Salvage Server Project Report 1:

Selecting Operating Systems for and Upgrading Older Computers

Produced by the Free Range Salvage Server Project, September 2003
email: fraw@gn.apc.org web: http://www.fraw.org.uk/ssp/

Computer systems are designed, produced and marketed to have a limited operational life. Although the hardware may last four or five years before total failure, the software used on it may only be supported for two to three years. When regenerating any computer equipment, it is important to select the right operating system to avoid problems with usability of the system. This report looks at the issues in installing software on older equipment.

The importance of Operating Systems

Restoring old computer systems can be fairly simple. There is a lot of scrap parts to repair and upgrade older computers. Bought in bulk, often the hardware costs more to transport than to buy. The great problem is software. Unless you can get software to install on the system, and work properly, you will just have a box of electrical circuits, not a computer.

Proprietary computer software – which for PC computers means Microsoft operating systems – has a short service life. Microsoft fully revise their software every three to four years. When they do, computer manufacturers and computer sellers completely change their stock in order to support the model of Microsoft's new software. At this point, support for the older versions will diminish. Support for the previous predecessor may completely disappear.

Free software – the Gnu/Linux operating system – does not wholly abandon past hardware. This is because the software is developed incrementally – it is not wholly revised. The revision of the modules within the operating system will usually retain 'backwards compatibility' with the older functions of the operating system – meaning the newer version of the system may work equally well on older equipment. This backwards compatibility also means its possible to leave out certain parts of the system, and perhaps use the older ones, to make the system work on older computers with less system resources.

When deciding what to do with old equipment it's important to plan the use of operating systems before you embark on repairing and upgrading the equipment. This is because the choice of operating system may limit which of the old machines are usable, or how much upgrading the usable machines require.

Proprietary systems

The Salvage Server project does not support, generally, proprietary software – either Microsoft or Mackintosh. This is because it's just so damn difficult to keep working in the longer term. You can't get the software support for newer hardware in order to keep the computer running. It's also a big legal liability. It doesn't matter that your software is eight years old, if you are using it without a license you're still breaching copyright. This leaves you open to legal action to recover damages if you get found out. Curiously, this can also mean that old proprietary software can cost the same now as it did when it was first sold – people who need to use software, say *Window 98*, legally have to pay £70 to £80 for a new copy.

The table over the page gives details of the processor, memory and hard disk space for different operating systems. This lists the details for the *Windows* systems used on older equipment. However, it is also important to obtain additional software to make the system usable. There are also other important issues with regard to setting up networks:

- Using MS-DOS with Windows 3.1 requires hardware drivers fro network cards and other devices in order to connect to a network. Windows 3.11 has minimal networking capabilities, but does not work well on a network without additional network driver programs. Software is also difficult to find. Forget about proper Internet support too – no support for modern modems.
- Windows 95 has better networking capabilities, but lacks some features like password encryption that can create problems mixing with other OSs on networks.
- Windows 98/Windows ME works well on networks, and compatible software is easier to obtain.

Hardware Requirements of Different Operating Systems

Operating system	*Processor*	*RAM, MB*	*Hard disk, MB*
Windows:			
MS-DOS/Windows 3.11	'386	2 (8)	50
Windows 95	'486	16 (24)	150
Windows 98/ME	'486	24(64)	500
Windows 2000	PI/133 (PII/300)	128 (256)	1,024
Windows XP	PII/300 (PIII/500)	128 (256)	1,584
Gnu/Linux:			
Debian 3.0 Linux ('Woody')	'386	12 (64)	250 (850)
Linux-Mandrake 7.1	'386	24 (64)	400 (800)
Linux-Mandrake 9.1	P-I	64 (128)	400 (1024)
Slackware Linux 9.0	'386	16 (64)	100 (500)
Red Hat 6.2	'386	16 (48)	200 (750)
Red Hat 9.0	P-I (P-II)	64 (128)	850 (1400)
SuSE Linux 6.4	'386	32 (64)	400 (750)
SuSE Linux 8.2	P-I	64 (128)	600 (1800)

Figures given are for text-only installations. Graphical installation figures are given in brackets. These are only general figures, and hard disk space in particular can be manipulated to minimise installation size.

