UNDERSTANDING COMPUTER OPERATING SYSTEMS

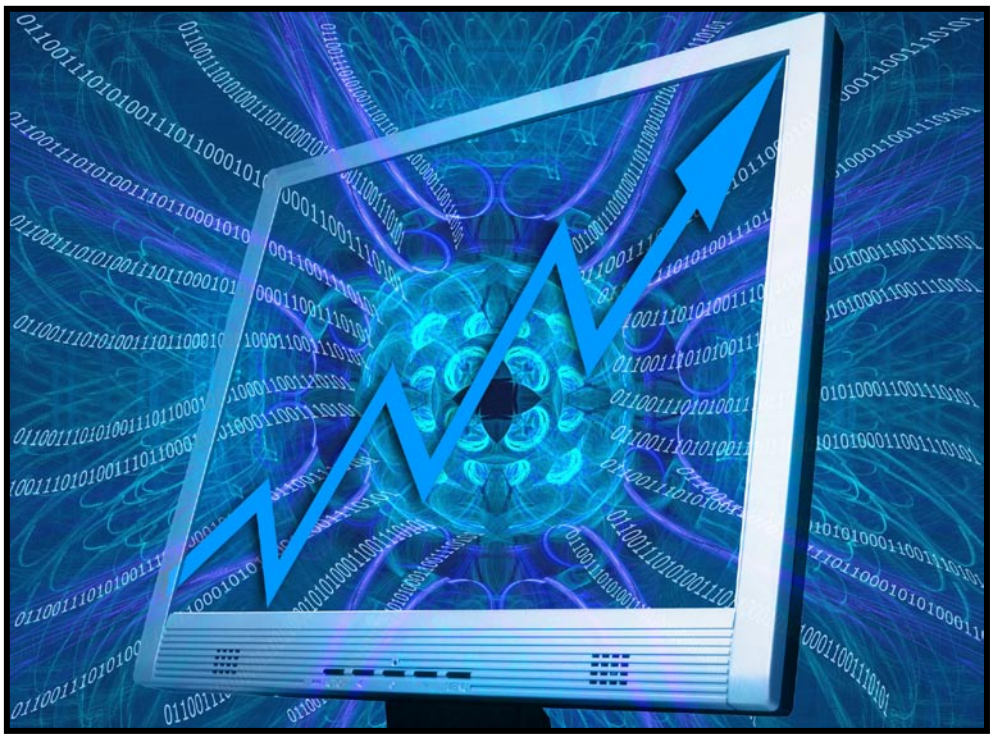

TABLE OF CONTENTS

INTRODUCTION

Our body couldn't function without our brains. The brain tells the various pieces of our body how to work and how to interact. Without a brain, we wouldn't be able to do anything at all.

An operating system is kind of like the brain of a computer. You have a bunch of hardware like the CPU tower, the monitor, and the keyboard, but without a CPU, they can't do anything but power up and turn on. The operating system organizes files and lets the hardware know what it should do.

In the early days of computers, there was just one operating system. As computers progressed, the OS turned into MS-DOS, but computers really weren't capable of doing much without software. Then Bill Gates came along.

With the founding of Microsoft, the computer operating system came into its own and brought computers to new levels of functioning and technology. Although the brand names of operating systems are few, they do perform different tasks depending on the requirements of the computer user.

While the dominant OS today would be Microsoft Windows, there are other types of operating systems that offer different features. Those would include Linux, UNIX, and OS X.

In our technological age, there are operating systems in more than just computers. Many of the sophisticated new cell phones have their own operating systems, and wireless access points have their

own OS to provide wireless internet to customers. In fact, the computer in a cell phone today is more powerful than a computer was twenty years ago.

As you can see, the operating system technology has evolved and is continuing to evolve. It seems like Microsoft is always coming out with a new and better operating system which leads people to wonder whether or not the system they are currently using is really the best one.

It can be confusing. But it doesn't have to be. In the pages of this book, we'll explore operating system in depth. You'll learn about what they do, how they work, and what needs specific systems can meet. Ultimately, the choice is a matter of preference, but it helps to be informed on what you are really getting when choosing an OS.

WHAT IS AN OPERATING SYSTEM?

An operating system – commonly referred to as an OS – is a set of computer programs that manage the hardware and software resources of a computer. The OS processes electronic devices with a rational response to commands that are approved by the system.

At the foundation of all system software, an operating system performs basic tasks like controlling and allocating memory, prioritizing system requests, controlling input and output devices, facilitating the network, and managing files. The OS can also provide a graphical user interface for higher functions. Essentially, the OS forms a platform for other system software as well as application software.

The operating system is the most important program that runs on a computer. Without an operating system, your computer would not work. It would not be able to process requests for print, simple calculations, or any other function. It is really the brain that runs the equipment.

For larger system, the OS has great responsibilities than with a PC. In larger systems, the operating system is kind of like a traffic cop. It makes sure that different users and programs running at the same time on different systems don't interfere with each other. It also acts as a security guard making sure that unauthorized users are not able to access the system.

