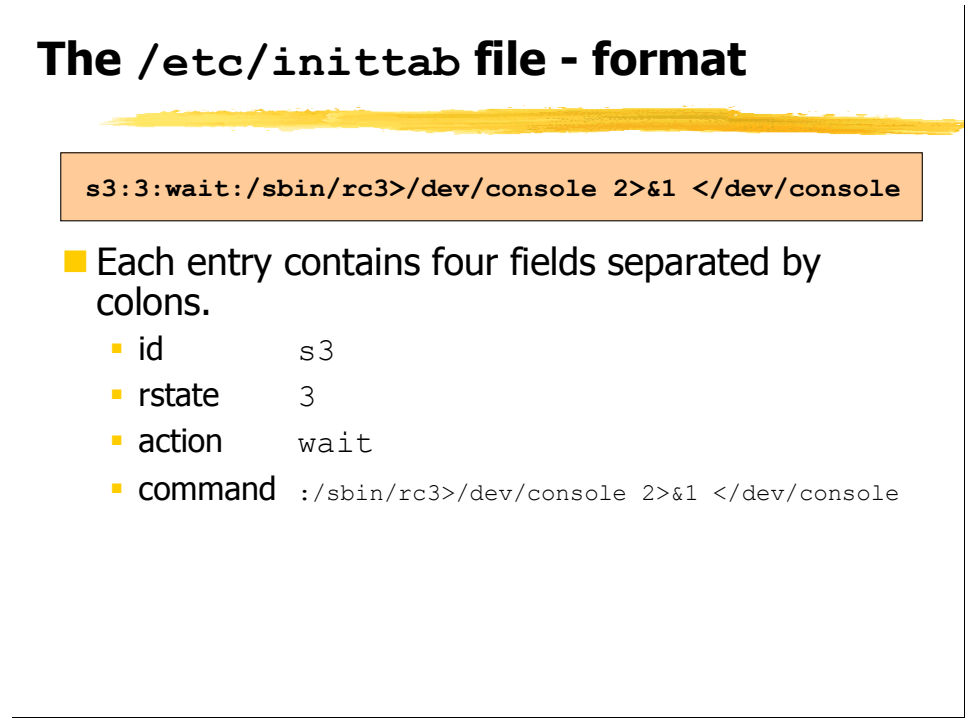
Runlevel	**System State**
0	Halt System
1	Single user mode
2	Basic multi user mode without NFS (The same as 3, if you don't have networking)
3	Full multi user mode (text based)
4	unused
5	Multi user mode with GUI
6	Reboot System

The /etc/inittab file



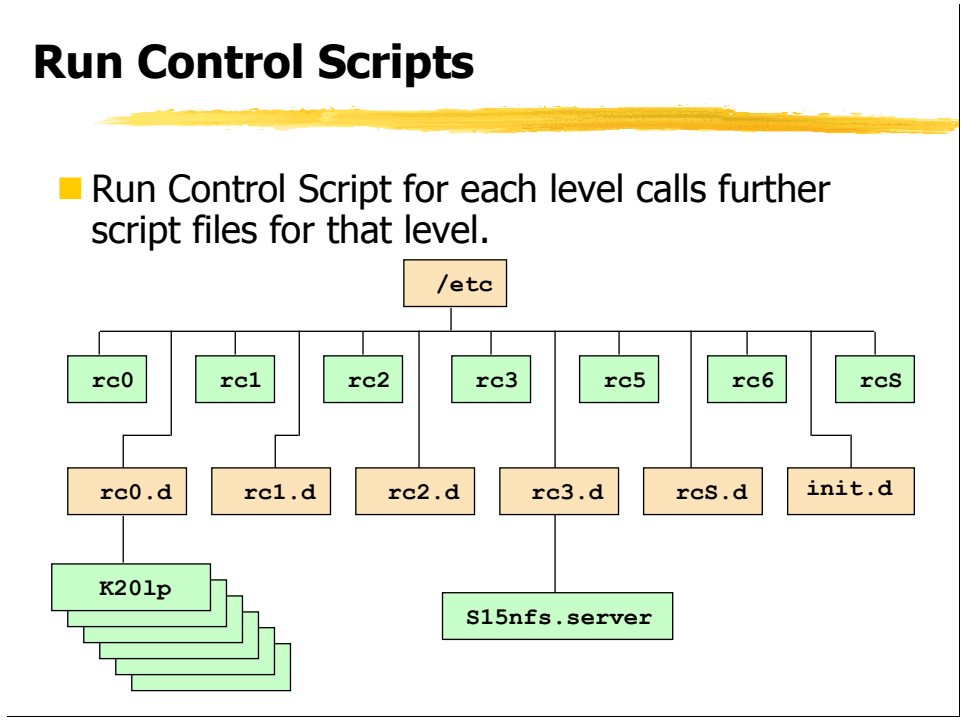
The /etc/inittab file defines three main items for the /sbin/init process.
The system's default run level.
What processes to start, monitor, and restart if they die.
What actions to be taken when the system changes run levels.

The /etc/inittab file - format



id This is a field of up to four characters which is used to uniquely identify the entry.
rstate This field defines at which run level to execute.
action This key word field tells init how to treat the process.
Valid keywords are as follows:
initdefault The default run level.
respawn Restart the process when it dies.
powerfail Start the process when init receives a power fail. (Useful for UPS systems.)
sysinit Start the process and wait for completion.
wait Start process and wait for it to finish.
cmd The command or script to execute.

Run Control Scripts



For each run level there will be an entry in /etc/inittab which calls the rc scripts. These controlling scripts are located in /sbin but are linked to files in /etc. Each run level script in turns executes the files found under the /etc/rc#.d directory in alphanumeric order.
Each run control scripts found in these sub directories is either in the format K## or S##. The K## files are 'kill files' and are used to stop processes when the system enters this run level. The S## files are 'start files' and are used to start processes.
There is also a directory /etc/init.d which is used for stopping and starting individual services or processes without having to change run levels. Each of these scripts is hard linked to the corresponding files in the /etc/rc*.d directories. These files are in simple to remember format are used with either the parameter stop or start.

/etc/init.d/lp stop
/etc/init.d/sendmail stop ; /etc/init.d/sendmail start

A Typical Run Control Script

A Typical Run Control Script

- Run Control files are simple Bourne shell scripts.

```
trinity$  cat /etc/init.d/lp
#!/sbin/sh
#
# Copyright (c) 1997, 2001 by Sun Microsystems, Inc.
# All rights reserved.
#
#ident  "@(#)lp 1.10    01/11/04 SMI"

case "$1" in
'start')
        if [ -z "$_INIT_PREV_LEVEL" ]; then
                set -- `/usr/bin/who -r`
                _INIT_PREV_LEVEL="$9"
        fi

        [ $_INIT_PREV_LEVEL = 2 -o $_INIT_PREV_LEVEL = 3 ] && exit 0

        PRINTERS=/etc/lp/printers
        CONFIG=configuration
...
```

The above example is used to start and stop the print process. The file has two other names: /etc/rc1.d/K20lp and /etc/rc2.d/S80lp. The K name ensure that the process is killed when entering run level 1, and the S name ensures that it is started when entering run level 2.

Note : To disable a start (or kill) file do not simply add on an extension. The conventional way is to change the K or S prefix to a lower case k or s. This works because the /sbin/rc scripts only search for K* and S*.

