PAUL MOBBS

A practical guide to sustainable IT

Unit 4

APC

This unit is one of 12 sections to a "A practical guide to sustainable IT", a hands-on guide to working with everyday technology in an environmentally conscious way. The guide has been written by environmental activist and ICT expert Paul Mobbs, and was commissioned by the Association for Progressive Comunications (APC) with the support of the International Development Research Centre (IDRC). To download the full text of the guide, or any of the other units, please visit: greeningit.apc.org

A practical guide to sustainable IT

Author

Paul Mobbs

Copy-editing

Alan Finlay

Layout proofreading

Lori Nordstrom

Publication production

Karen Banks and Flavia Fascendini

Graphic design

Monocromo
info@monocromo.com.uy
Phone: +598 2 400 1685

Commissioned by the Association for Progressive Communications (APC).

Conducted with support from the International Development Research Centre (IDRC).

The views expressed in this publication are those of the individual authors and not necessarily those of APC or IDRC.

A practical guide to sustainable IT

Published by the Association for Progressive Communications (APC) with support from the International Development Research Centre (IDRC).

South Africa
2012

APC-201206-SU-R-EN-DIGITAL-162

ISBN: 978-92-95096-71-4

Unit 4

OPERATING SYSTEMS: MAKING THE SYSTEM WORK

A good computer system isn't simply bought or built – it's designed. Even when you buy a system commercially, the process of defining the correct hardware specification and operating system components still applies. The process begins with the construction of computer hardware; then installing the operating system; and finally you decide what application programs to install, how to protect data on the machine, secure access to the system, and devise back-up systems for data to look after your information.

To a great extent, the specification of the hardware affects which operating system(s) can be installed on the machine. Even so, operating systems[1] – the collection of programs which run the hardware and provide programs and functions for users and networks – are themselves very versatile, and define what the system will provide for the user. A single machine can have many different functions: These can be very general, such as in a work station for office or creative work; or they can be highly specialised, such as a file server which shares information across a local network. What determines the role of a system is the type of operating software used (desktop, server or a thin-client terminal) and how it is installed and configured.

Traditionally "green IT" has concentrated on power consumption and carbon emissions, and has largely been centred on the needs of large data centre operators. In this kind of environment saving a few milliseconds here or there, by changing the way the operating system is configured, can make a big difference to the efficiency overall when multiplied by the millions of transactions a day that flow through a single large computer installation. However, equally significant savings in time and energy can be made for small office and individual computer users – and multiplied by the millions of computer users

[1] Wikipedia, Operating system. en.wikipedia.org/wiki/Operating_system

around the globe that potentially adds up to a very large amount of time and resources too.

If we include the human resource side of the system, not just the hardware, the use of operating systems, and configuring the features of the system to protect data, is a critical “resource” issue – as significant as energy consumption. If data is lost it has to be recreated; if files are corrupted, and there is no effective system of backing-up, the work has to be done again. Secure and resilient system design ensures that data loss is avoided and security is maintained, and in this way we avoid the need to expend more time and energy recreating lost or damaged information.

In this section we're not going to examine the mechanics of how an individual operating system is installed. There are more than enough books and guides already available on this subject, as well as the operating systems' own installation documentation. Instead we're going to look at how operating systems are configured to provide certain functions or services, what useful functions we can make these systems perform, and how we can maximise the benefit of IT equipment – including older equipment – to get the most benefit from it and minimise the ecological impacts. Even if you do not install the system on your machine, by understanding the general principles of how they work you can improve your everyday understanding of how systems operate.

4.1. THE DEBATE OVER FREE VERSUS PROPRIETARY SOFTWARE

When evaluating the potential uses of any computer hardware, we first need to decide which operating system to use. For some this decision has already made on their behalf as the new systems people buy are usually preinstalled. For PC-type machines bought from most commercial vendors that system will be a Microsoft Windows system. Macintosh computers have their own system that's specific to the hardware, but for PC-type machines there are also alternative free and open source software[2] (FOSS) operating systems which will efficiently meet the everyday needs of computer users. To understand the difference between these two models of computing – the proprietary software[3] world and the world of FOSS – you need to understand more about the objectives of each model.

4.1.1. Intellectual property rights

Intellectual property rights[4] are the line which demarcates the clash between the desire for freedom of use and access to data, and the need to preserve economic power in the new information age. The options available when installing operating system software are wholly entwined with intellectual property issues – and the choices we make will in turn affect how users are affected by the growing restrictions on intellectual property. Intellectual property rights are the issue which will, more than any other political or technical obstruction, define the economic and legal pressures governing how information systems and the internet will evolve in the near future.

Unlike the technology humans have developed over history, the technology of the information society[5] is not benign. Devices can be made to be semi-intelligent, able to control the terms of how we use the operating system and/or its stored information after we have bought them. In this way society is becoming increasingly constrained by the technological standards that manufacturers develop rather than the political decisions of policy-makers. And if, with the development of more intelligent systems, manufacturers choose to lock-up the functions of their products with digital rights management systems,[6] then society's use of these new tools will be restricted; not to improve our experience of the technology or to protect fundamental civic rights, but primarily to protect the economic interests of their manufacturers.

These new systems, backed up by the protection of intellectual property law, create other restrictions which generate a number of hidden costs to the computer user. Once a user commits to a certain system standard – for example, the choice of using either a Macintosh, Windows or Linux-based PC – the barriers that this creates can make it difficult to change after that decision is made. Primarily there is a cost penalty, either for changing the machine technology, buying new operating software and licensing it, or converting the media that you used with the old technological standard to the new. Technological standards create a means whereby system designers have control over the market for certain goods and services, locking people into using their products in the future. As we move towards a society where information systems are an inescapable part of everyone's lives, we have to pose a very important question about the information systems we create: How do we square the development of a new networked society, where information is intended to flow more freely, with an intellectual property system whose primary purpose is to restrict information flows in order to extract greater value from them?