There are four classifications of a computer operating system. They are:

- **Multi-User:** Allows two or more users to run programs at the same time. Some operating systems permit hundreds or even thousands of concurrent users

- **Multi-Processing:** Supports running a program on more than one CPU

- **Multi-Tasking:** Allows more than one program to run concurrently

- **Multi-Threading:** Allows different parts of a single program to run concurrently

- **Real Time:** Responds to input instantly. General-purpose operating systems, such as DOS and UNIX, are not real-time.

Operating systems provide a software platform on top of which other programs, called application programs, can run. The application programs must be written to run on top of a particular operating system.

Your choice of operating system, therefore, determines to a great extent the applications you can run. For PCs, the most popular operating systems are DOS, OS/2, and Windows, but others are available, such as Linux.

In any device that has an operating system, there's usually a way to make changes to how the device works. This is far from a happy accident; one of the reasons operating systems are made out of portable code rather than permanent physical circuits is so that they can be changed or modified without having to scrap the whole device.

For a desktop computer user, this means you can add a new security update, system patch, new application or often even a new operating system entirely rather than junk your computer and start again with a new one when you need to make a change.

As long as you understand how an operating system works and know how to get at it, you can in many cases change some of the ways it behaves. And, it's as true of your cell phone as it is of your computer.

So, essentially, when you turn on your computer, the first program is a set of instructions kept in the computer's read only memory. These instructions examine the system hardware to make sure everything is functioning properly. This power-on self test check the CPU, the memory, and the basic input/output systems (BIOS) for errors and stores the result in a special memory location.

Once the test has successfully completed, the software loaded in ROM (sometimes called the BIOS or firmware) will begin to activate the computer's disk drives. In most modern computers, when the computer activates the hard disk drive, it finds the first piece of the operating system: the bootstrap loader.

The bootstrap loader is a small program that has a single function: It loads the operating system into memory and allows it to begin operation. In the most basic form, the bootstrap loader sets up the small driver programs that interface with and control the various hardware subsystems of the computer.

It sets up the divisions of memory that hold the operating system, user information and applications. It establishes the data structures that will hold the myriad signals, flags and semaphores that are used to communicate within and between the subsystems and applications of the computer. Then it turns control of the computer over to the operating system.

It might be helpful for you to know the history of operating systems.

HISTORY OF OPERATING SYSTEMS

The earliest of computers didn't have an operating system. By the early 1960's, commercial computer vendors were supplying quite extensive tools for streamlining the development, scheduling and execution of jobs on batch processing systems.

Through the 1960's, several concepts were developed which drove the development of operating systems. The IBM System 360 produced a family of mainframe computer that served consumers with differing capacities and prices. A single operating system was planned for these computers rather than developing generic programs for every individual model.

This concept of a single OS that will fit an entire product line was crucial for the success of System 360. In fact, IBM's current mainframe operating systems are distant relatives of this original system. The advantage to this is that applications written for the OS 360 can still be run on modern machines.

The OS 360 system also contained another important advance affecting today's computers: the development of a hard disk permanent storage device which IBM called DASD.

A second key development was the concept of time sharing. Time sharing involves sharing the resources of expensive computers among multiple computer users interacting in real time with the system. What that essentially means is that all of the users have the illusion of exclusive access to the machine. The most famous of time sharing system was called Multics.

Multics served as an inspiration to a number of operating systems developed in the 1970's. Most notably was the Unix system. Another commercially popular mini-computer operating system was VMS.

The first microcomputers did not have the capacity or need for the elaborate operating systems that had originally been developed for mainframes and minis. Smaller operating systems were developed and often loaded from ROM and known as Monitors.

One notable early disk-based OS was CP/M which was supported on many early micro-computers and was largely cloned when MS-DOS was created. MS-DOS became wildly popular as the operating system chosen for the IBM PC.

The successive operating systems that came from MS-DOS made Microsoft one of the most profitable companies in the world with the development of Windows. The only other alternative throughout the 1980's was Mac OS which was tied intimately to the Apple McIntosh computer.

By the 1990s, the microcomputer had evolved to the point where it became increasingly desirable. Everyone wanted a home computer. Microsoft had already come out with Windows 95 and 98, but people longed for more power and more options. Microsoft's response to this change was the development of Windows NT which served as the basis for Microsoft's desktop operating system line that launched in 2001.

Apple was also rebuilding their own operating system on top of Unix core as Mac OS X also released in 2001 developing one of the business world's greatest rivalries.

Today, our operating systems usually have a graphical user interface (GUI) which uses a pointing device such as a mouse of stylus for input in addition to the keyboard. Older systems – and we mean REALLY OLD – use a command line interface asking for commands to be entered via the keyboard.

Both models are centered on a "shell" which accepts and processes commands from the user. The user may be asked to click on a button or type in a command upon an on-screen prompt.