X Windows and CDE

General Observations

- The search sequence for X-Server or CDE-settings is always:
  - `$HOME/.dt/...`
  - `/etc/dt/...`
  - `/usr/dt/...`
- System-wide configuration files should never be changed in `/usr/dt`
  - Edit files in `/etc/dt` as this allow a fallback position to be maintained in case something goes wrong.
- User-specific settings like menu extensions or changes to the front panel should be stored in `$HOME/.dt`

Customizing the Login Screen

- The CDE login screen can be customised to your organization
- Edit `/etc/dt/config/$LANG/Xresources`
- Logo
  - Use a Pixmap xpm or Bitmap xbm,
  - Beware of 24-bit graphics as they can be slow to load

```
Dtlogin*logo*bitmapFile: /usr/local/lib/X11/dt/bitmaps/Mylogo.bm
```

- General welcome

```
Dtlogin*greeting*labelString: Here's %LocalHost%!
```

- Welcome after user name is entered

```
Dtlogin*greeting*persLabelString: Hello %s
```

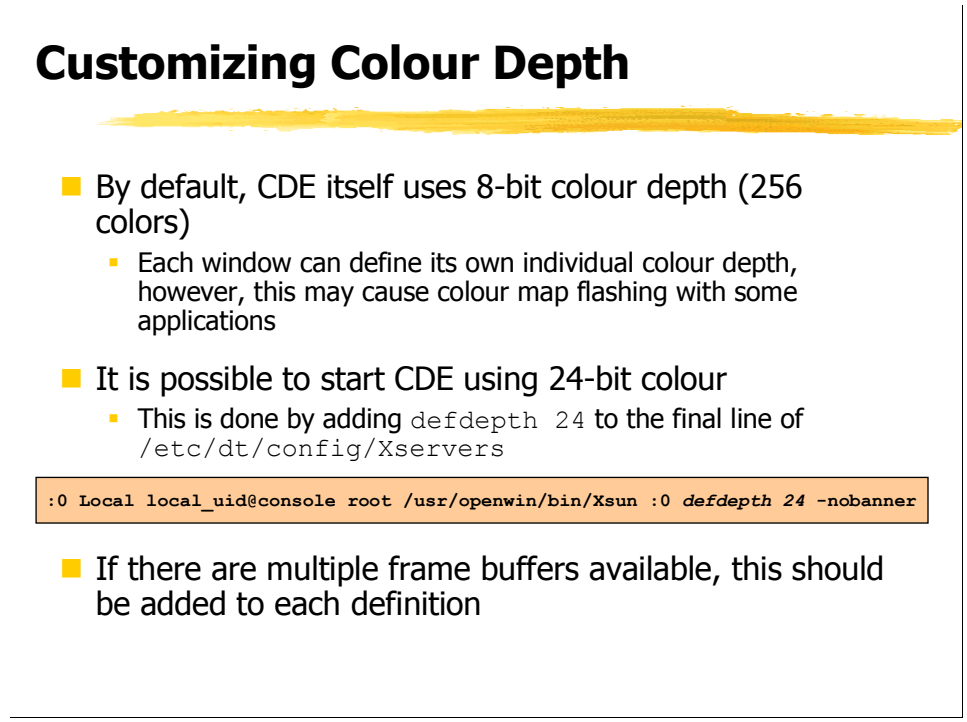
Customizing defaults

- It is possible to configure the way X-Windows starts up
- The most common things that are changed are
  - Number of colours used by the X server (colour depth)
  - How to use multiple displays
    - Multi-screen
    - Xinerama

Customizing defaults - cont.

- Multiple displays are only available on machines with graphics hardware
  - Available frame buffers are listed in `/dev/fbs`
- The X server is configured using the `Xservers` file
  - To change the defaults, copy the file from `/usr/dt/config` to `/etc/dt/config` and edit this file

Customizing Colour Depth



Note: Some legacy applications expect the default colour depth to be 8-bit, not 24-bit. This can cause problems.

Multi-Screen

- Multi-Screen means that CDE runs on every screen with a dedicated front panel
  - The mouse pointer can be moved between screens, but windows cannot
  - The display running on the first frame buffer is called :0.0, the display on the second :0.1, etc
- Add each frame buffer device into the last line of the Xservers file

```
:0 Local local_uid@console root /usr/openwin/bin/Xsun :0
        -dev /dev/fbs/ifb0 -dev /dev/fbs/ifb1 -nobanner
```

Xinerama

- Xinerama offers one virtual screen which covers all physical screens
  - Both the mouse pointer and windows can be moved among screens
  - CDE draws only one front panel
  - There is only one display number :0.0
- Xinerama is only possible with frame buffers of the same type.

Xinerama - cont.

- Configure a xinarama display by adding `+xinarama` to the final line of `/etc/dt/config/Xservers`

```
:0 Local local_uid@console root /usr/openwin/bin/Xsun :0
        +xinerama -dev /dev/fbs/ifb0 -dev /dev/fbs/ifb1 -nobanner
```

- Xinerama can define an overlap in X- and Y-direction.
  - This can be used for edge blending with overlapping projectors or to "double" an image on two monitors
  - Use `xoverlap=<horizontal resolution>`

```
:0 Local local_uid@console root /usr/openwin/bin/Xsun :0 +xinerama
        -xoverlap 256 -dev /dev/fbs/ifb0 -dev /dev/fbs/ifb1 -nobanner
```

Customizing the Front Panel

- The application start mechanism in CDE is based on so-called "actions."
- Their behavior defines what happens when the user clicks on an icon in CDE
- Once defined, actions are referenced in various environments:
  - menus
  - file manager
  - MIME-types
  - definition files for the front panel
- CDE looks for files with a `.dt` extension in the following order
  - `$HOME/.dt/types`
  - `/etc/dt/types/$LANG/`
  - `/usr/dt/types/$LANG/`

Example Action Definition

```
ACTION StarOffice52
{
        LABEL           StarOffice 5.2
        TYPE            COMMAND
        EXEC_STRING     "/opt/Office52/program/soffice" "%(File)Args%"
        ICON            So52
        WINDOW_TYPE     NO_STDIO
        DESCRIPTION     StarOffice
}
```

- Action definitions contain the name of an icon
- Icon names are resolved in this order
  - $HOME/.dt/icons
  - /etc/dt/icons/$LANG/
  - /usr/dt/icons/$LANG/
- Icons are named in the following way
  - <name>.<size=m|t|l>.<format=bm|pm>
  - So52, for example, is the icon file in $HOME/.dt./icons/So52.m.bm

Front Panels

- The front panel can be extended easily by adding files with a `.fp` extension into `$HOME/.dt/types`
- The default definitions are stored in:
  - `/etc/dt/types/$LANG/`
  - `/usr/dt/types/$LANG/`
- You may build a whole menu bar by adding additional CONTROLs in the SUBPANEL.

Frame Buffers and OpenGL

Configuring Frame Buffers - fbconfig

- `fbconfig` is the command for configuring frame buffers
- If a device is not specified, `fbconfig` operates on the default (console) frame buffer.
- To find which frame buffers are installed in a system, use

```
ls /dev/fbs
```

fbconfig - cont.

- `fbconfig` references frame buffers directly using their device names:

```
fbconfig -dev /dev/fbs/afb1 -propt -prconf
```

  - This prints configuration information for the second Elite3D frame buffer configured in the system.
  - The first Elite3D would have the device name afb0.

Changing screen resolution

- Find out which resolutions are available using

```
fbconfig -res \?
```

- Change resolution with

```
fbconfig -res "1280x1024x86" -try -now
```

  - `-try` changes the resolution for 10 seconds only
  - `-now` switches the resolution immediately, without the need to restart the X-Server
  - In some cases, the geometry of the X-Server may look distorted after -now
    - This is fixed by restarting the X-Server

Changing Gamma Level

- The default gamma value is 2.2, which often is perceived as too bright

```
fbconfig -g 1.7
```

  - Sets the gamma factor to 1.7
- The new gamma factor does not need an X-Server restart.
- After a reboot, the newly set value is still valid.

