UNIX Administration Course

Copyright 1999 by Ian Mapleson BSc.

Version 1.0

mapleson@gamers.org
Tel: (+44) (0)1772 893297
Fax: (+44) (0)1772 892913
WWW: http://www.futuretech.vuurwerk.nl/

Detailed Notes for Day 3 (Part 2)

UNIX Fundamentals: Organising a network with a server.

This discussion explains basic concepts rather than detailed ideas such as specific ’topologies’ to use with large networks, or how to organise complex distributed file systems, or subdomains and address spaces - these are more advanced issues which most admins won’t initially have to deal with, and if they do then the tasks are more likely to be done as part of a team.

The SGI network in Ve24 is typical a modern UNIX platform in how it is organised. The key aspects of this organisation can be summarised as follows:

- A number of client machines and a server are connected together using a hub (24-port in this case) and a network comprised of 10Mbit Ethernet cable (100Mbit is more common in modern systems, with Gigabit soon to enter the marketplace more widely).

- Each client machine has its own unique identity, a local disk with an installed OS and a range of locally installed application software for use by users.

- The network has been configured to have its own subdomain name of a form that complies with the larger organisation of which it is just one part (UCLAN).

- The server has an external connection to the Internet.

- User accounts are stored on the server, on a separate external disk. Users who login to the client machines automatically find their own files available via the use of the NFS service.

- Users can work with files in their home directory (which accesses the server’s external disk across the network) or use the temporary directories on a client machine’s local disk for better performance.

- Other directories are NFS mounted from the server in order to save disk space and to centralise certain services (eg. /usr/share, /var/mail, /var/www).

- Certain aspects of the above are customised in places. Most networks are customised in certain ways depending on the requirements of users and the decisions taken by the admin and management. In this case, specifics include:

    - Some machines have better hardware internals, allowing for software installation setups that offer improved user application performance and services, eg. bigger disk permits /usr/share to be local instead of NFS-mounted, and extra vendor software, shareware and freeware can be installed.

- The admin’s account resides on an admin machine which is effectively also a client, but with minor modifications, eg. tighter security with respect to the rest of the network, and the admin’s personal account resides on a disk attached to the admin machine. NFS is used to export the admin’s home account area to the server and all other clients; custom changes to the admin’s account definition allows the admin account to be treated just like any other user account (eg. accessible from within /home/staff).

- The server uses a Proxy server in order to allow the client machines to access the external connection to the Internet.

- Ordinary users cannot login to the server, ensuring that the server’s resources are reserved for system services instead of running user programs. Normally, this would be a more important factor if the server was a more powerful system than the clients (typical of modern organisations). In the case of the Ve24 network though, the server happens to have the same 133MHz R4600PC CPU as the client machines. Staff can login to the server however - an ability based on assumed privilege.

- One client machine is using a more up-to-date OS version (IRIX 6.5) in order to permit the use of a ZIP drive, a device not fully supported by the OS version used on the other clients (IRIX 6.2). ZIP drives can be used with 6.2 at the command-line level, but the GUI environment supplied with 6.2 does not fully support ZIP devices. In order to support 6.5 properly, the client with the ZIP drive has more memory and a larger disk (most of the clients have 549MB system disks - insufficient to install 6.5 which requires approximately 720MB of disk space for a default installation).

- etc.

This isn’t a complete list, but the above are the important examples.

Exactly how an admin configures a network depends on what services are to be provided, how issues such as security and access control are dealt with, Internet issues, available disk space and other resources, peripherals provided such as ZIP, JAZ, etc., and of course any policy directives decided by management.

My own personal ethos is, in general, to put users first. An example of this ethos in action is that /usr/share is made local on any machine which can support it - accesses to such a local directory occur much faster than across a network to an NFS-mounted /usr/share on a server. Thus, searching for man pages, accessing online books, using the MIDI software, etc. is much more efficient/faster, especially when the network or server is busy.

NFS Issues.

Many admins will make application software NFS-mounted, but this results in slower performance (unless the network is fast and the server capable of supplying as much data as can be handled by the client, eg. 100Mbit Ethernet, etc.) However, NFS-mounted application directories do make it easier to manage software versions, updates, etc. Traditional client/server models assume applications are stored on a server, but this is an old ethos that was designed without any expectation that the computing world would eventually use very large media files, huge applications, etc. Throwing application data across a network is a ridiculous waste of bandwidth and, in my opinion, should be avoided where possible (this is much more important for slower networks though, eg. 10Mbit).