To examine this idea in more detail lets consider something that we do every day – breakfast!

When we eat a modern, consumer-friendly, pre-prepared breakfast meal we eat brands – *we chew on other people's intellectual property.* At the same time, the acceptance of the brand means that you leave it to others to decide the taste or texture of your breakfast. It's difficult

[2] Wikipedia, Free and open source software. en.wikipedia.org/wiki/Free_and_open_source_software

[3] Wikipedia, Proprietary software. en.wikipedia.org/wiki/Proprietary_software

[4] Wikipedia, Intellectual property. en.wikipedia.org/wiki/Intellectual_property

[5] Wikipedia, Information society. en.wikipedia.org/wiki/Information_society

[6] Wikipedia, Digital rights management. en.wikipedia.org/wiki/Digital_rights_management

to customise the brand. To make a branded commodity meet the needs of the individual user invalidates the generic standard that the purchase of a national or global brand implies. Instead we are required to adhere to the standards the brand conveys, and conform to the conditions that this entails.

The process of combining one product with another to create a distinctive brand identity is called product bundling.[7] For example, Microsoft select what they believe you need to have on a computer or home entertainment console, and in the process confer a certain amount of economic power not just on their own products, but on the product or hardware providers that Microsoft retains an economic relationship with. Likewise Sky's satellite TV system is based upon bundling different pay-per-view channels; and increasingly telecommunications companies offer bundles containing land-line, broadband internet and mobile services, as well as the ability to download TV programmes or films.

4.1.2. Free software – the freedom to "unbundle"

The problem with bundling is that it removes choice and the ability for people to experiment and extend their knowledge without having to undertake costly and sometimes difficult changes to their systems. Using the breakfast cornflake analogy, if there was a separate sachet of chocolate powder it would be possible to choose the strength of the chocolate flavour on the corn flake. I can decide to add a tiny sprinkle, or I could pour on half the sachet over a single bowl. Free software enables this choice to be made. In contrast bundling restricts your ability to mix 'n match a computer system that suits your own needs. Often you get the lowest common denominator of computer system which may be too much for your budget, or too little for your needs.

It gets more interesting when we start to selectively buy the elements that make up our breakfast. If I can unbundle the brand I can chose who sells me my chocolate coating; if I don't like person X's chocolate coating, I can buy person Y's (this was the basis of the anti-trust lawsuits against Microsoft – specifically the way Microsoft incorporates software such as web browsers into their operating systems).[8] I could even produce my own coating if no one produced a flavour I liked. Also, if I find that the quality of the corn flake is not consistent enough, I can experiment and switch to another manufacturer's flake without sacrificing the use of the particular chocolate flavour I had grown to love. When we unbundle products we allow a greater number of options as to how the collection of products can be configured by the user. It is this kind of "real choice" that branding or bundling stifles; by managing the menu of consumer selections those involved dictate the market.

The fundamental difference between the proprietary versus the free/open model of computing can also be demonstrated using the corn flake analogy. The FOSS movement would stress the ability to chose the flavour of the flake and the quality of the chocolate coating. For example, the GNU/Linux operating system has many different "flavours", called distributions[9] (box 4.1). They are all roughly compatible, and run the same standard set of programs, even though they are made by different organisations. You can even combine elements of the different flavours to make your own customised version of Linux – and a number of Linux developers/distribution producers provide information and extra programs for free to help you do this.

To sum up the intellectual property issue and how it relates to software, what defines the difference between the "proprietary" and "open" models of the Information Society is: The difference between what constitutes "selection" versus "choice"; how this preserves our creativity and freedom of expression and communication; and which systems/software applications allow this without expensive lock-ins to one particular company or service provider. Distinguishing the-full range of options you have to solve a particular design, technological or communications problem, and managing the legal or other implications of pursuing those solutions, is the practical method by which you enact this freedom of choice. There is of course no easy way to acquire these skills; it's something you have to work at by learning more about the systems and technologies involved.

[7] Wikipedia, Product bundling. en.wikipedia.org/wiki/Product_bundling

[8] Wikipedia, European Union Microsoft competition case. en.wikipedia.org/wiki/European_Union_Microsoft_competition_case

[9] Wikipedia, Linux distribution. en.wikipedia.org/wiki/Linux_distribution

Box 4.1.

Free software and the GNU/Linux system

The free and open source software (FOSS) movement began in the 1980s as a reaction against the increasingly closed world of computer software. A large part of the computer industry grew out of academia, where the idea that knowledge should be open is generally cherished; being "open" for this community was not a novel concept. Sharing prevents the duplication of effort, and ensures that potential solutions can be peer reviewed and implemented far faster than if small groups are working in isolation. GNU/Linux, the first viable "free" computer operating system to be developed, is a perfect example of the strength of this model of working.

The Linux system kernel was developed by Linus Torvalds during the early 1990s while he was a computer science student in Finland. The internet allowed him to share his ideas with software writers around the world, which led him into organising a collective effort to develop an alternative, Unix-like operating system kernel for the PC. Rather than a wholly proprietary system like Windows, or a partially open system such as the Macintosh operating system, Linux has been developed under a free licence – allowing the code to be shared and modified by users.

The idea of "free" software was developed by Richard Stallman, founder of the GNU Project. The object of GNU was to support the development of free computer software by both professional and amateur programmers. These programs were made freely available for others to use or adapt under the conditions of the GNU General Public License or GPL. The Linux operating system is the product of these two groups; so it's often called GNU/Linux to reflect the contribution of both Linus Torvalds, in the development of the kernel, and the GNU Project, for the development of the licensing system and the other programs with which the Linux kernel was originally developed.