When obtaining software it is important to ensure that you get a copy of a license for the software – or buy the paper license and copy the software. Otherwise you're breaking the law and could be sued.

Gnu/Linux

Gnu/Linux, for the most part, does not have the same copyright restrictions as *Windows* – although some distributions like Red Hat or SuSE have some proprietary elements, such as the additional 'commercial' packages, or user-friendly graphical installation programs (e.g., SuSE's *YaST* program). The main benefit of Linux is that it works well across a range of hardware, and in most cases, it will work far better on older equipment than brand new equipment.

The critical factor in selecting a Linux distribution for the tasks you need to perform is to match hardware support with system requirements. For older equipment the two limiting factors are often memory size and hard disk size. But increasingly it's also the processor as some distributions abandon support for older hardware.

Different Linux distributions are optimised around particular types of use. There are many different types of Linux distribution in circulation, some masquerading as other systems – such as the *Smoothwall* firewall system. The main general purpose distributions are:

Red Hat Linux – The distribution that specialises in server applications.

Linux-Mandrake – General purpose distribution that provides a good desktop system.

SuSE Linux – General purpose distribution with a good desktop, but also good hardware support.

Slackware Linux – Distribution that specialises in providing support for older equipment, especially systems with minimal hardware.

Debian Linux – A general purpose distribution that has a good desktop set-up, but that also supports older hardware.

There is no reason why you have to use the latest Linux distribution when installing machines. The only time this is a problem is when you are installing a server that is connected to the Internet, or must be secure. In these situations the current distributions are required because you need the latest security patches to prevent abuse of the system. But in other cases, you can use older distributions. Often, you can pick older installations up for

nothing via Linux User Groups. As shown in the table on OS requirements, most older Linux distributions have lower hardware requirements than the comparable *Windows* system.

The other problem with some older distributions is that they require a little more knowledge of Linux systems in order to install and set them up. An example is Slackware Linux. If the machine can't boot a CD-ROM, you have to make a number of (at least three) boot floppy disks, depending upon the hardware configuration of the machine. It also has a very basic, text-menu installation process. Some distributions, such as SuSE Linux, have very good user-friendly installation programs. The problem with these systems is also that they can't be used on older equipment because these programs require 48MB to 64MB of RAM in order to copy lots of data into memory.

All Linux distributions install in roughly the same way. The different is usually the detail of the programs used to perform the installation. Most now use a single program to manage installation. Some boot into a Linux kernel running from memory that allows you to use standard utility programs to set-up the system for installation. You then install the software using another program. Even so, basic requirements such as partitioning and formatting the hard disks, selecting network addresses and setting passwords, require the same considerations no matter what distribution you use.

Upgrading memory

The limits of installation for most Linux distributions are easy to determine. These are usually given in the installation guide, or in 'release notes'. Before attempting to install you need to upgrade the system to meet these minimum levels. This presents a problem when using older computers. They nearly always require substantial upgrading.

'286 processor machines have no use. You can't install a easily usable Linux on them, and most of the parts such as the memory, is not interchangeable with other machines.

'386 and early '486 machines use '30-pin' memory. These come in 1 megabyte, 2 megabyte and 4 megabyte packages. This makes it possible to assemble between 4MB and 16MB of RAM, but

Creating Boot Disks

Most Pentium-I systems are able to boot directly from a CD-ROM. This simplifies installation because you just put in the CD, set the hardware BIOS settings to 'boot from CD', and the installation process starts.

On older machines, all '386s and nearly all '486s, you have to create a floppy disk from the installation CD. To do this you need a 'boot image' file. Conveniently these are around 1.44 megabytes (the same size as a floppy disk) in size. Usually the distribution has a file such as 'README' or 'BOOTING' to tell you which disk image to use.

When you locate the right disk image you need a program to create the floppy disk. But first, you must format the floppy disk for use with MS-DOS. The reason for this is that if there is just one bad block on the floppy disk, this process will not work. The floppy must be absolutely perfect before writing the boot floppy data to it. Also beware other unforeseen problems – such as using *Windows 2000* or *XP* to make boot disks. These don't use pure MS-DOS file systems and so sometimes boot disk creation fails.