By far, the most common operating system in use today is Windows XP, but Microsoft has just released their newest Windows project – Windows Vista. Linux is also another popular OS as is Unix. We'll explore them later on in the book, but each offers its own particular advantages and disadvantages.

Considering the boom of the technology market, it's really a surprise that there are so few operating systems in existence. There really isn't an easy explanation for this, but it is a reality. It would only seem logical that with all of the different computer manufacturers out there, there would be more of a choice for an OS than what there is. It is certainly another anomaly in the world of computer technology.

So what exactly do operating systems do? Since they really are the “brain” of the computer, they do quite a bit!

WHAT AN OPERATING SYSTEM DOES

As a user, you normally interact with the operating system through a set of commands. For example, the DOS operating system contains commands such as COPY and RENAME for copying files and changing the names of files, respectively.

The commands are accepted and executed by a part of the operating system called the command processor or command line interpreter. Graphical user interfaces allow you to enter commands by pointing and clicking at objects that appear on the screen.

But that really doesn't address the various ways that operating systems make your computer work easier and more efficiently. Their specific capacities are what make them help your computer operate as a user-friendly device. Let's look specifically at what an operating system does.

Process Management

Every program running on a computer whether it is a background service or an application is called a process. As long as von Neumann architecture is used to build a computer, only one process per CPU can be run at one time.

Older computer operating systems such as MS-DOS did not try to bypass this limit with the exception of interrupt processing and only one process could be run under them. Mainframe operating systems have had multi-tasking capabilities since the early 1960's. Modern

operating systems enable concurrent execution of many processes at once via multi-tasking even with one CPU.

Process management is an operating system's way of dealing with running multiple processes at once. Since most computers contain one processor with one core, multi-tasking is done by simply switching processes quickly. Depending on the operating system, as more processes run, either each time slice will become smaller or there will be a longer delay before each process given a chance to run.

Process management involves computing and distributing CPU time as well as other resources. Most operating systems allow a process to be assigned a priority which affects its allocation of CPU time. Interactive operating systems also employ some level of feedback in which the task with which the user is working receives higher priority.

Interrupt driven processes will normally run at a very high priority. In many systems, there is a background process such as the System Idle Process in Windows which will run when no other process is waiting for the CPU.

It's tempting to think of a process as an application, but that gives an incomplete picture of how processes relate to the operating system and hardware. The application you see (word processor, spreadsheet or game) is, indeed, a process, but that application may cause several other processes to begin, for tasks like communications with other devices or other computers.

There are also numerous processes that run without giving you direct evidence that they ever exist. For example, Windows XP and UNIX can have dozens of background processes running to handle the network, memory management, disk management, virus checking and so on.

A process, then, is software that performs some action and can be controlled -- by a user, by other applications or by the operating system.

It is processes, rather than applications, that the operating system controls and schedules for execution by the CPU. In a single-tasking system, the schedule is straightforward. The operating system allows the application to begin running, suspending the execution only long enough to deal with interrupts and user input.

Interrupts are special signals sent by hardware or software to the CPU. It's as if some part of the computer suddenly raised its hand to ask for the CPU's attention in a lively meeting. Sometimes the operating system will schedule the priority of processes so that interrupts are masked -- that is, the operating system will ignore the interrupts from some sources so that a particular job can be finished as quickly as possible.

There are some interrupts such as those from error conditions or problems with memory that are so important that they can't be ignored. These non-maskable interrupts (NMIs) must be dealt with immediately, regardless of the other tasks at hand.

While interrupts add some complication to the execution of processes in a single-tasking system, the job of the operating system becomes much more complicated in a multi-tasking system. Now, the operating system must arrange the execution of applications so that you believe that there are several things happening at once.

This is complicated because the CPU can only do one thing at a time. In order to give the appearance of lots of things happening at the same time, the operating system has to switch between different processes thousands of times a second. Here's how it happens:

- A process occupies a certain amount of RAM. It also makes use of registers, stacks and queues within the CPU and operating-system memory space.

- When two processes are multi-tasking, the operating system allots a certain number of CPU execution cycles to one program.

- After that number of cycles, the operating system makes copies of all the registers, stacks and queues used by the processes and note the point at which the process paused in its execution.

- It then loads all the registers, stacks and queues used by the second process and allow it a certain number of CPU cycles.

- When those are complete, it makes copies of all the registers, stacks and queues used by the second program, and load the first program.

All of the information needed to keep track of a process when switching is kept in a data package called a process control block. The process control block typically contains:

- An ID number that identifies the process
- Pointers to the locations in the program and its data where processing last occurred
- Register contents
- States of various flags and switches
- Pointers to the upper and lower bounds of the memory required for the process
- A list of files opened by the process
- The priority of the process
- The status of all I/O devices needed by the process

Each process has a status associated with it. Many processes consume no CPU time until they get some sort of input. For example, a

process might be waiting on a keystroke from the user. While it is waiting for the keystroke, it uses no CPU time. While it is waiting, it is “suspended”.