The success of GNU/Linux has been driven by its licence. It enables programs to evolve in a way that serves the needs of computer users rather than the ambitions of the developers – which gives it a significant advantage over "closed" systems such as Windows where adaptation or extension is usually prohibited. The fact that systems/programs can be downloaded for free, or bought at a low cost, has implications for its users too: You can install it on any number of computers without having to pay for extra licences; you can copy the system for friends or colleagues to use; if required, you can take the system apart and modify it to suit your own needs, and pass on those modifications to others. If these same actions were carried out with the programs developed by "closed" software developers, those involved would be liable for prosecution!

Linux is the kernel of the GNU/Linux system – the core program which runs the computer hardware and makes the memory, disks and other components available for programs to use. Within the whole GNU/Linux "system" there are hundreds of other programs required to provide all its functions, and the user-friendly applications that provide for people's everyday needs. Early on in the development of GNU/Linux, enthusiasts developed FTP sites containing all the files to install a Linux system. Then shortly after that, as the internet was slow when moving large quantities of data, they developed CDs with helpful installation programs which automated the process of installing and configuring the system for less experienced users. Eventually these efforts created commercial or semi-commercial organisations retailing their own version of a Linux system. These different versions of GNU/Linux became known as distributions, or "distros".

There are a whole range of Linux and Linux-related systems available today. Some are primarily desktop systems, providing an equivalent range of creative and office-based software that is commonly available with proprietary systems; others focus on data servers and more technical computer science related applications; while some have evolved into single-purpose systems such as network routers, firewalls, and even digital video recorders or media studios. Other than the versions meant to be conventionally installed on computers, systems are also configured as "live media" that boot direct from a CD/DVD or memory stick into a working desktop system.

The issue when comparing Linux distributions isn't necessarily "which is the best". Usually a distribution represents a certain design philosophy that is optimised for certain applications rather than attempting to be "the best distro". There are many different Linux-based systems to suit different needs, and you need to find the distro that meets your expectations, which is quite simple as you can freely download most Linux distros to try out. For the latest news about Linux distributions available go to the DistroWatch site: distrowatch.com

In the final analysis, in choosing which operating system to use on the equipment you have, the most important indicator must be that it satisfactorily "does what you need to do". Whether that is achieved using proprietary or free software must be part of the decision process that you undertake when you de- sign the specification of the computer system. The important issue is that you are aware that you have a "choice" between different operating systems, that you do not have to accept systems as they are installed or provided, and that those choices result in very different outcomes in terms of costs, intellectual property restrictions and compatibility/interoperability.[10]

4.2. OPERATING SYSTEMS AND OBJECTIVES

Operating systems are installed to fulfil a purpose; although the system might be the same, it is how the components of the system are selected and configured which give the system the ability to carry out the tasks we require of it. When it comes to defining the roles of different machines, and selecting the software to fulfil those roles, we see another distinction between proprietary and open source systems:

- With proprietary software there is a clear distinction between roles. For example, there is one system for Windows desktop use, and another system for Windows server machines. You purchase software to fulfil the role you require of it.

- With free software, and to some extent the Macintosh OS (because it is based on a Unix-like system), the line is blurred. It is not possible to enable a strict divide between what a desktop is and what a server is. In practice, as free software systems allow the installation of additional software more easily, it is very easy to set-up a desktop machine which can also act as a server on a local network (for example, running a web server to provide an intranet).

4.2.1. Finding the best fit of machine to operating system

Unlike the world of proprietary software, where you have a limited choice of hardware to fit the power demands of the latest Windows or Apple "iOS" operating system, with free software you have the option to design a system to fit the application you need it for. Older hardware, which would be difficult or impossible to use with recent versions of Windows, can be found new purposes/have its working life extended using free software. Even fairly new hardware can be configured in different ways to perform different functions, in the process creating different levels of performance or security to match the needs of the user.

For example, if you install a web server on a desktop machine, then it is possible to carry out complex web design, and then test the designs – including scripting and other functions – without a network connection to a dedicated server. Additionally, when using proprietary systems you're penalised for using more than one machine because of the need to license each installation. With free software there's no such financial penalty – *additional installations are free*. In this way we can design machines inexpensively to serve specific purposes, and undertake that task as securely and reliably as possible.

Using more than one machine, and through this consuming more electrical power, might not appear to be very "green". This ignores the embodied energy of the hardware. If you can make equipment last for longer – as such a large part of the overall impact is tied up in manufacturing the hardware – you reduce the impacts overall. Using an old machine can avoid the need to buy new/more powerful equipment, which saves money as well as energy and pollution. For example, an inexpensive way to backup a workstation computer, without the time and expense of making lots of data discs or buying a new external hard drive, is to take an old

[10] Wikipedia, Interoperability. en.wikipedia.org/wiki/Interoperability

machine and install it as a data server to store copies of your important data. This doesn't use a large amount of energy either, because older equipment often consumes less power, and in any case this type of system need only be turned on for an hour or two while the back-up is taking place.

The first decision you need to make is: What can I do with this equipment? This is a complex decision, and has a direct relationship between what functions you need from the machine and what its internal electronics are capable of supporting (see box 4.2). Breaking down our IT needs into a series of clear roles and specifications requires thought and planning to define what we really want. In the corporate environment the aim is often to provide new infrastructure, usually under contract, because it represents the most economically efficient solution. In contrast, in smaller organisations and the voluntary sector there is the option of adapting existing hardware to work more flexibly, using the skills of those who will use the system, because that represents the least-cost route to finding a solution to their needs.

As outlined in unit two, one of the important objectives of sustainable IT is to maximise the life of the equipment. There are two ways of looking at this:

- Firstly, we can maximise the use of a single machine so that it is using as much of its processing capacity as possible for as much time as possible. This is the objective for dedicated server machines and large data centre rack servers – often because there's a cost penalty to having unused machine capacity sitting idle. Achieving this outcome requires attention to the minute details of system configuration and the organisation of hardware.