If you are using *Windows* you need a program called `RAWRITE` – usually supplied with the Linux distribution. Copy the program and the disk image to the *Windows* C: drive and from the MS-DOS prompt enter the command –

```
C:> rawrite a: bootdisk.img
```

This instructs Windows to write the image file 'bootdisk.img' to floppy drive a:.

From Linux, you need to use the `dd` command. This transfers the image file from the CD to the floppy. You needn't mount the floppy disk to do this. So, assuming that the path to your CD drive is '/cdrom', you would issue a command like –

```
dd if=/cdrom/boot/disk.img of=/dev/fd0
```

This copies the image file from (`if=`)the CD (using whatever the correct path/file name is) to the (`of=`) floppy drive (/dev/fd0).

You can then test the boot disk by booting a machine with it. You may have to use this process more than once since some distributions have more than one boot disk (some require additional disks for kernel modules, like PCMCIA support on laptops). When you have booted your machine with the floppy disk, installation with the CD-ROM drive should proceed as normal.

Note that there are other options too. Using boot floppies you can initiate an installation over a local network (an 'NFS install') using the CD drive on another machine. This is useful when installing onto a machine that has no CD drive. However, it's a little more technical because you must 'export' a CD (or better still DVD because only one disk is required) drive from a working Linux machine. It is also much slower. For those who have the ability to set up SLIP or PPP via a serial port, it's even possible to instal via a serial port, although this will take many hours to complete!

finding 4MB packages is difficult so often you are restricted to 8MB. You can get 'stackers' to create larger blocks, but these are also difficult to find. For this reason these older machines are restricted in their use. However, some Linux-based applications, like the *Smoothwall* firewall system, can run on these older machines.

Older '486 machines and early Pentium-I machines use 72-pin memory. Organising 72-pin memory from a box of scrapped packages can be difficult because there are different types of package. In general, you can create memory configurations of between 8MB and 64MB. Enough to install some of the less demanding Linux distributions.

Some of the later Pentium-I machines used 168-pin DRAM memory packages. These have been the standard until the development of the later Pentium-III/Pentium-IV systems so there's a lot in circulation (although the early one work more slowly). These come in blocks of 32MB, 64MB or 128MB. Usually there are two or three slots, so memory configurations of 64MB to 256MB may be easily set up (although some early Pentium machines had limits on the amount of memory that could be installed). These systems are ideal to install some of the larger distributions, that give the greatest functionality on the desktop, because installing lots of memory is easy and cheap.

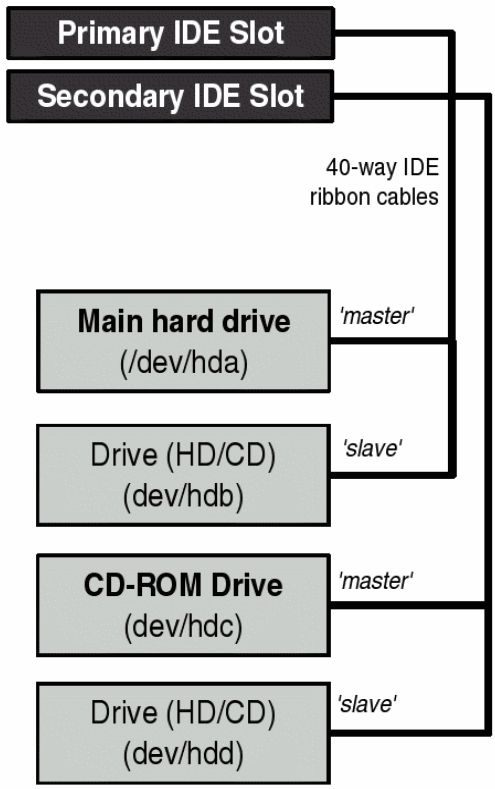
Finally, you should always try and install the most memory possible. The memory figures in the table earlier are the lowest practical level of memory. Ideally you should have twice these figures to get a comfortable speed of operation. Providing that the operating system is able to address the memory and use it, the extra memory will significantly speed up the execution of larger programs because less information will be buffered onto disk.