When the keystroke arrives, the OS changes its status. When the status of the process changes, from pending to active, for example, or from suspended to running, the information in the process control block must be used like the data in any other program to direct execution of the task-switching portion of the operating system.

This process swapping happens without direct user interference, and each process gets enough CPU cycles to accomplish its task in a reasonable amount of time. Trouble can come, though, if the user tries to have too many processes functioning at the same time. The operating system itself requires some CPU cycles to perform the saving and swapping of all the registers, queues and stacks of the application processes.

If enough processes are started, and if the operating system hasn't been carefully designed, the system can begin to use the vast majority of its available CPU cycles to swap between processes rather than run processes. When this happens, it's called thrashing, and it usually requires some sort of direct user intervention to stop processes and bring order back to the system.

One way that operating-system designers reduce the chance of thrashing is by reducing the need for new processes to perform various tasks. Some operating systems allow for a “process lite” called

a thread that can deal with all the CPU-intensive work of a normal process, but generally does not deal with the various types of I/O and does not establish structures requiring the extensive process control block of a regular process. A process may start many threads or other processes, but a thread cannot start a process.

So far, all the scheduling we've discussed has concerned a single CPU. In a system with two or more CPUs, the operating system must divide the workload among the CPUs, trying to balance the demands of the required processes with the available cycles on the different CPUs.

Asymmetric operating systems use one CPU for their own needs and divide application processes among the remaining CPUs. Symmetric operating systems divide themselves among the various CPUs, balancing demand versus CPU availability even when the operating system itself is all that's running.

Even if the operating system is the only software with execution needs, the CPU is not the only resource to be scheduled. Memory management is the next crucial step in making sure that all processes run smoothly.

Memory Management

The way computers are built, the memory is arranged in a hierarchal way. It starts with the fastest registers, the CPU cache, random access memory, and disk storage. An operating system's

memory manager coordinates the use of these various memory types by tracking which one is available, which one should be allocated or de-allocated and how to move data between them.

This function is referred to as virtual memory management and increases the amount of memory available for each process by making the disk storage seem like main memory. There is a speed penalty associated with using disks or other slower storage as memory. If running processes requires significantly more RAM than is available, the system may start “thrashing” or slowing down.

This can happen either because one process requires a large amount of RAM or because two or more processes compete for a larger amount of memory than is available. This then leads to constant transfer of each process’s data to slower storage.

Another important part of memory management is managing virtual addresses. If multiple processes are in the memory at the same time, they must be stopped from interfering with each other’s memory unless there is an explicit request to utilize shared memory. This is achieved by having separate address spaces.

Each process sees the whole virtual address space, typically from address 0 up to the maximum size of virtual memory as uniquely assigned to it. The operating system maintains a page tables that matches virtual addresses to physical addresses. These memory allocations are tracked so that when a process ends, all memory used by that process can be made available for other processes.

The operating system can also write inactive memory pages to secondary storage. This process is called “paging” or “swapping”. The terminology varies between operating system.

It is also typical for operating systems to employ otherwise unused physical memory as a page cache. The page cache contains requests data from a slower device and can be retained in memory to improve performance. The OS can also pre-load the in-memory cache with data that may be requested by the user in the near future.

The first task of memory management requires the operating system to set up memory boundaries for types of software and for individual applications.

As an example, let's look at an imaginary small system with 1 megabyte (1,000 kilobytes) of RAM. During the boot process, the operating system of our imaginary computer is designed to go to the top of available memory and then "back up" far enough to meet the needs of the operating system itself.

Let's say that the operating system needs 300 kilobytes to run. Now, the operating system goes to the bottom of the pool of RAM and starts building up with the various driver software required to control the hardware subsystems of the computer. In our imaginary computer, the drivers take up 200 kilobytes. So after getting the operating system completely loaded, there are 500 kilobytes remaining for application processes.

When applications begin to be loaded into memory, they are loaded in block sizes determined by the operating system. If the block size is 2 kilobytes, then every process that is loaded will be given a chunk of memory that is a multiple of 2 kilobytes in size. Applications will be loaded in these fixed block sizes, with the blocks starting and ending on boundaries established by words of 4 or 8 bytes.

These blocks and boundaries help to ensure that applications won't be loaded on top of one another's space by a poorly calculated bit or two. With that ensured, the larger question is what to do when the 500-kilobyte application space is filled.

In most computers, it's possible to add memory beyond the original capacity. For example, you might expand RAM from 1 to 2 megabytes. This works fine, but tends to be relatively expensive. It also ignores a fundamental fact of computing -- most of the information that an application stores in memory is not being used at any given moment.

A processor can only access memory one location at a time, so the vast majority of RAM is unused at any moment. Since disk space is cheap compared to RAM, then moving information in RAM to hard disk can greatly expand RAM space at no cost. This technique is called virtual memory management.