- Secondly, we can prolong the life of equipment – usually older equipment — by giving it less intensive processing tasks which match the capabilities of the hardware. For small offices, older machines can take on roles which support or extend the functions carried out by the machines people routinely use – thereby extending the functions available to the user, or freeing up capacity on their main machine to allow them to work more easily. Finding opportunities to do this requires that we understand more about the different roles machines are able to perform, and then match the available hardware to those roles.

Too often the "creative" IT world focuses on the use of a single, often new or powerful desktop machine when, if we were to think more openly about using a local network, it is possible create some very helpful solutions using older equipment. For example, routinely processing large video files or mastering video DVDs can tie up a system for long periods of time, affecting the completion of other work on that system. But if a second machine were set up with a remote desktop[11] accessed over a network you could undertake these same operation without restricting the capacity of the main workstation machine. This also does away with the need for the second machine to have a keyboard or monitor. Likewise, given that laptops use a lot less power than desktop machines, an old laptop is a much cheaper option to run 24-hours a day for file sharing, or to configure as a server running a small local network, rather than leaving a more power-hungry machine running 24/7.

4.2.2. Hardware specifications and practical applications

Let's begin with hardware and the demands of different operating systems. In table 3.1 we looked at the hardware specifications required by different operating systems. In table 4.1 we look at this from a different perspective – what applications can be achieved with a certain level of hardware. Remember that the minimum requirements of different operating systems, given in table 3.1, still apply. For example, to run a lightweight server on old hardware you would use a lightweight Linux distribution or Windows NT, rather than the latest fully functional Linux or Microsoft server systems.

Table 4.1 lists a number of different applications for different classes of hardware. Here's an explanation of what each one encompasses:

Desktops — office-related, graphical and communications tasks

- *High power desktop* – high-end desktop work such as graphic design, video editing and DVD mastering (less powerful machines can

[11] Wikipedia, Remote desktop software. en.wikipedia.org/wiki/Remote_desktop_software

Box 4.2.

Hardware specs and potential system applications

Matching the performance of hardware to potential applications is a rather complex subject. The more powerful the hardware, the simpler it is to install an operating system. The skill lies in being able to determine what is the lowest specification you require for a certain task so that you can maximise the lifetime of equipment and keep costs down. Installing systems on older equipment with restricted memory and disk space can be more of a challenge, especially if it requires the use of a text/command line interface (which requires more details knowledge of the operating system) rather than a graphical interface (where the options are usually presented as simple menus).

If you only want a desktop machine, you need a computer with a desktop-level specification. Even then, there's a considerable difference between simple typing/word processing applications and the type of power required to edit large colour images or process video files. When considering the more creative use of computers, such as backing-up or networking support, you can use a far wider range of equipment to solve a particular problem.

Table 4.1 lists various grades of hardware and the types of application that it can perform. Note that the demarcation between what is “possible” and what is “practical” isn't as clear cut as shown in the table – what matters is that the machine can run the tasks that you design it to perform, at the speed/reliability that you require of the system.

Table 4.1.

The use and reuse of computer hardware for different applications

Application	Desktop			Data processing		Support systems			Downcycle		Operating system	
System specification	High power desktop	Medium desktop	Lightweight desktop	Parallel processing	Power server	Download box	Lightweight server	Network support	Educational recycling	Scrap	GNU/Linux	Windows
Recent machine dual/quad core, +2GHz CPU, 2-4GB RAM	×				×						H	7/08
Good machine dual core, 1-2GHz CPU, 1-2GB RAM	×	×		×	×	×	×				H/M	V/03
Fairly good single core, ˜1GHz CPU, 512MB-1GB RAM		×		×	×	×	×	×			M	V/2K
Old machine 800MHz-1GHz CPU, 256-512MB RAM			×	×		×	×	×	×		M/L	X/2K
Older machine 500-800MHz CPU, 128-256MB RAM			×					×	×		L	X/NT
Very old machine less than 500MHz CPU, 128MB RAM								×	×	×	L	98/NT
Challenge! less than Pentium-3/ 500MHz and 128MB RAM									×	×	L	98/95
GNU/Linux system specs:	H, heavy (SuSE, RedHat); M, medium (Debian, Fedora, Mint, Ubuntu); L, lightweight (Damn Small, Puppy, Slackware).											
Windows system specs:	7, Win. 7; V, Vista; X, Win. XP; 2K, Win. 2000; 08, Win. Server 2008; 03, Win. Server 2003; 98, Win. ME/98; NT, Win. NT; 95, Win. 95.											
Note that Macintosh hardware hasn't been considered in the above list. Different versions of MacOS are designed for different generations of Mac hardware, and so the decision about which machine requires which version of MacOS is determined largely by the available hardware.												

do this, but will take much longer to do so). For these installations the latest versions of an operating system are used in order to have access to the latest software and hardware.

- *Medium desktop* – general purpose machine, capable of most office-type tasks, communications and some graphic design/editing. It could run a Linux-based web server for undertaking web design work. The latest operating systems would run slowly on these machines. While using the latest system is not impossible, it may be more productive to use slightly older software if that provided the required applications and hardware drivers.
- *Lightweight desktop* – for internet communications and simple word processing, or it could act as a "thin client" connected to a more powerful machine over a network. A problem when using Windows systems is that hardware drivers may no longer be available, especially for printers. Anti-virus software will also not be available and there may also be problems with network integration. This makes free software preferable as these system have better hardware support, use open-network standards, and malware isn't such an issue.