Upgrading disk drives

Having solved memory capacity problems, we move on to hard disk space. For text-based installations, such as those used for servers, the amount of disk space required is about half that of a graphical installation. However, even for a server, having a graphical installation can be useful as the graphical tools simplify the configuration and administration of the server for less experienced Linux users.

A server will also need an minimum one or two gigabytes of hard disk to allow them to be used for serving web sites, FTP file uploading, etc. Desktop installations, where users will be regularly using the system for graphics or work processing, may need more than this. So it's important to ensure that the disk capacity is enough for both the installation and use of the system.

IDE drive configuration



Most PCs use 'Integrated Drive Electronics' (IDE) disk drives (the main alternative, SCSI, isn't popular except in old servers). The IDE system allows up to four drives to be connected to the system (plus up to two more floppy drives on a single 34-way ribbon cable). These drives may be hard disks, CD, CD/RW DVD or Zip drives.

The drives are connected via two cables – the 'primary' and 'secondary' IDE cables – with two drives per cable. Small jumpers identify the drives as either the first 'master' drive, or the second 'slave' drive. The main hard disk is usually the 'primary master'. To allow the faster transfer of data between CD and disk, CD ROMs are usually on the 'secondary' cable and disk drives on the primary.

Most old computers come with a hard disk that is usually too small and too slow. This disk can be replaced with a larger hard disk to increase storage capacity. Alternately, a second hard drive can be connected to the system. The important thing to get right is the jumper connectors. When mixed on the same cable, one must be set to master, the other to slave. The positioning of the connectors varies between different types of drive. If the connectors are wrong it doesn't cause major damage, but the disks won't work (likewise if the cable is the wrong way around).

Machines older than the early Pentium-I series did not

have a CD-ROM fitted as standard, so you may have to add a CD-ROM drive. Sometimes this requires a little re-engineering of the case. Desktop cases are far harder to adapt than tower cases because desktops usually require special brackets to hold the CD drive in the drive bay.

If installing one or two extra drives you may have to get some new ribbon cables. The old cable may only have one connector, or there may only one cable with two connectors when you need two cables and three (the old drive, a new drive, and a CD drive). You may have to get some 'splitter' cables for the power connectors if there are not enough power plugs available from the power supply.

The aim should be to create enough disk space to store the operating system, and have enough space disk storage for the data that needs to be stored on the system. And although a CD drive isn't essential to install Linux, upgrading the CD ensures that the system installs and works far faster.

Partitioning the hard disks

Unlike *Windows* systems, the operating system need not be stored wholly on one disk drive. You can spread the different directories of the Linux system over two or three disk drives if necessary. Alternately, some versions of Linux, like Slackware's *Zip Slack*, can even be stored on a single removable Zip disk and booted when required.

One of the first tasks when installing any distribution is to create a partitioning scheme for the hard disk(s). Although different distributions have different looking systems for this, fundamentally it's the same process. You take your available hard disk space and divide it up to meet different needs. Some systems offer you an automated means of doing this. But on older computers, where disk space is short, you usually need to control how space is allocated.

An IDE hard disk can have up to four 'primary partitions'. Each of these is read as a single 'drive'. One of these primary partitions may in turn be configured as an special partition, to allow more 'extended' partitions to be configured within it. The primary partitions are numbered 1 to 4. Even if all of the primary partitions are not used, the extended partitions are numbered from 5.

On a small system you may only configure two or three partitions. But for systems where you need better security or reliability, you might set-up five or six. The box above lists the system directories which are commonly set up. The 'mount point' – the name of the directory – is the label that you apply to the partition. The 'device' name of the partition is the name of the disk drive plus the partition number. So if you set up the 'root' partition as the first partition, that will be `/dev/hda1`, and next (e.g., swap) will be `/dev/hda2`, and so on. You need to keep a note of these because later on you'll have to use both of these identities when performing system maintenance (but, if in doubt, the file `/etc/fstab` contains the details).