Disk storage is only one of the memory types that must be managed by the operating system, and is the slowest. Ranked in order

of speed, the types of memory in a computer system are:

- **High-speed cache** - This is fast, relatively small amounts of memory that are available to the CPU through the fastest connections. Cache controllers predict which pieces of data the CPU will need next and pull it from main memory into high-speed cache to speed up system performance.

- **Main memory** - This is the RAM that you see measured in megabytes when you buy a computer.

- **Secondary memory** - This is most often some sort of rotating magnetic storage that keeps applications and data available to be used, and serves as virtual RAM under the control of the operating system.

The operating system must balance the needs of the various processes with the availability of the different types of memory, moving data in blocks (called pages) between available memory as the schedule of processes dictates.

Disk and File Systems

All operating systems include support for a variety of file systems. Modern file systems are made up of directories. While the idea is similar in concept across all general purpose file systems, some differences in implementation exist.

Two examples of this are the character that is used to separate directories and case sensitivity. By default, Microsoft Windows separates its path components with a backslash and its file names are not case sensitive.

However, UNIX and Linux derived operating systems along with Mac OS use the forward slash and their file names are generally case sensitive. Some versions of Mac OS (those prior to OS X) use a color for a path separator.

File systems are either journaled or non-journaled. A journaled file system is a safer alternative in the event of a system crash. If a system comes to an abrupt stop in a crash scenario, the non-journaled system will need to be examined by the system check utilities. On the other hand, a journaled file systems recovery is automatic.

The file systems vary between operating systems, but common to all these is support for file systems typically found on removable media like CDs, DVDs, and floppy disks. They also provide for the rewriting of CDs and DVDs as storage mediums.

Networking

Another aspect of an operating system has to do with the networking capabilities contained in each. The network links separate computers together from different locations.

Most current operating systems are capable of using the TCP/IP networking protocols. That means that one system can appear on a network of the other and share resources such as files, printers, and scanners using either wired or wireless connections.

Security

Security is important in any computer system. The operating system provides a level of security that can protect your computer and the data on it. System security is based on two principles:

- That the operating system provides access to many resources either directly or indirectly. That could mean files on a local disk, privileged system calls, person information about user, and the services offered by the programs running on the system.

- That the operating system is capable of distinguishing between those who are allowed to access the resource and those who are forbidden to do so. While some systems may simply distinguish between “privileged” and “non-privileged”, most commonly have a form of register identity such as a user name.

Requesters of information are further divided into two categories:

- Internal security in an already running program. On some systems, once a program is running, it has no limitations, but

commonly, the program has an identity which it keeps. That identity is used to check all of its requests for resources.

- External security as in a new request from outside the computer. This could be in the form of a new request from outside the system such as a login at a connected console or some kind of network connection. To establish identity, there may be a process of authentication.

  Often a username must be entered and every username should have a password. Other methods of authentication such as magnetic cards or biometric data may be used instead. In some cases, especially connections from the network, resources may be accessed with no authentication at all.

In addition to the allow/disallow model of security, a system with a high level of security will also offer auditing options. These would allow tracking of requests for access to resources as in “who has been reading this file?”

Operating system security has long been a concern of programmers because of highly sensitive data held on some computers. This is both of a commercial and a military nature.

The US Government Department of Defense created their own criteria of standards that sets basic requirement for assessing the effectiveness of OS security. This became of vital importance to operating system makers because this system was used to classify and

select system being considered for the processing, storage ad retrieval of sensitive or classified information.

Internal Security

Internal security can be thought of as a way to protect the computer's resources from the programs concurrently running on the system. Most operating systems set programs running natively on the computer's processor. That brings on the problem of how to stop these programs from doing the same task and having the same privileges as the operating system which is just a program too.

Processors used for general purpose operating systems are automatically blocked from using certain hardware instructions such as those to read or write from external devices like disks. Instead, they have to ask the privileged program, or operating system kernel) to read to write. The operating system, therefore, gets the chance to check the program's identity and allow or refused the request.

An alternative strategy available in systems that don't meet pre-set requirements is the operating will not run user programs as native code. Instead, they either emulate a processor or provide a host for a "p-Code" based system such as Java.

Internal security is especially important with multi-user systems as it allows each user of the system to have private files that the other users cannot tamper with or read. Internal security is also vital if

auditing is to be of any use since a program can potentially bypass the operating system without bypass auditing.

External Security

Typically, an operating system offers various services to other network computers and users. These services are usually provided through ports or numbered access points beyond the operating systems network address. These services include offerings such as file sharing, print services, e-mail, web sites, and file transfer protocols (FTP).

At the front line of security are hardware devices known as firewalls. At the operating system level, there are a number of software firewalls available. Most modern operating systems include a software firewall which is enabled by default.

A software firewall can be configured to allow or deny network traffic to or from a service or application running on the operating system. Therefore, one can install and be running an insecure service such as Telnet or FTP and not have to be threatened by a security breach because the firewall would deny all traffic trying to connect to the service on that port.

Graphical User Interfaces

Today, most modern operating systems contain Graphical User Interfaces (GUIs). A few of the older ones tightly integrated the GUI to the kernel – one of the central components of the operating system. More modern operating systems are modular separating the graphics subsystem from the kernel.