Data processing – heavy-duty data processing, storage and network control

- *Parallel processing* – use as a second machine for processing tasks that interfere with your everyday work. For example, video file conversion, DVD mastering, creating DVDs, and any other tasks that runs for many hours or use a lot of power. You could use a screen and keyboard, but it's easier to access the box from your main machine using a remote desktop (for graphical applications) or secure shell (for command line control). As well as single machines, many small machines can be connected together to form a more powerful "cluster"[12] for specialist applications (e.g. video animation, scientific analysis).
- *Power server* – a dedicated server machine for a small office, or to serve content to a busy network, running a range of protocols, databases, or even thin clients[13] (using older machines as the clients, and the powerful machine as the terminal server).

Support systems – machines to run networks and store data

- *Download box* – a dedicated machine to run 24/7 internet services such as file sharing or remote data/file access (for non-professional/small-scale applications this is most efficiently done using an old laptop due to its inherently low power consumption). Where peer-to-peer file sharing is used for information distribution the machine could be installed with a standard desktop operating system. For low volume web or FTP access, or to run encrypted tunnels for remote access to the network, a server system is a better option.
- *Lightweight server* – a simple server machine to support services across a network, rather than a single user. Like the "power server" class above, powerful machines could be used as a terminal server service for a thin-client network, but many older machines might struggle to do that for more than one or two client terminals. Instead older equipment could be configured as a file server/encrypted storage server for backing up desktop machines (using FTP or a shared/networked file system), or a simple web server to run an intranet for a small office, or a mail server to forward incoming mail for users on the local network.
- *Network support* – if good quality, it could perform some lightweight server functions, although it may be difficult creating enough hard drive storage to make it worthwhile. If using a system which required minimal hardware it could be used to run the programs required to operate a local network, and perhaps act as a firewall/gateway for a broadband internet connection. Note that as well as installing a full operating system, there are a number of stand-alone installations[14] which will provide dedicated secure network services, Wi-Fi access points and router/firewalls. They boot from CD/DVD, install largely automatically, and can be configured from menus rather than requiring the more expert skills

[12] Wikipedia, Cluster (computing). en.wikipedia.org/wiki/Cluster_(computing)

[13] Wikipedia, Thin client. en.wikipedia.org/wiki/Thin_client

[14] For example: Smoothwall www.smoothwall.net; En-Garded Linux www.engardelinux.org; or Zeroshell www.zeroshell.net

required to configure an operating system to perform these functions.

Downcycling[15] – productive end-of-life uses for old machines

- *Educational reuse* – if the machine can still run a lightweight Windows or Linux system then it can still perform a useful function as a training platform to learn/practice hardware and operating system installation skills. By dismantling, re-assembling and then installing the machine people get the skills/confidence required to do the same types of operation on more expensive equipment, but without making expensive mistakes if they do something wrong. Any quality of machine can perform this purpose, although because of the risk of damage to the machine it should be considered "expendable".
- *Scrap* – "scrap" doesn't necessarily mean throw the whole thing away. Old boxes can be scavenged for their screws, cables/connectors, fans, heat sinks and other reusable components – to be kept as spares for the repair of other machines. Hard drives can be used with drive caddies to back-up large amounts of data from other machines. Unless they have no further use or have a poor picture quality, one or two old monitors can be kept for use in emergencies if another monitor fails. The only thing that must go for electronic waste disposal and recycling, as they have little value for reuse, are the metal chassis of the desktop box, plastic enclosures and motherboard/electronic circuits.

Looking at the list of potential uses for equipment above, it might seem a rather complex approach to design different machines to perform different jobs. In practice, to perform a task well with a computer, the machine should be designed to serve its purpose as well and securely as possible. For example, a general purpose desktop machine could undertake all the roles listed in table 4.1, but to do so simultaneously would drain some of the CPU's power if it all took place on the same machine (every extra program running on the system takes a little bit of power away from the important job you're doing at the time). It's more efficient to design boxes for specific tasks, optimising the hardware and software to provide both functionality and an efficient level of power consumption for the job concerned (for example, server machines do not need a monitor).

Making a redundant computer into something useful doesn't require an expensive upgrade of its components (as outlined in unit 3) – as is usually the case when installing a fully functional desktop system. Again, it's a matter of design, and matching the minimum specification of the job to the hardware available. Of course there comes a point when a machine is beyond practical use, usually when it can only run a lightweight[16]/text-only operating system, or you can't install enough hard disk storage or memory to provide a useful level of service. Even then, old machines can be useful for learning how to maintain or rebuild computers, or installing operating systems. These matters are examined further in unit 10, *Reuse and reclamation*, and unit 9, *End of life and disposal*.

[15] Wikipedia, Downcycling. en.wikipedia.org/wiki/Downcycling

[16] Wikipedia, Lightweight Linux distribution. en.wikipedia.org/wiki/Lightweight_Linux_distribution

Box 4.3.

System documentation

Even where one user looks after their own system, remembering all the relevant information about the system is very difficult. Where a group of people look after a number of computer systems, it's wholly impractical. It's a good time-saving and security practice to document each system installation. Begin with details such as the model of the computer and any relevant serial numbers (this is the sort of data insurers, the police, or repair agencies may require). Then there's the hardware profile itself (e.g. knowing what type of memory modules used in case you need to replace them). Finally there's the system-specific information – such as the type of system installed, hard drive configuration and critical passwords. If you buy a system pre-installed, then you should be able to pull together much of this information from the vendor's hardware specifications and the data provided by BIOS or the operating system.

By documenting this information and storing it securely it simplifies the process of system maintenance, especially where there are a number of machines administered by different people. If you store this data as a simple text file (*not on the machine it is related to* – in the event of a fault you will need this data!) it can be used as a working document; adding information on system maintenance, any faults which occur, and any remedial action which was taken. In this way it s possible to keep a log for the machine, which makes system maintenance a far simpler procedure.