OpenGL

- OpenGL® is not installed automatically during Solaris installation
  - Although it has been packaged with the installation CDs since Solaris 2.5.1
- To checks if OpenGL is installed correctly and which version is available, use

```
/usr/openwin/demo/GL/ogl_install_check
```

Note: The rotating ring comes in handy when judging gamma correction values

OpenGL - cont.

- It is recommended to always use the latest version of OpenGL
  - Xinerama, for example, is only supported with OpenGL 1.2.1 or later
- OpenGL can be downloaded from

```
http://www.sun.com/software/graphics/opengl/download.xml
```

Direct and Indirect Graphics Access

- Direct Graphics Access (DGA) is the standard fast method of accessing the frame buffer
- If you switch users during a session, DGA will be disabled
  - This is a security feature in the OS
  - To remove the security feature, as root
    - Edit the permission in `/etc/logindevperm` from `0600` to `0666`:
      `/dev/console 0666 /dev/fbs/* # frame buffers`
    - Reboot
  - **Note:** any user will now have read/write access to the contents of your screen

Direct and Indirect Graphics Access

- To check if you are using DGA, use

```
ogl_install-check
```

  - `GLX: context is direct`
    - OpenGL uses DGA
  - `GLX: context is indirect`
    - OpenGL does not use DGA

- DGA will also be switched off when $DISPLAY is set to "hostname:display", e.g. `trinity:0.0`
  - Better to use the default local display name `:0.0`

VNC and inetd

- VNC can be used to access an X windows display
  - The VNC X server is used
  - This is often used to allow remote access to a single desktop
- VNC can also be invoked using inetd
  - This starts a CDE login session for each VNC connection
  - Allows multiple desktops to be shared
  - Enforces standard security

Configuring VNC and inetd

- Define new ports in `/etc/services`

```
vnc-800x600x16     5960/tcp
vnc-800x600x24     5961/tcp
```

  - These are usually in the 5900 range
  - Used as the port to connect to
  - Only uses one port
    - Easier to allow through a firewall

Network Connections

/etc-files

- Solaris software stores network configuration files in the directory `/etc`
- All configuration data, such as IP addresses, gateways, and so on, can be defined and changed at runtime level
- However, only those definitions stored in `/etc` files will survive a reboot

Hostname

- The hostname can be found in these files:
  - `/etc/nodename`
  - `/etc/hostname.<interface-name>`
  - `/etc/inet/hosts`
  - `/etc/net/{ticlts,ticots,ticotsord}/hosts`

/etc/nodename

- `/etc/nodename` defines the name of the system
- A system with multiple network interfaces and multiple IP addresses can have multiple names
  - These are defined in `/etc/inet/hosts`
  - One of these names may be equal to the nodename, but this is not necessarily the case
- A system with a single network interface may still have multiple names defined in DNS
  - Allows services to be accessed by a single name

Interfaces

- The connection between IP addresses and interfaces
  - These are named `/etc/hostname.<interface-name><number>`
- The common interface names include:
  - `le` Ethernet on older SPARC® systems
  - `hme` FastEthernet on most UltraSPARC® systems
  - `eri` FastEthernet on most UltraSPARC-III systems
  - `qfe` FastEthernet on QuadFastEthernet-extension cards
- Do not create hostname.* files for unused interfaces
  - Causes error messages about interfaces with no network cable attached

Interfaces - cont.

- A system with more than one network interface will automatically route between the subnets
  - During boot, the message "machine is a router" is printed.
  - If this behavior is not desired, create an empty file `/etc/notrouter` and reboot.
- The netmasks for the IP addresses are defined in the file `/etc/netmasks`
- Do not change the nodename while the X-Server is running!
  - It is better to log out from the current session and log in again using "Command Line Login" before changing the nodename or hostname.

Virtual Interfaces

- It can be useful to define multiple IP addresses for a system with only one physical network interface.
  - For example, a demo workstation could keep its usual IP address while getting an additional address to communicate with the network at the demo location
- This can be done with "virtual interfaces"
- Virtual interfaces are assigned IP addresses in the same way as physical interfaces
- Virtual interface names end in :<number>
  - `hme0:1` is the first virtual interface on the first `hme` physical interface

Managing interfaces - ifconfig

- To list all interface definitions, physical as well as virtual ones, with IP-addresses and netmasks use `ifconfig -a`
- `ifconfig` is also used to define IP addresses and netmasks during runtime
  - Works with both physical and virtual interfaces

```
trinity$ ifconfig hme0:1 plumb
trinity$ ifconfig hme0:1 10.100.0.1 netmask 255.255.0.0 up
trinity$ ifconfig -a
lo0: flags=1000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv4> mtu 8232 index 1
        inet 127.0.0.1 netmask ff000000
hme0: flags=1004843<UP,BROADCAST,RUNNING,MULTICAST,DHCP,IPv4> mtu 1500 index 2
        inet 172.16.0.122 netmask ffffff00 broadcast 172.16.0.255
        ether 8:0:20:4:e8:34
hme0:1: flags=1000842<BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
        inet 10.100.0.1 netmask ffff0000 broadcast 10.255.255.255
```

Routing

- The default router (or standard-gateway) can be entered with its IP address into the file `/etc/defaultrouter`
  - If you specify a host name in that file, ensure the host name is defined in `/etc/inet/hosts`.
- It is also possible to specify a default router at runtime

```
trinity$ route flush
trinity$ route add default 10.100.0.1
trinity$ netstat -r

Routing Table: IPv4
  Destination           Gateway           Flags  Ref   Use   Interface
-------------------- -------------------- ----- ----- ------ ---------
10.100.0.0           trinity              U         1      2  hme0
BASE-ADDRESS.MCAST.NET netrax2              U         1      0  hme0
default              router.matrix.com    UG        1      0
localhost            localhost            UH        2     48  lo0
```

IP Forwarding

- If more than one network interface is available IP forwarding is enabled
  - To check if IP forwarding is switched on, use

```
ndd -get /dev/ip_forwarding
```

  - To turn it on, use:

```
ndd -set /dev/ip_forwarding 1
```

  - To turn ip-forwarding off, use:

```
ndd -set /dev/ip_forwarding 0
```

  - Another way of turning IP forwarding off is to create the file `/etc/notrouter`

NIS Client

- NIS (Network Information Service) is one of the name services available with Solaris
- To configure a machine to be a NIS client
  - Enter the name of the NIS domain into `/etc/defaultdomain`
  - Define the NIS server in `/etc/inet/hosts`
  - Run the NIS client configuration: `ypinit -c`
  - Enter the name of the NIS server
  - Start the NIS service using `/usr/lib/netsvc/yp/ypstart`
  - Configure `nsswitch` to use NIS
    - Make a copy of `/etc/nsswitch.conf` and replace it with `/etc/nsswitch.nis` which is a predefined configuration file

DNS Client

- To configure a machine to be a DNS client
  - Enter your DNS server into the file `/etc/resolv.conf` in the format
  - nameserver <ip address>
  - Configure nsswitch to use DNS
    - Either use the `/etc/nsswith.dns` configuration file or append dns to the files line in your current `/etc/nsswitch.conf` file
- Testing DNS
  - Use the `nslookup` and ping commands to ensure than hostnames are translated to IP addresses