In the case of the Ve24 network, other considerations also come into play because of hardware-related factors, eg. every NFS mount point employed by a client system uses up some memory which is needed to handle the operational overhead of dealing with accesses to that mount point. Adding more mount points means using more memory on the client; for an Indy with 32MB RAM, using as many as a dozen mount points can result in the system running out of memory (I tried this in order to offer more application software on the systems with small disks, but 32MB RAM isn’t enough to support lots of NFS-mounted directories, and virtual memory is not an acceptable solution). This is a good example of how system issues should be considered when deciding on the hardware specification of a system. As with any computer, it is unwise to equip a UNIX system with insufficient resources, especially with respect to memory and disk space.

Network Speed.

Similarly, the required speed of the network will depend on how the network will be used. What applications will users be running? Will there be a need to support high-bandwidth data such as video conferencing? Will applications be NFS-mounted or locally stored? What kind of system services will be running? (eg. web servers, databases, image/document servers, etc.) What about future expansion? All these factors and more will determine whether typical networking technologies such as 10Mbit, 100Mbit or Gigabit Ethernet are appropriate, or whether a different networking system such as ATM should be used instead. For example, MPC uses a fast-switching high-bandwidth network due to the extensive use of data-intensive applications which include video editing, special effects, rendering and animation.

After installation, commands such as netstat, osview, ping and ttcp can be used to monitor network performance. Note that external companies, and vendor suppliers, can offer advice on suggested system topologies. For certain systems (eg. high-end servers), specific on-site consultation and analysis may be part of the service.

Storage.

Deciding on appropriate storage systems and capacities can be a daunting task for a non-trivial network. Small networks such as the SGI network I run can easily be dealt with simply by ensuring that the server and clients all have large disks, that there is sufficient disk space for user accounts, and a good backup system is used, eg. DDS3 DAT. However, more complex networks (eg. banks, commercial businesses, etc.) usually need huge amounts of storage space, use very different types of data with different requirements (text, audio, video, documents, web pages, images, etc.), and must consider a whole range of issues which will determine what kind of storage solution is appropriate, eg.:

- preventing data loss,
- sufficient data capacity with room for future expansion,
- interupt-free fast access to data,

- failure-proof (eg. backup hub units/servers/UPS),
- etc.

A good source of advice may be the vendor supplying the systems hardware, though note that 3rd-party storage solutions can often be cheaper, unless there are other reasons for using a vendor-sourced storage solution (eg. architectural integration).

See the article listed in reference [1] for a detailed discussion on these issues.

Setting up a network can thus be summarised as follows:

- Decide on the desired final configuration (consultation process, etc.)
- Install the server with a default installations of the OS. Install the clients with a default or expanded/customised configuration as desired.
- Construct the hardware connections.
- Modify the relevant setup files of a single client and the server so that one can rlogin to the server from the client and use GUI-based tools to perform further system configuration and administration tasks.
- Create, modify or install the files necessary for the server and clients to act as a coherent network, eg. /etc/hosts, .rhosts, etc.
- Setup other services such as DNS, NIS, etc.
- Setup any client-specific changes such as NFS mount points, etc.
- Check all aspects of security and access control, eg. make sure guest accounts are blocked if required, all client systems have a password for the root account, etc. Use any available FAQ (Frequently Asked Questions) files or vendor-supplied information as a source of advice on how to deal with these issues. Very usefully, IRIX 6.5 includes a high-level tool for controlling overall system and network security - the tool can be (and normally is) accessed via a GUI interface.
- Begin creating group entries in /etc/group ready for user accounts, and finally the user accounts themselves.
- Setup any further services required, eg. Proxy server for Internet access.
- etc.

The above have not been numbered in a rigid order since the tasks carried out after the very first step can usually be performed in a different order without affecting the final configuration. The above is only a guide.

Quotas.

Employing disk quotas is a practice employed by most administrators as a means of controlling disk space usage by users. It is easy to assume that a really large disk capacity would mean an admin need not bother with quotas, but unfortunately an old saying definitely holds true: "Data will expand to fill the space available."