The table below outlines the types of data that is relevant to keep. Whether you keep all of this depends upon the technical level of control you wish to have over the system:

Date installed/technician ID	It's useful to keep a note of the date installed, and the identity of the person installing the system
Operating system/version	Include details of any service pack upgrades/Linux kernel ID
Processor type	Make, model, speed, number of cores, L1/L2 cache
System memory	Note the type, speed & no. of modules, not just the memory size
Hard drives	For each hard drive note the make/model, its capacity, and an outline of the partitions on the drive and file systems used
BIOS serial number	Usually displayed on the BIOS splash screen – useful to keep in case of theft, or if the insurer or repair contractor asks for it
BIOS key	Which key to press to enter the BIOS configuration menu
BIOS passwords	Note the BIOS set-up/administrator and user passwords, if used
Motherboard	Make and model, as well as the serial number if you can find it
Plug-in cards/peripherals	Make and model, and serial number
IP configuration/address	Note whether the interface is configured with a static IP address or DHCP, its gateway/DNS addresses, and its MAC address
Administration/Root password	Critical password – you should log this and keep it safe
User accounts	For each user account, document user ID and any user-specific configurations used with the account (e.g. access permissions)
Boot loader password (Linux)	A boot loader password prevents bypassing of the root password
Hard drive encryption password	Critical password – if you use hard drive encryption, log this
Enabled network services	Note which services are enabled on the machine

Finally, all this information is of course very sensitive. It's not the sort of data that you would print and pin to the wall! For that reason the file should ideally be encrypted. Keeping the information on an encrypted memory stick or hard disk is one option. Another is to use a symmetric cipher so that it is not necessary to keep a public/private key pair with the file. In the event of someone needing to access that data they could read the file from the network and decrypt it to obtain the information they require. If changes are made it's a simple matter to decrypt, edit and re-encrypt the file, and then replace the copy kept for maintenance with the new file.

4.3. PLANNING AND DOCUMENTING INSTALLATIONS

When installing systems, planning is important – each user needs to access the resources they require to work. That might be on one machine, or it might require configuring machines across the network – for example, allowing access to file servers or shared resources such as printers. It helps to plan these details first so that you can organise how you will install any new machines, or modify the configuration of others. Where a group of people look after a small network, keeping written logs and procedures is essential so that everyone can co-ordinate their efforts and not lose vital information about changes or modifications to machines or system settings. Even when there is only a single machine in use, there is still information that is essential to its operation and maintenance – such as the critical passwords for the machine, and also the passwords to access and configure a broadband router or similar devices.

Each "system" that is installed should have its own log of critical and background data (box 4.3). A "system" includes details of not just the machine, but also any hardware that one or more machines are attached to. While not essential, it's good practice. If a machine should fail, or if a critical password is forgotten by its user, keeping a log securely stored on another machine (or on a back-up memory stick) allows quick and easily access to the data you need to unlock the system – or gives information to technical support staff so they can diagnose problems more easily. Even if a machine is not in use, if you have a log of its vital statistics, it makes it easier to decide if it can be re-installed and put into service (or stripped for the required parts) if another machine fails.

In addition to the hardware, it's also important to keep a log of operational data. For example, the connection details/passwords for broadband access, access/configuration details of email and internet services if they are run by a third party, and even details of insurance or maintenance contracts covering any equipment. By having a secure, central location for this critical information you will save much time and resources at a later date should there be a problem – be that with the hardware, or an important staff member meeting with an accident.

4.4. POWER MANAGEMENT SETTINGS

Following installation, the operating system is configured to optimise its functions to meet the intended purpose(s) of the machine. When configuring a new/newly installed desktop system, after configuring user accounts, the next step is to configure power management (box 4.4). Power management is important, not just because of the power consumption issue. When you configure power management you're also setting up features such as screen savers. These are basic security measures to control access to the system. Setting a short lock-down period on a screen saver or the power management menu doesn't just save energy – it makes the system more secure by preventing casual access.

4.4.1. How power management functions

Power management offers various options. The primary one is giving the machine time limits to define what happens after it has been idle for a certain period. There are five levels of power management which determine how much power the system consumes:

- *1. **Fully operational*** – the machine is working normally. Note that the level of "normal" power consumption can be also modified in various ways. The hard drives can be powered down after a certain period of inactivity, and the priority given to different parts of the operating system. The user applications can also be changed to manage the load on the operating system. This is normally only implemented on laptop systems, but there's no reason you cannot implement the same policies on a desktop machine. The trade-off for optimising power management is to lose some power for user applications, which often means that users playing high power-hungry video games will deliberately disable power management.

- ***2. Stand-by, sleep or suspend*** – this is a low-power standby mode in which the machine's operation is halted, hard drives and the video display are powered down, but power is maintained to the memory. It does not consume as much compared to running at full power, but it's significantly more than if the machine is put into hibernation or shut down. When configuring screen savers, setting the suspend period to a few minutes after the screen saver activates will save a lot of power. The problem is that those who use their computer as a high-powered media player will object because their music will stop playing. Consequently it s often less intrusive to leave the suspend option for ten minutes to half an hour after the screen saver activates. A problem with suspend is when power is interrupted, or the laptop battery runs down. On a laptop the normal procedure is to transfer to hibernation mode or to execute a soft shutdown. On a desktop, which has no battery back-up, a power cut or inadvertently turning off a machine in sleep mode can cause data loss from any open applications. For this reason, of the two options, hibernation is more secure.
- ***3. Hibernate*** – this halts the machine, backs-up an image of the working machine on the hard drive, and then powers down to an off state. The difference between hibernation and sleep modes is the amount of time it takes to restart. A machine in sleep mode will wake up after five or ten seconds of the wake button being pressed. Restoring full operation from hibernation can take thirty seconds or more, depending on the speed of your BIOS self-check (it is quicker than having to power-down and reboot, but some users find the wait problematic). Hibernation can sometimes cause problems if an error occurs when the hardware is powered down, or the machine is switched-off at the mains. When the machine restarts, instead of booting the operating system, it copies the memory image back from the hard drive and resumes operation from the point it was told to hibernate. Errors can cause the reboot to crash, and the machine has to be reset and rebooted from scratch.
- ***4. Soft shutdown*** – telling the machine to "shutdown" powers down the system. This means that the system is "off" but the power is still connected at the power supply (hence "soft"shutdown as only the software has been stopped). With soft shutdown you restart the machine by pressing the power button, and then the system reboots from scratch.
- ***5. Hard shutdown*** – the human operator physically switches off the machine at the mains. This disconnects all power so that the machine consumes nothing.