A GUI is basically the pictures you see on the screen that help you navigate your computer. They include the icons and the menus. Many operating systems allow the user to install or create any user interface they desire.

Graphical user interfaces tend to change and evolve over time. For example, Windows has modified its user interface almost every time a new version of Windows is released. The Mac OS GUI changed dramatically with the introduction of Mac OS X in 2001.

Device Drivers

A device driver is a specific type of computer software developed to allow interaction with hardware devices. Typically, this constitutes an interface for communicating with the device through the specific computer bus or communications subsystem that the hardware is connected to.

Device drivers provide commands to and/or receiving data from the device and on the other end, the requisite interfaces to the operating system and software applications.

You cannot have a CD-ROM drive, for example, without a device driver for that specific piece of equipment. You have drivers for a printer, scanner, and even your mouse.

It is a specialized hardware-dependent program which is also operating system specific that enables another program – typically an operating system or applications software package or computer program running under the operating system kernel.

This allows the system to interact transparently with the hardware device and usually provides the requisite interrupt handling necessary for any time-dependent hardware interfacing needs.

The key design goal of device drivers is abstraction. Every model of hardware is different. Newer models also are released by manufacturers that provide more reliable or better performance and these newer models are often controlled differently.

Computers and their operating systems cannot be expected to know how to control every device both now and in the future. To solve this problem, operating systems essentially dictate how every type of device should be controlled. The function of the device driver is then to translate these OS mandated function calls into device specific calls.

In theory, a new device which is controlled in a new manner should function correctly is a suitable driver is available. This new driver will insure that the device appears to operate as usual from the operating system's point of view for any person.

Some operating systems come with pre-installed drivers or a variety of common drivers to choose from. When you buy a new piece of hardware such as a joy stick, they will often come with a disk that contains the device driver that you can install. Other drivers or updated drivers are available online at the manufacturer's website.

Application Interface

Just as drivers provide a way for applications to make use of hardware subsystems without having to know every detail of the hardware's operation, application program interfaces (APIs) let application programmers use functions of the computer and operating system without having to directly keep track of all the details in the CPU's operation. Let's look at the example of creating a hard disk file for holding data to see why this can be important.

A programmer writing an application to record data from a scientific instrument might want to allow the scientist to specify the name of the file created. The operating system might provide an API function named MakeFile for creating files. When writing the program, the programmer would insert a line that looks like this:

MakeFile [1, %Name, 2]

In this example, the instruction tells the operating system to create a file that will allow random access to its data (signified by the 1 -- the other option might be 0 for a serial file), will have a name typed in by the user (%Name) and will be a size that varies depending on how much data is stored in the file (signified by the 2 -- other options might be zero for a fixed size, and 1 for a file that grows as data is added but does not shrink when data is removed). Now, let's look at what the operating system does to turn the instruction into action.

The operating system sends a query to the disk drive to get the location of the first available free storage location.

With that information, the operating system creates an entry in the file system showing the beginning and ending locations of the file, the name of the file, the file type, whether the file has been archived, which users have permission to look at or modify the file, and the date and time of the file's creation.

The operating system writes information at the beginning of the file that identifies the file, sets up the type of access possible and includes other information that ties the file to the application. In all of this information, the queries to the disk drive and addresses of the beginning and ending point of the file are in formats heavily dependent on the manufacturer and model of the disk drive.

Because the programmer has written the program to use the API for disk storage, the programmer doesn't have to keep up with the

instruction codes, data types and response codes for every possible hard disk and tape drive. The operating system - connected to drivers for the various hardware subsystems - deals with the changing details of the hardware -- the programmer must simply write code for the API and trust the operating system to do the rest.

APIs have become one of the most hotly contested areas of the computer industry in recent years. Companies realize that programmers using their API will ultimately translate this into the ability to control and profit from a particular part of the industry. This is one of the reasons that so many companies have been willing to provide applications like readers or viewers to the public at no charge.

They know consumers will request that programs take advantage of the free readers, and application companies will be ready to pay royalties to allow their software to provide the functions requested by the consumers.

As we've stated before, there are operating systems in all sorts of products – not just computers. Cell phones, DVD recorders, and TiVo players also have operating systems, however, those OSs are not readily noticeable to the consumer and they do not have any control over them.

This might be a good time to review some of the computer operating systems that are on the market today.

MICROSOFT WINDOWS

Back in the late 1970's, two enterprising young computer programmers named Paul Allen and Bill Gates developed an adaptation for the BASIC computer language that would help run newly created personal computer just coming on the technology market. As with any technology, their original creation changed and grew.

The two friends decided they had the product and the capability to become successful, so they formed a company now known as Microsoft. Over the years, Microsoft has grown to a giant in the computer industry with successes never before seen by a "from scratch" endeavor.