DHCP Client

- To configure an interface to use DHCP
  - Create two empty files
    - `/etc/hostname.<interface>`
    - `/etc/dhcp.<interface>`
- By default, Solaris waits 30 seconds for a DHCP server to answer
  - This can be modified in `/etc/dhcp.<interface>` by adding
    -

```
WAIT <time in seconds>
```

- It is also possible to start DHCP on an interface at runtime

```
trinity$ ifconfig qfe0 plumb
trinity$ ifconfig qfe0 dhcp start
```

sys-unconfig

- The `sys-unconfig` command is used to restore a system's configuration to an "as-manufactured" state, ready to be reconfigured again
  - Root password is removed
  - All network interfaces are deconfigured
  - All network service configurations, including DHCP, NIS, DNS and LDAP, are removed
  - Timezone is reset
  - SSH keys are regnerated
  - When compete, the machine is halted
- On next system boot, the system configuration tools from installation are re-run
- This command should be used with extreme caution

Querying interfaces

- The state of a network interface can be examined using the `ndd` command

```
ndd <device> <option>
```

  - Options include
    - link_status 0 = down 1 = up
    - link_speed 0 = 10Mbit 1 = 100Mbit
    - link_mode 0 = half-duplex 1 = full duplex

Setting interface parameters

- The `ndd` command can also be used to set network speed and duplex type, for networks which do not auto negotiate

```
ndd -set <device> <option> <setting>
```

  - Options include
    - adv_100fdx_cap, adv_100hdx_cap
    - adv_10fdx_cap, adv_10hdx_cap
    - adv_autoneg_cap
  - Settings
    - 0 disables the capability
    - 1 enables the capability
- The settings can also be defined in the `/etc/system` file

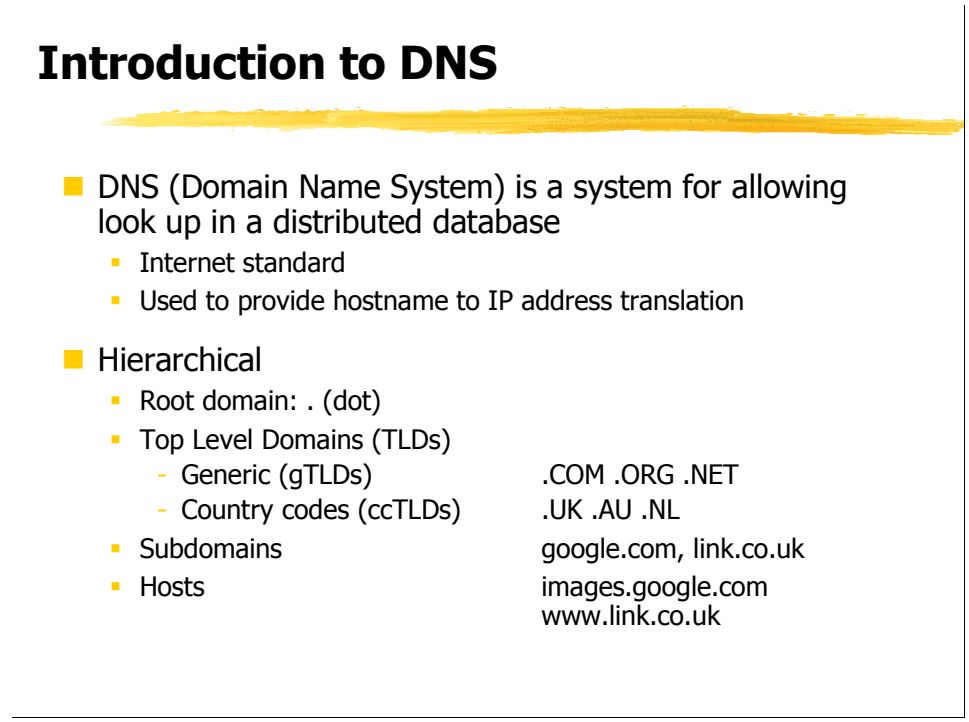
```
set hme:adv_10hdx_cap = 1
```

Netgroups

- Netgroup is a network wide group of users or machines
  - They are stored in naming services, e.g. LDAP
- Netgroups can be used to control
  - NFS mount access
  - remote login and shell access
  - local login access
- Netgroups do not directly set permissions or access rights
  - Netgroup names are used in places where a user name or machine name would normally be used

Administration

Introduction to DNS



The Domain Name System is an Internet Standard. It is described in RFC 1034 and 1035, with a number of later RFCs augmenting this description.
The purpose of the Domain Name System is to create a system that allows lookups in a tree-like database. These lookups are mostly (but not only) finding an IP address that belongs to a "node" (a hostname) in the Domain Name System. A hostname in this respect is always a "Fully Qualified Domain Name" (FQDN).
The DNS system knows a hierarchical structure:
• The root node is the "dot" domain. This dot is the origin of all domains. It is comparable with the root of a UNIX filesystem.
• Below the root node you will find a number of Top Level Domains (TLDs). These can further be distinguished in Generic Top Level Domains (gTLD), such as com, org and net, and Country Code Top Level Domains (ccTLDs), such as nl (for the Netherlands), au (for Australia) and uk (for the United Kingdom).
• Below a Top Level Domain an organization can apply for a subdomain. The application criteria and procedure for this varies from TLD to TLD. When an application has been granted, then that organization becomes the "owner" of a domain, and can use it to store information about its own hosts and (possibly) other subdomains.
Furthermore, the DNS system is decentralized. This means that there is no central database which holds all the information, but organizations all keep their own databases on their own servers. Through special so-called "glue records", these databases all point to each other, making global lookups possible.

Introduction to DNS - cont.

- Globally decentralized
  - Every domain implements its own tables and servers
  - Every domain can do its own delegation
- Locally centralized
  - In each domain, the hostname to address translation is centralized on the nameserver
  - Avoids separate `/etc/inet/hosts` file being maintained on every machine

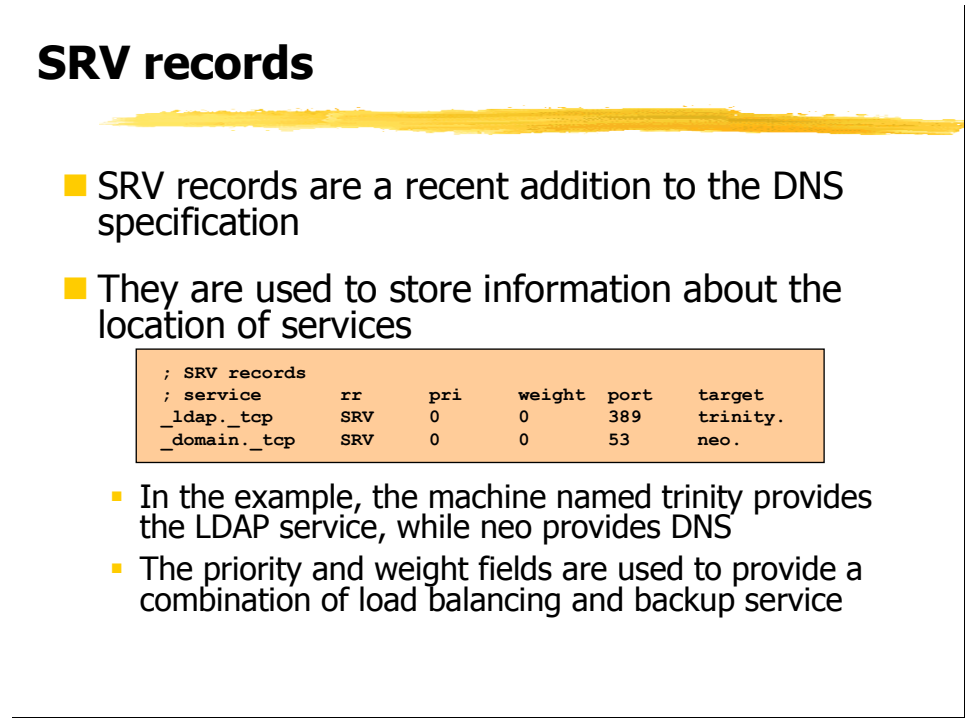
Resource Records (RRs)