On laptop systems, not only are the major components integrated into a single unit, they are optimised to use as little power as possible – but to do this you trade off some of the processing power available with desktop machines. For demanding jobs the power saving limitations on laptops mean that jobs can take longer, unless you deliberately turn off the power saving features on the laptop (at least while its running on AC power).

However, for most everyday office-related tasks a lot of processing power isn't required, and so enabling these features will have little effect on the speed of operation.

You can also enable similar power saving features[17] to those used on laptops on most desktop systems in order to reduce their everyday power consumption:

- For *Windows XP* to *Windows 7*, the standard power management system offers three modes: *High Performance*, *Balanced*, and *Power Saver*. If you're not editing large colour images or manipulating large databases and spreadsheets, Power Saver mode will make little difference to the way the system appears to work – and you can always switch back temporarily if you need to undertake more powerful data manipulation. There are

[17] Wikipedia, PC power management. en.wikipedia.org/wiki/PC_power_management

Box 4.4.

Power management and the machine hardware

Power management is an issue that developed alongside the growth in mobile equipment during the early 1990s – especially laptops, where prolonging battery life was an important consideration to the usability of the device. As the technology has evolved, it has become an important factor in managing the power consumed by desktop and server machines too.

New microprocessors, particularly the multi-core type, can be configured to run at different speeds. The amount of power the chip consumes is proportional to its operating speed and the supply voltage of the processor. Varying the speed of the CPU from the operating system performs two roles: It prevents excessive heat generation when the power of the chip is not required; and, more importantly, it reduces power consumption so that the amount of energy the machine uses can be scaled to the demands on the system. On laptops it's often used to deliberately extend the battery life of the machine at the expense of operating speed.

Depending on the age of your hardware, there are two power management systems in use:

- The *Advanced Power Management* (APM) system was developed in 1992. Support for APM began with Windows 95, but was not included with Vista (the Linux 2.6 kernel still supports APM in most distributions). Any hardware which only supports APM will not be able to function with Vista or Windows 7 unless you have a specific driver for those systems.
- The *Advanced Configuration and Power Interface* (ACPI) was developed in 1996, and has superseded APM in most PC hardware. ACPI is more versatile, and offers more options to configure power consumption by the processor, display, hard drives and peripheral devices. ACPI has been supported since Windows 98, and is supported in most Linux distributions.

With Windows systems the power management software for specific types of hardware must be loaded along with the other driver software during system installation. Power management is configured from the *Control Panel:* In *Windows XP*, you'll find the *Power Options* icon on the *Performance and Maintenance* menu; in *Vista* the icon is on the *System and Maintenance* menu; in *Windows 7* it's on the *Hardware and Sound* menu. The power options dialogue allows you to configure the power management features available with the hardware you are using.

On the Macintosh (OSX), you'll find the power management icon in the hardware section of the *System Preferences* menu.

With Linux-based systems the APM/ACPI service is usually enabled automatically – but to function you must be using hardware which conforms to the APM/ACPI standard. This is a particular problem with certain brands of laptop and desktop hardware developed for use primarily with Windows. As is usually the case when installing Linux, you need to check the compatibility of the hardware before you start. As GNU/Linux has become a more popular operating system, power management is no longer the obstacle that it was five or more years ago. To be certain that all hardware functions are available, check the information and reviews of motherboard, peripherals and laptop systems at linux-hardware.net or www.linux-drivers.org

The power management controls of the operating system interface with the APM/ACPI firmware in the machine's motherboard. The motherboard will then control both the devices connected to the motherboard, as well as sending signals to the monitor and any compatible USB devices. Depending upon the motherboard's BIOS settings, it s possible that power management has been disabled or limited. Although power management is often enabled by default, you should check the configuration of power management in BIOS, and change as required (if necessary identify the make/model of the motherboard, then search for and download the board's user manual to check power management options).

also a number of third-party[18] power applications which allow you to refine these settings, and which manage the programs that run on the system to reduce the demands on the processor, memory and hard drive.

- Most Linux distributions offer similar power management features to Windows. As with the FOSS world in general, there are also specialised projects[19] which develop power management systems for use across the Linux community. Many of the popular Linux distributions are now producing their own detailed system documentation on implementing power saving features/policies in server and desktop machines – and these guides are available free on the distribution website. For the more technically minded, there are also free programs which work across most distributions to audit and improve power performance. The leading application is powertop,[20] which not only monitors how the system/running programs consume energy, it also suggests ways of improving system configuration to reduce power consumption.
- MacOS also has its own built-in power management modes and there are third-party applications which can monitor and optimise the operation of the system.

4.4.2. Advanced power management options

For servers, where you don't always have a person on-hand to press the "power" button, starting the machine can be a problem – especially if there are hundreds of them. For machines being used as servers there's an option called "wake on LAN" (WOL). If a data packet is sent to a machine in suspension or hibernation containing the card's unique address number (some also require a security password), the machine will restart, provided you have a WOL-compatible network card, Later the machine can be remotely put into suspension/hibernation, or it can suspend operations according to its own power management configuration, until it is needed again. Using WOL allows a single operator to have control over hundreds of server machines – often using load monitoring programs to shut them down or wake them up in order to match demand while minimising power consumption. For small system users, it can also be configured to wake-up file servers or data processing machines so that they don't use as much power when not required. The standard power management menu allows you to configure time limits to put the machine into either sleep, hibernation or soft shutdown depending upon how long it has been idle for; or, for laptops, when the battery is running down. You will also have the option to spin down the hard drives and shut down the display a certain time before entering suspend/hibernation. This reduces power consumption without the inconvenience of locking out the user while the system re-initialises. As noted above, of the two options, provided the hardware is fully ACPI compatible, hibernate is the more secure state.