Users are lazy where disk space is concerned, perhaps because it is not their job to manage the system as a whole. If quotas are not present on a system, most users simply don’t bother deleting unwanted files. Alternatively, the quota management software can be used as an efficient disk accounting system by setting up quotas for a file system without using limit enforcement.

IRIX employs a quota management system that is common amongst many UNIX variants. Examining the relevant commands (consult the ’SEE ALSO’ section from the ’quotas’ man page), IRIX’s quota system appears to be almost identical to that employed by, for example, HP-UX (Hewlett Packard’s UNIX OS). There probably are differences between the two implementations, eg. issues concerning supported operations on particular types of file system, but in this case the quota system is typical of the kind of OS service which is very similar or identical across all UNIX variants. An important fact is that the quota software is part of the overall UNIX OS, rather than some hacked 3rd-party software addon.

Quota software allows users to determine their current disk usage, and enables an admin to monitor available resources, how long a user is over their quota, etc. Quotas can be used not only to limit the amount of available disk space a user has, but also the number of files (inodes) which a user is permitted to create.

Quotas consist of soft limits and hard limits. If a user’s disk usage exceeds the soft limit, a warning is given on login, but the user can still create files. If disk usage continues to increase, the hard limit is the point beyond which the user will not be able to use any more disk space, at least until the usage is reduced so that it is sufficiently below the hard limit once more.

Like most system services, how to setup quotas is explained fully in the relevant online book, "IRIX Admin: Disks and Filesystems". What follows is a brief summary of how quotas are setup under IRIX. Of more interest to an admin are the issues which surround quota management - these are discussed shortly.

To activate quotas on a file system, an extra option is added to the relevant entry in the /etc/fstab file so that the desired file system is set to have quotas imposed on all users whose accounts reside on that file system. For example, without quotas imposed, the relevant entry in yoda’s /etc/fstab file looks like this:

```
/dev/dsk/dks4d5s7 /home xfs rw 0 0
```

With quotas imposed, this entry is altered to be:

```
/dev/dsk/dks4d5s7 /home xfs rw,quota 0 0
```

Next, the quotaon command is used to activate quotas on the root file system. A reboot causes the quota software to automatically detect that quotas should be imposed on /home and so the quota system is turned on for that file system.

The repquota command is used to display quota statistics for each user. The edquota command is used to change quota values for a single user, or multiple users at once. With the -i option, edquota can also read in quota information from a file, allowing an admin to set quota limits for many users with a single command. With the -e option, repquota can output the current quota statistics to a file in a format that is suitable for use with edquota’s -i option.

Note: the editor used by edquota is vi by default, but an admin can change this by etting an environment variable called ’EDITOR’, eg.:

```
     setenv EDITOR jot -f
```

The -f option forces jot to run in the foreground. This is necessary because the editor used by edquota must run in the foreground, otherwise edquota will simply see an empty file instead of quota data.

Ordinary users cannot change quota limits.

Quota Management Issues.

Most users do not like disk quotas. They are perceived as the information equivalent of a straitjacket. However, quotas are usually necessary in order to keep disk usage to a sensible level and to maintain a fair usage amongst all users.

As a result, the most important decision an admin must make regarding quotas is what limit to actually set for users, either as a whole or individually.

The key to amicable relations between an admin and users is flexibility, eg. start with a small to moderate limit for all (eg. 20MB). If individuals then need more space, and they have good reason to ask, then an admin should increase the user’s quota (assuming space is available).

Exactly what quota to set in the first instance can be decided by any sensible/reasonable schema. This is the methodology I originally adopted:

- The user disk is 4GB. I don’t expect to ever have more than 100 users, so I set the initial quota to 40MB each.

In practice, as expected, some users need more, but most do not. Thus, erring on the side of caution while also being flexible is probably the best approach.

Today, because the SGI network has a system with a ZIP drive attatched, and the SGIs offer reliable Internet access to the WWW, many students use the Ve24 machines solely for downloading data they need, copying or moving the data onto ZIP for final transfer to their PC accounts, or to a machine at home. Since the ZIP drive is a 100MB device, I altered the quotas to 50MB each, but am happy to change that to 100MB if anyone needs it (this allows for a complete ZIP image to be downloaded if required), ie. I am tailoring quota limits based on a specific hardware-related user service issue.