Resource Records (RRs)

- Common RRs for hosts
  - A (Address) — IP address of the node
  - PTR (Pointer) — Hostname of the node
  - CNAME (Common Name) — Alias of node name
  - HINFO (Host Info) — Information about this host
- Common RRs for domains:
  - NS (Name Server) — The nameserver
  - MX (Mail Exchange) — The mail server
  - SOA (Start of Authority) — Indicates top of local tree
  - SRV (Service) — Used to locate services other than mail

The hierarchical structure as shown in the previous visual can be thought of as the key to the database. With an FQDN we can find the record for a specific host. The next thing we need to retrieve is the data that is stored about this host. This is done through a series of "resource records".
Each resource record stores something about the "node", as each host or domain is called in DNS-speak. What is stored, depends on the resource record type. There are several resource records possible. Some are typically only used for a host, and others are typically only used for a domain. But there is no general rule in this respect. In fact, the DNS system doesn't even know the difference between a host or a domain.
Common RRs for a host include:
A (Address) This RR gives the IP address of a node.
PTR (Pointer) This gives the FQDN of a node.
CNAME (Common Name)
This is used to define aliases. The CNAME is stored with the alias and lists the official name of a node.
HINFO (Host Information)
This gives information about the host itself, such as hardware, operating system, administrative contact and so on.1
Common RRs for a domain include:
NS (Name Server) This identifies a name server for this node.
MX (Mail Exchanger) This identifies the mail server for this node.
SOA (Start of Authority) This indicates that a node and all nodes below it are managed by a different organization from the nodes above. It identifies the organization that manages this node and below, and gives some timing parameters for this domain. These parameters have to do with how long entries may be cached and how often slave name servers need to check for updates, for instance

SRV records



An SRV record holds the following information:
Service: the symbolic name of the desired service.
Protocol: this is usually either TCP or UDP.
Domain name: the domain for which this record is valid.
TTL: standard DNS time to live field.
Class: standard DNS class field (this is always *IN*).
Priority: the priority of the target host.
Weight: A relative weight for records with the same priority.
Port: the TCP or UDP port on which the service is to be found.
Target: the hostname of the machine providing the service.

Clients always use the SRV record with the lowest priority value first, and only fall back to other records if the connection with this record's host fails. Thus a service may have a designated "fallback" server, which will only be used if the primary server fails.
If a service has multiple SRV records with the same priority value, clients use the **weight** field to determine which host to use. The weight value is relevant only in relation to other weight values for the service, and only among records with the same priority value.

Using SRV records to start services

- SRV records can be used to start services on a machine
- Add a script to the start up sequence
  - Read SRV records from DNS
  - Check targets listed against local hostname
  - Start any services where the target and hostname match
- A more complex script might run continuously
  - Check DNS periodically for updates
  - Start new services as they appear

Configuring Simple DNS

- DNS requires at least one configuration file and four data files in order to function
  - Configuration file is `/etc/named.conf`
  - Configures name server daemon
- DNS data is stored in plain text files also known as zone files
  - Contain resource records
  - Root file for accessing internet name servers
  - Local file for the loopback interface
  - Forward lookup file for the domain
    - Converts names to IP addresses
  - Reverse lookup file for the domain
    - Converts IP addresses to names

Configuring Simple DNS - cont.

- One hostname may have many IP addresses
  - A name that references more than one IP address offers a simple form of load balancing
- One IP address may have many names
  - A single machine can hold many services
  - Reference each service by service name
    - Makes it easier to manage moving the service to another machine
    - DNS is changed and all hosts automatically reference new location
  - This can also be done using CNAMES
    - Useful for referencing hosts in other domains

Sample named.conf

Sample named.conf

```
options {
         directory "/var/named";
         };
zone "." IN {
        type hint;
        file "named.ca";
};
zone "localhost" IN {
        type master;
        file "localhost.zone";
        allow-update { none; };
};
zone "matrix.com" IN {
        type master;
        file "matrix.com.zone";
};
zone "0.10.100.in-addr.arpa" IN {
        type master;
        file "0.10.100.in-addr.arpa.zone";
};
```

Sample forward lookup file

Sample forward lookup file

```
$ORIGIN matrix.com
$TTL 5h
;
; SOA rec
@        IN       SOA      localhost.       root.localhost (
                                    2005051901 ; serial number (YYYYMMDD##)
                                    10800 ; refresh every 3 hours
                                    10800 ; retry every 3 hours
                                    604800 ; expire after a week
                                    86400 ) ; Time To Live of 1 day
; Name Servers
                 IN       NS       localhost.
@                IN       A        127.0.0.1
; Mail Servers
matrix.com.      IN       MX       10       mail
matrix.com.      IN       MX       20       remote.mail.backup.org.
; Addresses
trinity          IN       A        10.100.0.100
niobe            IN       A        10.100.0.25
smith            IN       A        10.100.0.1
; Aliases
mail             IN       A        trinity
www              IN       A        smith
```

Sample reverse lookup file

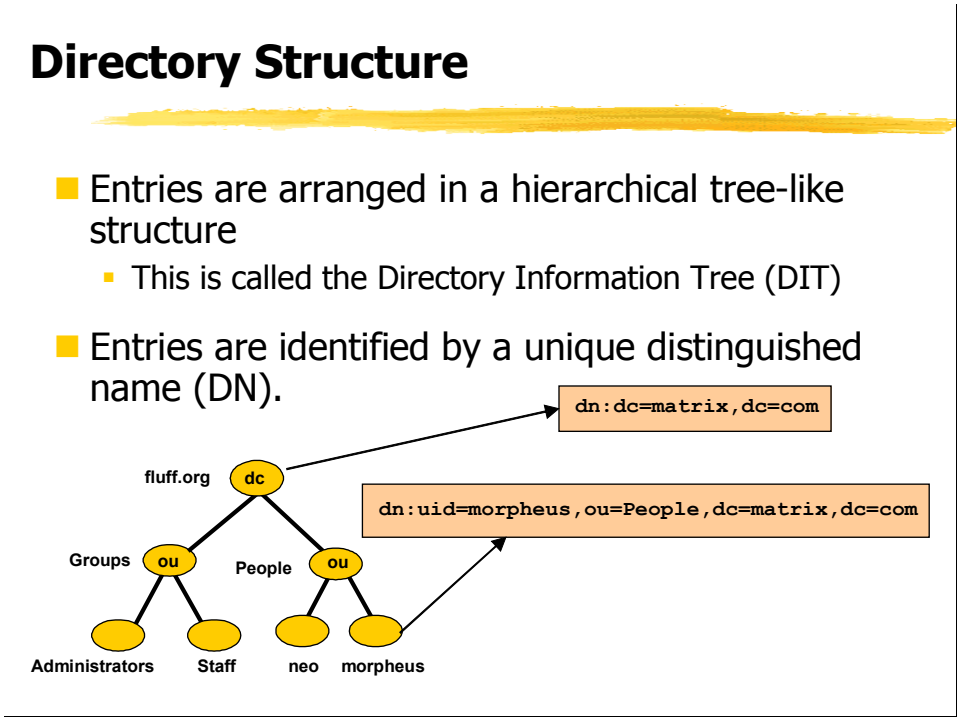
Sample reverse lookup file

```
$TTL 5h
;
; SOA rec
@        IN       SOA      localhost.       root.localhost (
                                    2005051901 ; serial number (YYYYMMDD##)
                                    10800 ; refresh every 3 hours
                                    10800 ; retry every 3 hours
                                    604800 ; expire after a week
                                    86400 ) ; Time To Live of 1 day
; Name Servers
                 IN       NS       localhost.
@                IN       A        127.0.0.1
; Addresses
100              IN       PTR      trinity.matrix.com.
25               IN       PTR      niobe.matrix.com.
1                IN       PTR      smith.matrix.com.
```