Microsoft was responsible for the development of not only several computer languages like COBOL and PASCAL, but also for the development of the earliest operating system MS-DOS. In partnership with IBM, who was just introducing the personal computer to the individual consumer, all of the IBM computers used MS-DOS on their systems. The year was 1981.

Even though originally, the Apple Corporation was in competition with Microsoft and IBM, the company eventually began working on developing an operating system for the company's Macintosh personal computers.

Then, in 1985, an industry changing product was starting to evolve. This new operating system would perform many functions already in MS-DOS, but the difference would be that this new product would focus on "gooeys" Graphical User Interfaces.

The development of the GUI would change the world of computers making it easier for the everyday consumer to navigate their personal computer. The industry was changing – and it was changing fast!

Windows operating system made the world of personal computing accessible and easy for the everyday Joe. Now, even students in the schools were able to use personal computers for their school work and in class. No one knew just how far this new OS would take the world of computing technology.

Just like with any computer technology, changes are constantly being made to improve on the product. After the initial launch of Windows, several other versions evolved each one offering new options and new features and each with their own bugs and problems.

Some people weren't big fans of Windows because at times it seemed as if Microsoft would release the product prior to fully testing it. It became famed for crashes and bugs that would cause the system to behave erratically, but Microsoft addressed each problem promptly and Windows continued to be THE operating system on the market.

The release of Windows 3.1 operating system was revolutionary in that it offered users more options that couldn't be found with its predecessor MS-DOS. One of the most helpful innovations was adding the use of a mouse to navigate and manipulate data with one hand simply and easily. 3.1 also gave users the convenience of not having to memorize MS-DOS commands.

Windows 3.1 was the first product to fully utilize graphical user interface for ease of controlling what the computer would do. Windows also now allowed the user to multitask, meaning the user could now run multiple applications at once without having to close out of each program before running another.

The next major Windows product to hit the market was Windows 95 released in 1995 hence the name! New features included the following:

- **Plug and Play -** Allows hardware devices to be automatically installed into the computer with the proper software. Does not require jumpers to be played with

- **32 Bit** - 32-Bit operating system allowing the computer to run faster and more efficiently

- **Registry** - Combines the power of multiple configuration files into two files, allowing the system configurations to be located easier

- **Memory** - Windows 95 had an improved memory handling processes compared to Windows 3.11

- **Right mouse click** - Allows you new access and text manipulation by utilizing both buttons instead of one

- **CD-Player** - Enhanced CD-Player with improved usability and AutoPlay feature.

Windows 95 also had some extra software included Windows Explorer, Paint, Scan Disk, and Sound Recorder. Games were added as were system tools that would de-fragment the hard drive and allow you to back up files for use later.

Windows 95 was succeeded by Windows 98 released in – you guessed it – 1998! While the release of this OS wasn't as big as 95, the 98 version still contained some significant updates, fixes, and support for new peripherals.

- **Protection** - Windows 98 included additional protection for important files on your computer such as backing up your registry automatically.

- **Improved support -** Improved support for new devices such as AGP, DirectX, DVD, USB, MMX, etc.

- **FAT32** - Windows 98 had the capability of converting your drive to FAT32 without losing any information.

- **Interface** - Users of Windows 95 and NT would enjoy the same easy interface.

- **PnP** - Improved PnP support, to detect devices even better than Windows 95.

- **Internet Explorer 4.0 -** Included Internet Explorer 4.0

- **Customizable Taskbar** - Windows added many nice new features to the taskbar which 95 and NT did not have.

- **Plus! -** Included features only found in Microsoft Plus! free.

- **Active Desktop** - Included Active Desktop which allowed for users to customize their desktop with the look of the Internet.

Windows 98 also upgraded some of its security features and added Dr. Watson which is a diagnostic tool that will look for problems on your computer and then offer up a resolution automatically. They also added a maintenance wizard that allows you schedule certain maintenance tasks such as Scan Disk to be run once a week.

In keeping with the theme of naming versions of Windows after the year it was released, the next version was Windows 2000. Windows 2000 was known as the professional version and was geared toward business use. So, it was often referred to as Windows Professional.

Features of this new operating system included:

- Support for FAT16, FAT32 and NTFS.
- Increased uptime of the system and significantly fewer OS reboot scenarios.
- Windows Installer tracks applications and recognizes and replaces missing components.
- Protects memory of individual apps and processes to avoid a single app bringing the system down
- Encrypted File Systems protects sensitive data.
- Secure Virtual Private Networking (VPN) supports tunneling in to private LAN over public Internet.
- Personalized menus adapt to the way you work
- Multilingual version allows for User Interface and help to switch, based on logon.
- Includes broader support for high-speed networking devices, including Native ATM and cable modems.
- Supports Universal Serial Bus (USB) and IEEE 1394 for greater bandwidth devices.

While that might not sound significant to the everyday computer user, these new advancements made for a smoother running system with more capabilities than other Windows versions.