Linux requires two partitions to be created – the 'root' partition, and the swap partition. All other partitions are optional, and if not created, will be set up as ordinary directories within the root partition.

The 'root' partition should be big enough to hold the size of the installation, plus about 10% to 20% extra space to hold the logs and other files generated as part of ordinary system operation. The swap partition is usually set to be twice the size of the system RAM. At its simplest, partitioning requires that you subtract twice the size of the RAM from the total hard disk size, set up the root directory to be that size, and set the rest as a swap partition.

You set up other partitions for reasons of simplifying the operation of the system or to improve reliability. Most

Linux directories that are often set-up as partitions

name	*ID*	*use*
/	'root' directory	The base of the Linux file system – any directories not defined within their own partition will be located within the root partition. This partition is mandatory.
{swap}	swap space	Use by the Linux system to store data swapped out of memory – helps the system work better. This partition is mandatory.
/home	home directory	This is where the home directories for the user accounts are located. Setting this up allows user data to be separated from the root partition, improving security/reliability.
/var	'variable' directory	This is where the system stores and utilises data. Log files, print spools and other forms of variable data are stored here. Like /home, set up improve security/reliability.
/boot	booting directory	Contains the information and boot images that Linux uses during the initial system start-up process. Useful of you regularly change the way the machine boots.

Linux File System Capacity

The table below lists the maximum file size and total file system capacity of different file system. Note that most file systems can be 'tuned' to improve storage capacity or performance.

File system	File size limit, GB	File system limit, GB
Ext2/Ext3 with 1kB blocks	16	2,048
Ext2/Ext3 with 2kB blocks	256	8,192
Ext2/Ext3 with 4kB blocks	2,048	16,384
Reiser FS 3.5	4	16,384
Resier FS 3.6	1.073×10^9	16,384

commonly, a `/home` partition is created to limit a particular area of hard disk to system users. If a user kept filling their home directory they could take all the free space from the Linux system causing it to crash. Restricting them to a 'home' partition ensures that user's can't adversely impact the whole file system. The `/var` partition contains system data that is changing continually, and so is segregated to allow better security. Once `/var` and `/home` have been taken out of root, what's left shouldn't change hardly at all. This allows better security on systems like web servers since changes to the root system are easier to detect.

The next matter to decide is how to format the partitions. Swap partitions must be formatted as 'swap' partitions. But there are a number of other file systems that can be used with other partitions. The main contenders are 'Ext2', 'Ext3' and 'Reiser' (see table above), although there are many other file systems that can be used with Linux systems. The main difference, introduced over the least year or so, is between 'journalling' and 'non-journalling' file systems. If you switch off without shutting down the computer will check the entire file system for errors when you reboot. But a journalling file system keeps notes on which files are in use at any time. This means that when restarting after a crash/power failure it doesn't check the entire file system, only those files that were in use. This significantly speeds up the reboot time.

The main file systems are listed in the table above:

- ***Ext2***, the 'second extended file system', was the main Linux file system since Linux's early development. It isn't journalling, but for this reason uses less memory and disk space, and is better suited to older/slower computers.
- ***Ext3*** is the journalling version of Ext2. Ext3 takes care to journal the file system reliably. Although this makes it slower than other journalling file systems, it's more reliable.
- ***Reiser FS***'s structure gives better disk space utilisation, but it doesn't journal fully. This makes it faster to use, but it's journalling isn't fully reliable (although this is due to be fixed in future versions).

How you selected the file systems you use is up to you. On old machines with little memory and hard disk *Ext2* has the advantage of being less resource hungry. Where there is no so much problem with RAM and disk space, *Ext3* or *Reiser* give the advantage of fast rebooting if you don't shut down properly.

Software selection

The last big decision in the installation is what to install. What is crucial here is the available disk space, and the design of the programs that you install.

Most distributions allow you to select what software is installed. This is either organised as 'package groups' – like *KDE programs*, *Gnome programs* or *Games* – or you select individual program packages, or both. What you have to watch here is that clicking on a package group may suddenly eat large quantities of your available disk space. In these situations you may have to deselect another package group, or select the package group, and then with great care manually remove individual programs if the installation systems allows you to do it.