If a user exceeds their quota, warnings are given. If they ask for more disk space, an admin would normally enquire as to whether the user genuinely needs more space, eg.:

- Does the user have unnecessary files lying around in their home directory somewhere? For example, movie files from the Internet, unwanted mail files, games files, object files or core dump files left over from application development, media files created by ’playing’ with system tools (eg. the digital camera). What about their Netscape cache? Has it been set to too

> high a value? Do they have hidden files they’re not aware of, eg. .capture.tmp.* directories, capture.mv files, etc.? Can the user employ compression methods to save space? (gzip, pack, compress)

If a user has removed all unnecessary files, but is still short of space, then unless there is some special reason for not increasing their quota, an admin should provide more space. Exceptions could include, for example, a system which has a genuine overall shortage of storage space. In such a situation, it is common for an admin to ask users to compress their files if possible, using the ’gzip’, ’compress’ or ’pack’ commands. Users can use tar to create archives of many files prior to compression. There is a danger with asking users to compress files though: eventually, extra storage has to be purchased; once it has been, many users start uncompressing many of the files they earlier compressed. To counter this effect, any increase in storage space being considered should be *large*, say an order of magnitude, or at the very least a factor of 3 or higher (I’m a firm believer in future-proofing).

Note that the find command can be used to locate files which are above a certain size, eg. those that are particularly large or in unexpected places. Users can use the du command to examine how much space their own directories and files are consuming.

Note: if a user exceeds their hard quota limit whilst in the middle of a write operation such as using an editor, the user will find it impossible to save their work. Unfortunately, quitting the editor at that point will lose the contents of the file because the editor will have opened a file for writing already, ie. the opened file will have zero contents. The man page for quotas describes the problem along with possible solutions that a user can employ:

> "In most cases, the only way for a user to recover from over-quota conditions is to abort whatever activity is in progress on the filesystem that has reached its limit, remove sufficient files to bring the limit back below quota, and retry the failed program.
>
> However, if a user is in the editor and a write fails because of an over quota situation, that is not a suitable course of action. It is most likely that initially attempting to write the file has truncated its previous contents, so if the editor is aborted without correctly writing the file, not only are the recent changes lost, but possibly much, or even all, of the contents that previously existed.
>
> There are several possible safe exits for a user caught in this situation. He can use the editor ! shell escape command (for vi only) to examine his file space and remove surplus files. Alternatively, using csh, he can suspend the editor, remove some files, then resume it. A third possibility is to write the file to some other filesystem (perhaps to a file on /tmp) where the user’s quota has not been exceeded. Then after rectifying the quota situation, the file can be moved back to the filesystem it belongs on."

It is important that users be made aware of these issues if quotas are installed. This is also another reason why I constantly remind users that they can use /tmp and /var/tmp for temporary tasks. One machine in Ve24 (Wolfen) has an extra 549MB disk available which any user can write to, just in case a particularly complex task requiring alot of disk space must be carried out, eg. movie file processing.

Naturally, an admin can write scripts of various kinds to monitor disk usage in detailed ways, eg. regularly identify the heaviest consumers of disk resources; one could place the results into a regularly updated file for everyone to see, ie. a publicly readable "name and shame" policy (not a method I’d use unless absolutely necessary, eg. when individual users are abusing the available space for downloading game files).

UNIX Fundamentals: Installing/removing internal/external hardware.

As explained in this course’s introduction to UNIX, the traditional hardware platforms which run UNIX OSs have a legacy of top-down integrated design because of the needs of the market areas the systems are sold into.

Because of this legacy, much of the toil normally associated with hardware modifications is removed. To a great extent, an admin can change the hardware internals of a machine without ever having to be concerned with system setup files. Most importantly, low-level issues akin to IRQ settings in PCs are totally irrelevant with traditional UNIX hardware platforms. By traditional I mean the long line of RISC-based systems from the various UNIX vendors such as Sun, IBM, SGI, HP, DEC and even Intel. This ease of use does not of course apply to ordinary PCs running those versions of UNIX which can be used with PCs, eg. Linux, OpenBSD, FreeBSD, etc.; for this category of system, the OS issues will be simpler (presumably), but the presence of a bottom-up-designed PC hardware platform presents the usual problems of compatible components, device settings, and other irritating low-level issues.