Introduction to LDAP

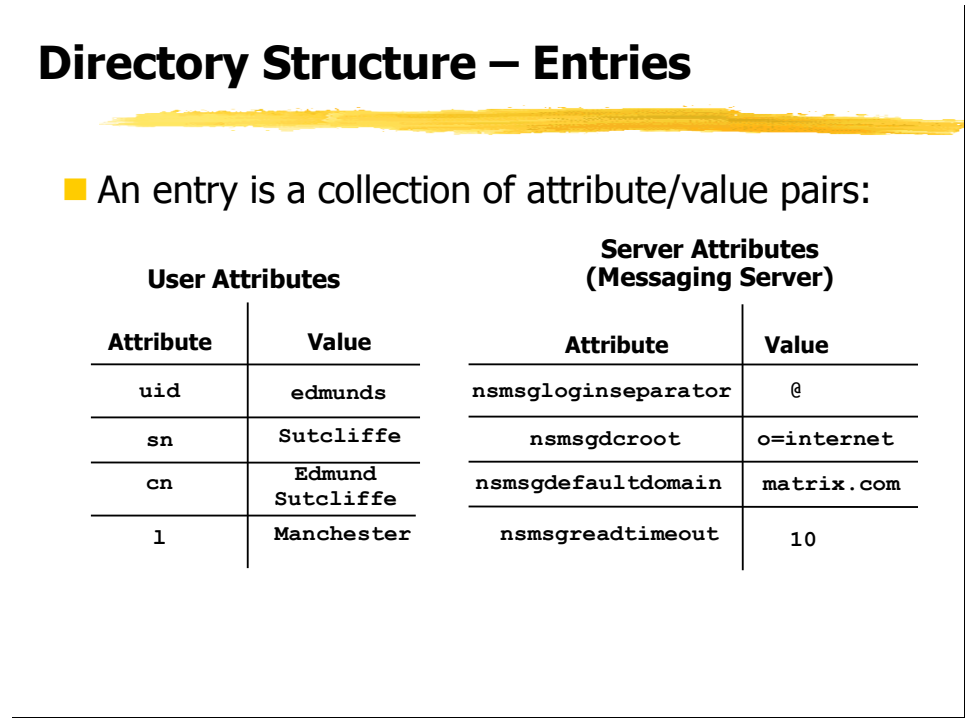
- LDAP, Lightweight Directory Access Protocol, is a platform-independent method of accessing directory services
- Directories are usually used to hold
  - User identification data
  - Application configuration data
- Directories can be stored and managed centrally
  - This allows, for example, users and groups to be available on all machines

Directory Structure



LDAP directories store data in a hierarchal structure that can be easily queried and updated. The data structure resembles the root system of a tree with many branches fanning out from a single root.
Some attributes identify an entry's location in the directory tree. For example, in the preceding illustration, the domain Component (dc) attributes and the organizational Unit (ou) attribute, are used to organize the directory structure. The dc attribute (RFC 2247) was created to provide organizations with a method of creating distinguished names that correspond to those that they have obtained for use on the Internet.
To allow applications to locate entries in the directory structure, LDAP uses a unique identifier called the *Distinguished Name* (DN). The DN consists of a list of key entry attributes. No two entries in the directory server's database can have the same DN. See the above example,the DN for Edmund
Sutcliffe is uid=edmunds,ou=People,dc=fluff,dc=org.
The *suffix* is defined as the DN that identifies the root level of the directory. In the example, the suffix is dc=fluff,dc=org.

Directory Structure – Entries



The directory database consists of entries. An *entry* is a collection of information about a particular object, expressed in terms of attributes. Attributes can have any type of value. In LDAP terminology, attributes and their values are referred to as *attribute/value pairs*.
Attributes contain information about a specific aspect of the entry. Each attribute is defined by an attribute type and can have one or more values.

The attribute type defines what data an attribute value can contain. An attribute type could, for example, be either a string, a binary value, or even another attribute. The attribute value must match the attribute type. For example, the value for a uid attribute must be a string.

Introduction to DHCP

- DHCP (Dynamic Host Configuration Protocol) is a method of automatically assigning IP addresses
- By default, Solaris uses Sun's DHCP server
- On non-Solaris systems, the common DHCP server is provided freely by the Internet Systems Consortium (ISC)

Advantages of DHCP

- DHCP centralizes and simplifies management of IP address assignment
  - Dynamic pools can assign addresses automatically to "floating" machines, such as laptops, whenever they appear on the network
  - Static addresses can be assigned to Ethernet (MAC) addresses for servers, and static workstations
  - Clients can use DHCP to obtain the information needed to boot from a server on the network
  - Clients can be grouped to receive different information

Sun DHCP vs. ISC DHCP

- The Sun DHCP server uses a proprietary binary format to store its data
- ISC DHCP uses plain text files
- It is possible to script conversions between ISC DHCP and Sun DHCP formats

Hosts

- The `/etc/inet/hosts` file contains a list of names that refer to IP addresses
  - `/etc/hosts` is a link to `/etc/inet/hosts`
- The hosts file generally only refers to the local machine
- Certain definitions are required by some applications
  - The syslog daemon expects to find a loghost
  - The sendmail daemon expects to find an alias containing at least one dot
  - The primary host in `/etc/inet/hosts` usually looks like this:

```
172.16.0.25 trinity trinity.matrix.com loghost
```

Serial Ports

- Most systems have two serial ports
- Port A, the first serial port, is usually used for the serial console
- Serial ports can be configured with tools such as `admintool` and the "Solaris Management Console", `smc`

Printers

- Printers can also be configured with `admintool`
- Solaris 9 comes with a "Printer Setup Wizard"
  `/usr/sadm/admin/bin/printmgr`

User Creation

- Users can be created using `admintool` or `smc`
- Users can also be created using the `useradd` command
- It is often useful to use a script to create users
- By default, `/home/<username>` is used by the automounter and is not writeable because of this.
  - Configure the automounter or use `/export/home/<username>` as the home directory

Shells and Environment Variables

- Solaris provides several shells including `sh`, `ksh`, `csh` and `bash`
- `sh`, the Bourne shell, is the default shell for root
- The default shell for users is `ksh`
  - This is defined in the `/etc/passwd` file or the name service
  - `bash` is a popular alternative user shell

Setting Environment Variables

- The syntax for working with variables depends on which shell is used

Shell	Setting	Appending	Retreiving
sh	NAME=value1 export NAME	NAME=value2:${value1}	set (local) env (global)
ksh	NAME=value1 export NAME	NAME=value2:${value1}	set (local) env (global)
csh	setenv NAME value1	setenv NAME value2:${value1}	printenv
bash	NAME=value1 export NAME	NAME=value2:${value1}	set (local) env (global)