In most operating systems strict criteria for power management are not configured as a default they have to be set by the user or the system installer. That is why configuring power management should always be one of the final steps in system installation/configuration. In addition, it's also possible to get power optimisation programs which monitor system usage and suggest ways of changing power settings and system timing signals to reduce power consumption – although these often need more advanced knowledge of the operating system in order to enable them. There are various proprietary programs available for Windows, a number of which allow remote system access over a network, so that power setting can be configured to optimise power consumption (for example, Data Synergy's PowerMAN or 1E's NightWatchman, and Verdiem produce a free application called Edison). There are also proprietary applications for MacOS. For Linux there are applications developed for large servers, and, as suggested, there are also free programs such as powertop.

[18] Bemowski, Karen. Windows IT Pro (September 2010). Buyer's Guide: Power Management Software for Windows Workstations. www.windowsitpro.com/article/buyers-guide/Power-Management-Software-for-Windows-Workstations-Buyers-Guide Download the product table from www.windowsitpro.com/content/content/125800/101910-updated-125800-table.xls

[19] For example, the Less Watts projects www.lesswatts.org

[20] powertop www.lesswatts.org/projects/powertop/

Box 4.5.

Operating systems check – list

Always match operating systems to hardware specifications

A machine which doesn't have enough system resources to run its installed software will struggle, and waste the user's time; a system that is too powerful for the software installed upon it represents a waste of resources – *in effect the user is wasting the system's time*. What we should find, by planning our needs and matching them to a machine specification, is a balance between these two extremes.

Define a role for a machine and install a system that meets that role

Operating systems work well when they perform a defined purpose. That could be a general purpose like a desktop machine, or a specific purpose like a data server. Sometimes you can combine both – such as a desktop with a web server installed to allow web development. In practice it's better to manage the load on machines in a way that optimises their role rather than obstructing it. Find a role for the hardware you have, and only buy new hardware when what you have can no longer perform the tasks you require reliably.

Old machines can be useful if you assign them a suitable role in the system

Some roles require little processing power, such as a printer server, file server, or running network services such as DHCP and email. A new server will do all this and more – but if you have a suitable old machine the ecological option is to use that equipment until it can no longer reliably fulfil the role.

Always document the machine hardware and the system installation

Knowing your hardware is essential to maintaining it, and especially for dealing with faults and bugs when they crop up. More importantly, data secured behind password access controls, and especially encrypted data, can be lost if the password is forgotten. Securely stored system documentation, listing system and hardware configurations, can avoid this.

Configure power management options to reduce consumption, but avoid settings that are so obstructive that the user disables them

Power management can make savings in energy use, provided that it does not become so obstructive that the user disables it. After a period of idleness, switching to hibernation is more secure than sleep mode as it s less likely to create data loss in the event of a power failure/ the machine being switched off.

Give feedback on power management

The best way for a user to understand the load of their system is to get direct feedback on the desktop – allowing them to change their behaviour and not over-tax their system.

4.4.3. Providing feedback on system load

Finally, for the user at their computer desktop, there is very little indication as to the level of "stress" that their machine is experiencing at any point in time. The machine can appear to slow for no apparent reason, or the mouse can stop moving, leading them to click buttons and press keys, which will often exacerbate the problem. The solution is to install a small power monitor application on the task bar of the desktop to give feedback. This will show when the system load is reaching a problematic level. More technical applications will also break down the display to show the overall system load, processor(s) speed/load, network activity and hard drive activity.

By giving feedback you allow the user to moderate their demands on the system, making it use less power, and avoiding overloading the system to the point where applications crash or the system hangs. There are free load monitors available for Windows systems, and Linux desktops usually come pre-configured with applications such as *system-monitor*.

A practical guide to sustainable IT

This practical guide to sustainable IT offers a detailed, hands-on introduction to thinking about sustainable computing holistically; starting with the choices you make when buying technology, the software and peripherals you use, through to how you store and work with information, manage your security, save power, and maintain and dispose of your old hardware. Suggestions and advice for policy makers are also included, along with some practical tips for internet service providers.

Written by IT expert and environmentalist Paul Mobbs, the purpose of the guide is to encourage ICT-for-development (ICTD) practitioners to begin using technology in an environmentally sound way. But its usefulness extends beyond this to everyday consumers of technology, whether in the home or office environment. We can all play our part, and the practice of sustainable computing will go a long way in helping to tackle the environmental crisis facing our planet.

This is also more than just a "how to" guide. Mobbs brings his specific perspective to the topic of sustainable IT, and the practical lessons learned here suggest a bigger picture of how we, as humans, need to live and interact in order to secure our future.

The guide is divided into 12 sections (or "units"), with each unit building thematically on the ones that have come before. They can be read consecutively, or separately. The "unit" approach allows the sections to be updated over time, extracted for use as resource guides in workshops, or shared easily with colleagues and friends.

The guide has been developed on behalf of the Association for Progressive Communications (APC), with funding support from the International Development Research Centre (www.idrc.ca). It is part of a APC's GreeningIT initiative, which looks to promote an environmental consciousness amongst civil society groups using ICTs, and amongst the public generally. Other publications and research reports completed as part of the GreeningIT initiative can be downloaded at: greeningit.apc.org

APC

IDRC | CRDI