This discussion uses the SGI Indy as an example system. If circumstances allow, a more up-to-date example using the O2 system will also be briefly demonstrated in the practical session. Hardware from other UNIX vendors will likely be similar in terms of ease-of-access and modification, though it has to be said that SGI has been an innovator in this area of design.

Many system components can be added to, or removed from a machine, or swapped between machines, without an admin having to change system setup files in order to make the system run smoothly after any alterations. Relevant components include:

- Memory units,
- Disk drives (both internal and external),
- Video or graphics boards that do not alter how the system would handle relevant processing operations.
- CPU subsystems which use the same instruction set and hardware-level initialisation libraries as are already installed.
- Removable storage devices, eg. ZIP, JAZ, Floptical, SyQuest, CDROM, DVD (where an OS is said to support it), DAT, DLT, QIC, etc.
- Any option board which does not impact on any aspect of existing system operation not related to the option board itself, eg. video capture, network expansion (Ethernet, HiPPI, TokenRing, etc.), SCSI expansion, PCI expansion, etc.

Further, the physical layout means the admin does not have to fiddle with numerous cables and wires. The only cables present in Indy are the two short power supply cables, and the internal SCSI device ribbon cable with its associated power cord. No cables are present for graphics boards, video options, or other possible expansion cards. Some years after the release of the Indy, SGI’s O2 design allows one to perform all these sorts of component changes without having to fiddle with any cables or screws at all (the only exception being any PCI expansion, which most O2 users will probably never use anyway).

This integrated approach is certainly true of Indy. The degree to which such an ethos applies to other specific UNIX hardware platforms will vary from system to system. I should imagine systems such as Sun's Ultra 5, Ultra 10 and other Ultra-series workstations are constructed in a similar way.

One might expect that any system could have a single important component replaced without affecting system operation to any great degree, even though this is usually not the case with PCs, but it may come as a far greater surprise that an entire set of major internal items can be changed or swapped from one system to another without having to alter configuration files at all.

Even when setup files do have to be changed, the actual task normally only involves either a simple reinstall of certain key OS software sub-units (the relevant items will be listed in accompanying documentation and release notes), or the installation of some additional software to support any new hardware-level system features. In some cases, a hardware alteration might require a software modification to be made from miniroot if the software concerned was of a type involved in normal system operation, eg. display-related graphics libraries which controlled how the display was handled given the presence of a particular graphics board revision.

The main effect of this flexible approach is that an admin has much greater freedom to:

- modify systems as required, perhaps on a daily basis (eg. the way my external disk is attatched and removed from the admin machine every single working day),

- experiment with hardware configurations, eg. performance analysis (a field I have extensively studied with SGIs [2]),

- configure temporary setups for various reasons (eg. demonstration systems for visiting clients),

- effect maintenance and repairs, eg. cleaning, replacing a power supply, etc.

All this without the need for time-consuming software changes, or the irritating necessity to consult PC-targeted advice guides about devices (eg. ZIP) before changes are made.

Knowing the scope of this flexibility with respect to a system will allow an admin to plan tasks in a more efficient manner, resulting in better management of available time.

An example of the above with respect to the SGI Indy would be as follows (this is an imaginary demonstration of how the above concepts could be applied in real-life):

- An extensive component swap between two indys, plus new hardware installed.

Background information:

CPUs.

All SGIs use a design method which involves supplying a CPU and any necessary secondary cache plus interface ASICs on a ’daughterboard’, or ’daughtercard’. Thus, replacing a CPU merely involves changing the daughtercard, ie. no fiddling with complex CPU insertion sockets, etc. Daughtercards in desktop systems can be replaced in seconds, certainly no more than a minute or two.

The various CPUs available for Indy can be divided into two categories: those which support everything up to and including the MIPS III instruction set, and those which support all these plus the MIPS IV instruction set.

The R4000, R4600 and R4400 CPUs all use MIPS III and are initialised on bootup with the same low-level data files, ie. the files stored in /var/sysgen. This covers the following CPUs:

```
100MHz R4000PC (no L2)
100MHz R4000SC (1MB L2)
100MHz R4600PC (no L2)
133MHz R4600PC (no L2)
133MHz R4600SC (512K L2)
100MHz R4400SC (1MB L2)
150MHz R4400SC (1MB L2)
175MHz R4400SC (1MB L2)
200MHz R4400SC (1MB L2)
```

Thus, two Indys with any of the above CPUs can have their CPUs swapped without having to alter system software.