Shell Initialization Files

Shell Initialization Files

- Every shell reads a set of files during startup
  - These differ if the shell is a subshell or a login shell

shell	on login	on new shell
sh	/etc/profile $HOME/.profile	
csh	/etc/.login ~/.cshrc ~/.login	~/.cshrc
ksh	/etc/profile ~/.profile ~/.kshrc	~/.kshrc
bash	/etc/profile ~/.bash_profile or ~/.bash_login or ~/.profile	~/.bashrc

Typically the ~/.bash_profile contains

```
if [ -f  '~/.bashrc' ];
    then source '~/.bashrc';
fi
```

as the last entry to get ksh like behaviour

/etc/profile

- Global environment configuration is stored in `/etc/profile`
- It is read by all login shells
  - Avoid making changes in this file
    - They are local to a machine
    - They can break all users
  - Use `$HOME/.profile` instead
  - May also create a `/home/`*`groupname`*
    - Source `$GROUPHOME/.profile` in a user's `.profile`
    - Allows environment to be managed for groups of users in a single place

Useful Environment Variables

- `$PATH`
  - Contains a list of one or more directories where commands are stored.
  - In general, don't have `.` (current directory) in `$PATH`
    - Especially for root
  - Set on start up of login shell
- `$PWD`
  - Contains absolute path of current working directory
- `$$`
  - Contains the PID of the current process

Useful Environment Variables - cont.

- `$?`
  - Contains the exit status of last completed command
- `${0..n}`
  - positional parameters from command line
    - `$0` first parameter, i.e. the name of the command
    - `$#` number of parameters
- `"$@"`
  - a string containing all parameters with each one quoted individually

Displaying the environment

Displaying the environment

- `set`
  - Displays the current shell variable space (local)
- `env`
  - Displays the current environment space (global)

```
trinity$ set
ERRNO=1
FCEDIT=/bin/ed
HOME=/
IFS='
'
LINENO=1
LOGNAME=root
MAILCHECK=600
OPTIND=1
PATH=/usr/bin:/bin:/usr/sbin:/sbin
```

```
trinity$ env
_=/usr/bin/env
SSH_TTY=/dev/pts/1
PATH=/usr/bin:/bin:/usr/sbin:/sbin
LOGNAME=root
USER=root
SHELL=/sbin/sh
HOME=/
SSH_CLIENT=10.10.0.25 34415 22
TERM=screen
PWD=/
```

Some variables appear in both listings.

Shell Scripts

- Scripts are files that contain lists of commands that are executed one after the other
  - Must have executable bit set in order to run
    - Use `chmod +x` *`filename`*
  - Run, like all commands, by entering the name of the file
- `#!/bin/{sh|ksh|csh|bash}`
  - First line of script file
  - Defines file as a script, and explains which shell will run the commands within the script

Shell Scripts

- Scripts can be simple
  - Run commands sequentially
- Scripts can be complex
  - Flow control offers branching and looping
- Conditions tested using the `test` command
  - Often seen in shorthand form `[ condition ]`
    - Available tests include
      - File tests
      - Numeric and string comparison
      - See man test

Shell Scripts

- Branching
  - Use the `if/then` construct to run one or more command depending on a condition being true
  - Use `elif/then` to run a different test should the preceding condition be false
    - Allows cascading tests
  - Use `else` to run a different set of commands should no other condition be evaluated as true
  - `if` statements may be nested

Network File System

NFS Server

- NFS, Network File System, shares directories around a network
  - NFS servers hold the data
- It is configured with the file `/etc/dfs/dfstab`
  - This file is a script containing share commands
  - Stores which directories are shared, which machines can mount them and what access is granted
- If this file is empty
  - The NFS server daemon is not started during reboot
  - `/etc/init.d/nfs.server` start does nothing

Configuring NFS Server

- Sharing `/export/home`
  - Edit `/etc/dfs/dfstab`, adding:

```
share -F nfs /export/home
```

  - If this is the first entry, start the NFS server daemon: `/etc/init.d/nfs.server start`
  - Share the directory using: `shareall`
  - Check if `/export/home` has been correctly exported

```
trinity# share
-               /export/home   rw   ""
```

NFS-Client

- A NFS client does not need any special daemons
- It does need to be able to access the NFS server by name
- All NFS servers must be defined in DNS or `/etc/inet/hosts`
- To manually mount a network directory use:

```
mount server:/export/home /tmp/home
```

NFS-Client - cont.

- If a filesystem should always be mounted at boot time, add the mount to `/etc/vfstab`

```
server:/shared/directory - /mount/point nfs - yes rw
```

- Local `root` user does not act as `root` on the network directory
  - Unless configured otherwise, `root` will act as user `nobody`
    - File access permissions for other users will be enforced
  - Share filesystem with `anon=0` to allow root access as root.

NFS on other UNIX Systems

- NFS was originally developed by Sun and licensed to many other vendors
- There are now differences between the implementations
  - The file `/etc/dfs/dfstab` is a Solaris specific file
  - Other popular UNIX systems (Linux, AIX, and so on) use `/etc/exports` which has a different syntax
  - The analogous command to `shareall` on other systems is `exportfs -a`

Automounter

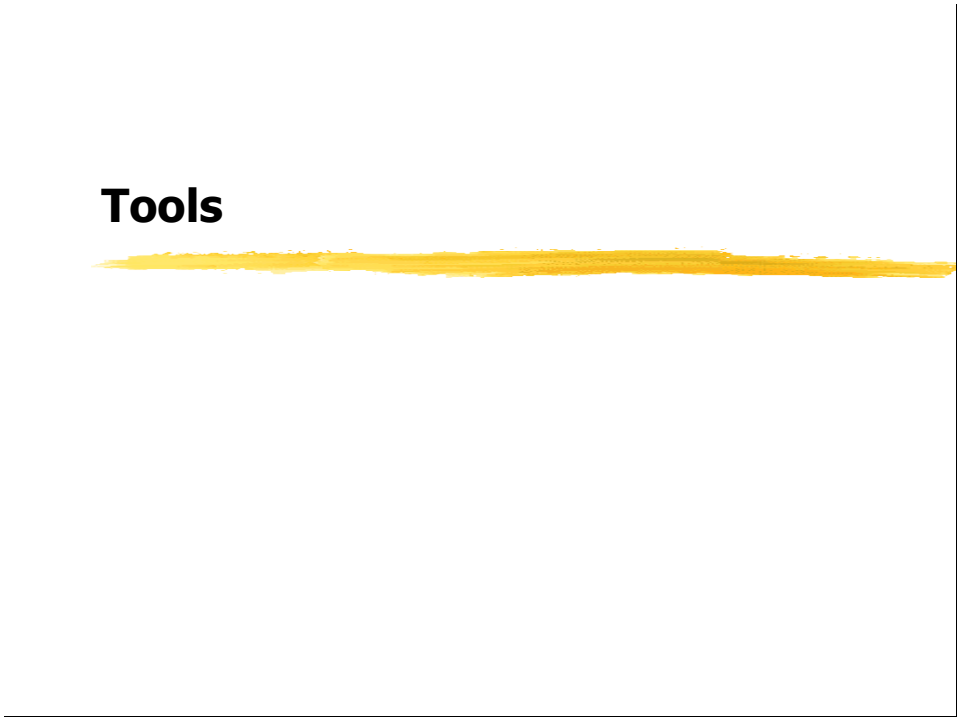
- The automount daemon automatic mounts a filesystem when a directory is entered
- It is useful to automount directory paths like `/net/<hostname>` or `/home/<username>`
  - `/net/`*`hostname`* accesses a share on a remote machine
  - `/home/`*`username`* provides consistent home directories over the entire network
- The automounter is configured using
  - `/etc/auto_master` main configuration
  - `/etc/auto_*`

auto_master

- `auto_master` comes with definitions for `/net` and `/home`

```
/net -hosts -nosuid,nobrowse
/home auto_home -nobrowse
```

  - `/net` is automatically extended with the proper host name when accessed
  - `/home` is defined in `auto_home`


Tools

CVS

- CVS (Concurrent Versions System) is an open source version control system
  - Using it, you can record the history of your source files
- It is provided for Solaris as part of the Sun Freeware project
  - Install SFWcvs from the Companion CD

Using CVS - Repository

- The CVS repository stores a complete copy of all the files and directories which are under version control
- Creating a repository
  - Repositories are created using the CVS init command

```
cvs -d /usr/local/cvsroot init
```

  - The repository should be accessible (directly or via a networked file system) from all machines which want to use CVS in server or local mode
    - The client machines need not have any access to it other than via the CVS protocol.