In general, if you are installing a small server, things like games and multimedia programs are not required – so you can deselect these groups. The problem for inexperience Linux users installing servers is that it can be useful to have a graphical interface to manage the system. So you might choose to install one desktop, and then deselect a large number of the programs within that group you don't need in order to conserve disk space.

The other thing to be aware of is that the programs themselves may make additional demands for space. Some programs, like the office suites *Star Office* or *Open Office*, require large quantities of memory to work (*Star Office* requires 32 megabytes of free RAM). If you've only a small amount of memory there's no point installing these programs. Use a less demanding program, like *AbiWord* or *KOffice*. But the main problem are the *libraries* that programs require to work. Gnome programs use the *GTK* graphics library. KDE uses *Qt*. Programs that use neither KDE or Gnome may also use their own graphical libraries. Beware when selecting additional programs because they

may require more disk space than they actually state. This is because the 'dependencies' they create will require additional programs and libraries to be installed.

Installation is a matter of practice. The more you do it, and use the systems you install, the more experience you'll gain about how different programs run under different system conditions. For beginners, unless you've a good reason not to, package groups are an easy way to get started. For those with a little more experience, you can experiment by manipulating different program packages.

In any case, most distributions today manage programs as 'packages', like Red Hat/SuSE's *RPM* packages, Debians *DEB* packages, or Slackware's *PKG* packages. This makes it far easier to add and remove programs after installation, rather than having to juggle system resources during the installation process. The only problem with installing packages after installation is that it can be harder to work out all the dependencies the program has, and find where all those libraries are to install them.

Conclusion

This report doesn't try and tell you how to upgrade an old computer. That's not it's purpose, and in any case there are plenty of books and web sites to tell you how to do that. It also doesn't advocate any particular Linux distribution. That a matter for your own judgement, given the needs of the tasks you need to perform and the restrictions of the system that you are installing on.

What we hope we convey is that the design of the installation is crucial to whether the computer will work well, poorly, or at all. The issues we outline here are the most critical in relation to setting up a system (proprietary- or Linux-based), and by addressing them you improve the chances of getting a good, usable computer system.

Other reports developed by the Salvage Server project will look at other aspects of developing trash tech. systems. You will probably find that other aspects of system installation, especially how you set-up networking, are answered in these other reports. For more information you should consult the Salvage Server Project web site as this will be incrementally updated with new information as the project develops.

Further Information

Salvage Server Project:

To keep up-to-date go to the web site:

Salvage Server Project http://www.fraw.org.uk/ssp/

Gnu/Linux:

For more general links on distributions and Gnu/Linux, see the main Linux web sites:

Linux Online	http://www.linux.org/
Linux Documentation	http://www.linuxdoc.org/
Linux Links	http://www.linuxlinks.com/
GNU Project	http://www.gnu.org/
Linux Newbie	http://www.linuxnewbie.org/

Linux Distributions:

Every distribution has books and masses of online documentation associated with it. For more information see the relevant web site:

Debian	http://debian.org/
Mandrake	http://www.linux-mandrake.com/
Red Hat	http://www.redhat.com/
Slackware	http://www.slackware.com/
SuSE	http://www.suse.de/

Useful books:

Linux in a Nutshell (O'Reilly Press):
Essentially this book is just a collection of the 'man' pages, detailing how you use all the various Linux commands. But as it's in a book, it doesn't matter that you're computer doesn't work – you can still find out what to do. Cost around £25/US$35.

Upgrading and Repairing PCs (Linux Edition) (QUE):
As good as all those other book on repairing and upgrading PCs, but this one specifically looks at the hardware issues in relation to the use of Gnu/Linux. Costs around £44/US$60.

Linux for Dummies (IDG Books):
Feeling stupid and insecure? – try this. Whilst good on basics of client installation, it doesn't do much on servers.

The Salvage Server Project has been developed by the Free Range Network to promote the use of redundant IT equipment as a resource for community and grass roots campaigning organisations. This report has been produced to support the work of the project, and is made freely available to encourage the objectives of the project.