Similarly, the MIPS IV CPUs:

```
150MHz R5000PC (no L2)
150MHz R5000SC (512K L2)
180MHz R5000SC (512K L2)
```

can be treated as interchangeable between systems in the same way.

The difference between an Indy which uses a newer vs. older CPU is that the newer CPUs require a more up-to-date version of the system PROM chip to be installed on the motherboard (a customer who orders an upgrade is suppled with the newer PROM if required).

Video/Graphics Boards.

Indy can have three different boards which control display output:

```
8bit XL
24bit XL
24bit XZ
```

8bit and 24bit XL are designed for 2D applications. They are identical except for the addition of more VRAM to the 24bit version. XZ is designed for 3D graphics and so requires a slightly different installation of software graphics libraries to be installed in order to permit proper use. Thus, with respect to the XL version, an 8bit XL card can be swapped with a 24bit XL card with no need to alter system software.

Indy can have two other video options:

- IndyVideo (provides video output ports as well as extra input ports),

- CosmoCompress (hardware-accelerate MJPEG video capture board).

IndyVideo does not require the installation of any extra software in order to be used. CosmoCompress does require some additional software to be installed (CosmoCompress compression API and libraries). Thus, IndyVideo could be installed without any post-installation software changes. swmgr can be used to install the CosmoCompress software after the option card has been installed.

Removable Media Devices.

As stated earlier, no software modifications are required, unless specifically stated by the vendor. Once a device has its SCSI ID set appropriately and installed, it is recognised automatically and a relevant icon placed on the desktop for users to exploit. Some devices may require a group of DIP switches to be configured on the outside of the device, but that is all (settings to use for a particular system will be found in the supplied device manual). The first time I used a DDS3 DAT drive (Sony SDT9000) with an Indy, the *only* setup required was to set four DIP switches on the underside of the DAT unit to positions appropriate for use with an SGI (as detailed on the first page of the DAT manual). Connecting the DAT unit to the Indy, booting up and logging in, the DAT was immediately usable (icon available, etc.) No setup files, no software to install, etc. The first time I used a 32X CDROM (Toshiba CD-XM-6201B) not even DIP switches had to be set.

System Disks, Extra Disks.

Again, installed disks are detected automatically and the relevant device files in /dev initialised to be treated as the communication points with the devices concerned. After bootup, the fx, mkfs and mount commands can be used to configure and mount new disks, while disks which already have a valid file system installed can be mounted immediately. GUI tools are available for performing these actions too.

Thus, consider two Indys:

System A	System B
200MHz R4400SC	100MHz R4600PC
24bit XL	8bit XL
128MB RAM	64MB RAM
2GB disk	1GB disk
IRIX 6.2	IRIX 6.2

Suppose an important company visitor is expected the next morning at 11am and the admin is asked to quickly prepare a decent demonstration machine, using a budget provided by the visiting company to cover any changes required (as a gift, any changes can be permanent).

The admin orders the following extra items for next-day delivery:

- A new 4GB SCSI disk (Seagate Barracuda 7200rpm)
- IndyVideo board
- Floptical drive
- ZIP drive

- 32X Toshiba CDROM (external)
- DDS3 Sony DAT drive (external)

The admin decides to make the following changes (Steps 1 and 2 are carried out immediately; in order to properly support the ZIP drive, the admin needs to use IRIX 6.5 on B. The support contract means the CDs are already available.):

1. Swap the main CPU, graphics board and memory components between systems A and B.

2. Remove the 1GB disk from System B and install it as an option disk in System A. The admin uses fx and mkfs to redine the 1GB disk as an option drive, deciding to use the disk for a local /usr/share partition (freeing up perhaps 400MB of space from System A's 2GB disk).

3. The order arrives the next morning at 9am (UNIX vendors usually use couriers such as Fedex and DHL, so deliveries are normally very reliable). The 4GB disk is installed into System B (empty at this point) and the CDROM connected to the external SCSI port (SCSI ID 3). The admin then installs IRIX 6.5 onto the 4GB disk, a process which takes approximately 45 minutes. The system is powered down ready for the final hardware changes.