Using CVS - Managing files

- CVS commands expect the CVSROOT environment variable to be set to the location of the CVS server repository
- Checking things into a repository
  - cvs commit *files*
- Checking things out of a repository
  - cvs checkout *modules*

Compiler

- The C, C++, and Fortran compilers typically are installed in `/opt/SUNWspro/bin`
- Common tools like make are found in `/usr/ccs/bin`

GNU Tools

- Almost all GNU Tools are available as Solaris packages
- Some important tools like Perl, gzip, and Apache are automatically installed
- Other GNU-Tools can be found on the Solaris Companion CD
- GNU Tools can also be downloaded from

```
http://www.sunfreeware.com/
```

SunPCi™ and SunPCi-II

- Most workstations can be equipped with a SunPCi card.
  - This is a complete PC on a PCi card, running Windows (98, ME, NT, 2000, XP).
- Here are some hints for the operation of this card:
  Encapsulate /opt/SUNWspci2/bin/sunpci into a small script that sets some environment variables:
  setenv NVL_INTERFACE=hme0 (If SunPCi should use the hme0 network interface)
  setenv LANG=C
  setenv KBCP=850 (Example: German keyboard and codepage)
  setenv KBTYPE=GR (Remember MS-DOS ?)
  Make the C: image writeable not only for root, but for each user on the workstation: chmod 666. When a user starts sunpci for the first time, a subdirectory pc is created in the HOME directory of the user. To make the user work with the global C: image, interrupt the following creation of a user-specific image and edit the sunpci.ini file to point to the global image, as in /pc/C.diskimage,for example.

StarOffice Software

- If StarOffice software is installed as "root" with setup /net, all users can choose a "workstation installation" during their individual install
- A workstation installation only copies 1 Mbyte of data into the home directory of the user and otherwise uses the central installation
- If StarOffice software is installed in /opt/Office60, a user can start his or her workstation installation by calling /opt/Office60/program/soffice.

Netscape Navigator™ Browser

- Netscape Navigator V4.7 is used as the standard web browser
  - It is installed in `/usr/dt/bin/netscape`
- Solaris 9 also offers the Mozilla-based version 7 of Netscape Navigator
  - This version can be found in `/usr/dt/appconfig/SUNWns/netscape`


Hint Collection

Top Hints

- C-Shell
  - Define `set filec` and `set history=100 savehist=50` in `~/.cshrc`.
    - The shell will complete file names and directories once you hit the ESC key, and history lists the last 100 commands
  - With `!13`, command #13 can be re-issued
  - `!$` can be used as a variable for the last word of the former command
    - e.g. `mkdir /export/home/demo/test`, then `cd !$`
  - `tcsh` is available
    - allows editing the command line and browsing the history using the cursor keys.
- Process control
  - ps -ef | grep <search text> and then kill <PID> to search and kill a process can be replaced with pkill <search text>
  - kill -9 -1 terminates all processes belonging to the current user

Displaying Hardware Configuration

- To display all information on the hardware configuration of the machine, use

```
prtdiag -v
```

  - This displays the number and type of CPUs, RAM, extension cards, etc

Displaying Device Configuration

- To list the hierarchical device tree use

```
prtconf
```

- `/etc/path_to_inst` contains mappings between physical devices and logical drivers
  - This file is re-written during a boot -r or devfsadm
  - Editing `/etc/path_to_inst` manually can leave a system unbootable

Displaying Hardware Configuration

- Sun Explorer Software
  - This tool can be downloaded from SunSolve

```
http://sunsolve.sun.com/pub-cgi/show.pl?target=explorer/explorer
```

  - It collects all important system information
  - It can be used to prepare service calls, backup configuration data, snapshot configurations, etc.

Analysis of Runtime Data

- `time` (0,1 sec. resolution)
  `ptime` (1 msec resolution)
  - These commands are used to fine the time taken by a command to execute
  - The result is divided into user (=application) time, system (=OS) time, and elapsed time.
- `prstat`
  - The `prstat` utility iteratively examines all active processes on the system and reports statistics based on the selected output mode and sort order
  - It is similar to `top` on other systems

Analysis of Runtime Data - cont.

Analysis of Runtime Data - cont.

- `ps`
  - The `ps` command prints information about active processes
  - It is often used as `ps -ef` to give a full listing of all processes
- `vmstat`
  - vmstat reports virtual memory statistics regarding kernel thread, virtual memory, disk, trap, and CPU activity
  - `vmstat -S n` displays the statistics every *n* seconds
  - mpstat prints similar information, one line per processor

vmstat output
free: Free memory
page pi/po & si/so: Paging activity, measured in memory pages/second. Moderate paging activity is going on all the time and should not be considered harmful until combined with swapping activity (si/so greater than zero), which causes I/O to the swap device
cpu us: User-time. This percentage is consumed by applications
cpu sy: System-time. This percentage is consumed by Solaris
cpu id: Idle-time

Analysis of Runtime Data - cont.

- iostat
  - The iostat utility iteratively reports terminal, disk, and tape I/O activity, as well as CPU utilization
  - The first line of output is for all time since boot
  - Each subsequent line is for the prior interval only
  - Useful options include
    - -x show extended data
    - -t numbers of characters writen to terminals per sec
    - -c % of user and system time waiting for I/O
    - -P reports per-partition statistics
    - -E reports errors on devices

Graphical runtime data displays

- sdtperfmeter
  - Draws the vmstat-data as a bar or line chart
  - A minimized version of this tool is displayed in the CDE front panel
  - Prints cumulated CPU and I/O load on multi processor machines
- xcpustate
  - Prints bar chart for each CPU and disk in a machine
  - Available on the Companion CD
- sdtprocess
  - Lists all processes in a sorted way
  - It is possible to look further into process properties, terminate processes, and so on.

32-Bit or 64-Bit?

- `isainfo -kv`
  - Displays if running kernel is 32-bit or 64-bit
  - The 64-bit kernel is used automatically on modern Sun hardware
  - On UltraSPARC II based systems, he 32-bit kernel can be loaded explicitly
    - boot kernel/unix at the OK prompt
  - With UltraSPARC III systems, only 64-bit kernels are available

Debugging - cont.

- truss *command*
  - Lists all system calls that an application calls during its execution
  - Useful for finding where a command is searching for libraries or other files
- snoop
  - snoop prints all packets that are sent through a network interface

```
snoop -d hme0 host smith
```

  - monitors traffic through "`hme0`" and displays anything to or from the machine called "`smith`"

Debugging processes

- pstack PID
  - Prints the last function calls of a process
- pldd PID
  - Prints a list of shared libraries
- pmap -x PID
  - Prints the memory usage of the process and its modules