4. The IndyVideo board is installed in System B (sits on top of the 24bit XL board, 2 or 3 screws involved, no cables), along with the internal Floptical drive above the 4GB disk (SCSI ID set to 2). The DAT drive (SCSI ID set to 4) is daisy chained to the external CDROM. The ZIP drive is daisy chained to the DAT (SCSI ID 5 by default selector, terminator enabled). This can all be done in less than five minutes.

5. The system is rebooted, the admin logs in as root. All devices are recognised automatically and icons for each device (ZIP, CDROM, DAT, Floptical) are immediately present on the desktop and available for use. Final additional software installations can begin, ready for the visitor's arrival. An hour should be plenty of time to install specific application(s) or libraries that might be required for the visit.

I am confident that steps 1 and 2 could be completed in less than 15 minutes. Steps 3, 4 and 5 could be completed in less little more than an hour. Throughout the entire process, *no* OS or software changes have to be made to either System A, or to the 6.5 OS installed on System B's new 4GB after initial installation (ie. the ZIP, DAT and Floptical were not attatched to System B when the OS was installed, but they are correctly recognised by the *default* 6.5 OS when the devices are added afterwards).

If time permits and interest is sufficient, almost all of this example can be demonstrated live (the exception is the IndyVideo board; such a board is not available for use with the Ve24 system at the moment).

How does the above matter from an admin's point of view? The answer is confidence and lack of stress. I could tackle a situation such as described here in full confidence that I would not have to deal with any matters concerning device drivers, interupt addresses, system file modifications, etc. Plus, I can be sure the components will work perfectly with one another, constructed as they are as part of an integrated system design. In short, this integrated approach to system design makes the admin's life substantially easier.

The Visit is Over.

Afterwards, the visitor donates funds for a CosmoCompress board and an XZ board set. Ordered that day, the boards arrive the next morning. The admin installs the CosmoCompress board into System B (2 or 3 more screws and that’s it). Upon bootup, the admin installs the CosmoCompress software from the supplied CD with swmgr. With no further system changes, all the existing supplied software tools (eg. MediaRecorder) can immediately utilise the new hardware compression board.

The 8bit XL board is removed from System A and replaced with the XZ board set. Using inst accessed via miniroot, the admin reinstalls the OS graphics libraries so that the appropriate libraries are available to exploit the new board. After rebooting the system, all existing software written in OpenGL automatically runs ten times faster than before, without modification.

Summary.

Read available online books and manual pages on general hardware concepts thoroughly.

Get to know the system - every machine will either have its own printed hardware guide, or an equivalent online book.

Practice hardware changes before they are required for real.

Consult any Internet-based information sources, especially newsgroup posts, 3rd-party web sites and hardware-related FAQ files.

When performing installations, follow all recommended procedures, eg. use an anti-static strap to eliminate the risk of static discharge damaging system components (especially important for handling memory items, but also just as relevant to any other device).

Construct a hardware maintenance strategy for cleansing and system checking, eg. examine all mice on a regular basis to ensure they are dirt-free, use an air duster once a month to clear away accumulated dist and grime, clean the keyboards every two months, etc.

Be flexible. System management policies are rarely static, eg. a sudden change in the frequency of use of a system might mean cleansing tasks need to be performed more often, eg. cleaning monitor screens.

If you’re not sure what the consequences of an action might be, call the vendor’s hardware support service and ask for advice. Questions can be extremely detailed if need be - this kind of support is what such support services are paid to offer, so make good use of them.

Before making any change to a system, whether hardware or software, inform users if possible. This is probably more relevant to software changes (eg. if a machine needs to be rebooted, use ’wall’ to notify any users logged onto the machine at the time, ie. give them time to log off; if they don’t, go and see why they haven’t), but giving advance notice is still advisable for hardware changes too, eg. if a system is being taken away for cleaning and reinstallation, a user may want to retrieve files from /var/tmp prior to the system’s removal, so place a notice up a day or so beforehand if possible.

References:

1. "Storage for the network", Network Week, Vol4 No.31, 28th April 1999, pp. 25 to 29, by Marshall Breeding.

2. SGI General Performance Comparisons:

   `http://www.futuretech.vuurwerk.nl/perfcomp.html`