The year 2000 also saw the release of Windows ME or Windows Millennium. This version was meant as an upgrade for Windows 95 and 98 and was designed with end users in mind. Overall, Windows ME has the look and feel of Windows 98 with some additional fixes and features not available in previous operating systems.

While Windows ME includes some of the latest fixes and updates and some enticing new features, this update was recommended only for users that may find or want some of the new features or for users who are purchasing a new computer with this operating system included.

Key updated features include:

- **Revert back to backup of computer** - Windows ME allowed the user to automatically restore an old backup in case files are corrupted or deleted.

- **Protect important system files** - Windows Me allowed the user to protect important system files and would not allow these files to be modified by any type of other software.

- **Movie editor** - Allowed users to edit and or combine Microsoft movie files. Importing movies required additional hardware.

- **Windows Media Player** - Included Media Player 7, which enabled users a more advanced way of listening and organizing their media files.

And the next year – 2001 – saw the release of Windows XP which is the version most users have to date. The XP stands for experienced and combined the two major Windows operating systems into one. It is available in both a home as well as a professional edition.

Windows XP is designed more for users who may not be familiar with all of Windows features and has several new abilities to make the Windows experience easier for those users.

Windows XP includes various new features not found in previous versions of Microsoft Windows. Below is a listing of some of these new features.

- **New interface** - a completely new look and ability to change the look.

- **Updates** - new feature that automatically obtains updates from the Internet.

- **Internet Explorer 6** - Includes internet explorer 6 and new IM.

- **Multilingual support** - added support for different languages.

In addition to the above features, Windows XP does increase reliability when compared to previous versions of Microsoft Windows.

Finally, the most recent upgrade of Windows was just released in 2007. Called Windows Vista, this new version is intended as an upgrade to XP and 2000 users. While it does have many new features, this version is intended to give computer users an overall better experience with a dramatic new look.

Added features over previous Windows versions include:

- Windows Aero, a completely new GUI and unlike any previous version of Windows.

- Basic file backup and restore.

- Improved DVD support with the ability to easily create custom DVD movies.

- Easy transfer, a feature that allows you to easily transfer files from an older computer to the new computer.

- File encryption.

- Instant search available through all Explorer windows.
- Support for DirectX 10.
- Self-healing, the ability to automatically detect and correct problems that may be encountered on the computer.
- Shadow copy, a feature that allows you to recover deleted files.
- Improved photo gallery and control of photographs.
- Windows Sidebar and gadgets that allows you to add an almost endless list of different gadgets.
- More parental control.
- Improved Windows Calendar, with the ability to set tasks and appointments.
- And much more.

Some people who are Windows enthusiasts hail this new product as a step into a new technological era. Windows Vista is flashy, pretty, and impressive, but it comes with its own unique faults as well. Reviewers report that there are many device drivers lacking in the software and that its size requires a large amount of memory which can cause your computer to run slower and less efficiently.

All versions of Windows do come with the original MS-DOS operating system included in the background for those “old-schoolers” who will want to enter their own computer commands. All versions also come with a basic word processing program and Internet Explorer for surfing the net. You will find standard games such as solitaire in Windows as well.

Most computers today are outfitted with Windows XP as their operating system, but with the release of Vista, that will probably change in the near future. The Microsoft Windows operating system is the most popular choice among computer users today, but there are other operating systems.

Let’s take a look at the most popular operating system for Apple Macintosh computers.

MAC OS

In 1984, Apple Computer introduced the Apple Macintosh personal computer. The first version was the Macintosh 128K model which came bundled with the Mac OS operating system then known as the “System Software”. The Mac is often credited with popularizing the graphical user interface (GUI).

The Mac OS has been pre-installed on almost every Macintosh computer ever sold. The operating system is also sold separately from the computer just as with Microsoft Windows. The original Mac OS was heavily based on the Lisa OS previously released by Apple for the Lisa computer in 1983. It also used concepts from other operating systems previewed by Apple executives.

In 1984, Apple partnered with Microsoft in an agreement that would have Microsoft creating versions of Word and Excel for the Mac OS. For the majority of the 1980’s, the Mac OS lacked a large amount of compatible software, however, the introduction of System 7 saw more software becoming available for the platform.

System 6 was the first major revision of the operating system, although the Mac OS kernel was kept the same from the System 7 revision until Mac OS 9.

The Macintosh project started in early 1979 with Jeff Raskin who envisioned an easy-to-use low-cost computer for the average consumer. In September of ’79, Raskin was given permission to start hiring for the project.

In January of 1981, Steve Jobs completely took over the Macintosh project. Jobs and a number of Apple engineers visited

Xerox PARC in December of 1979 which was three months after the Lisa and Macintosh project had begun.

After hearing about the pioneering GUI technology being developed at Xerox PARC from former employees like Raskin, Jobs negotiated a visit to see the Xerox Alto computer and Smalltalk development tools in exchange for stock options. This was probably one of the best business moves Jobs had ever made.

The final Lisa and Macintosh operating systems used concepts from the Xerox Alto, but many elements of the GUI were creating by Apple including a menu bar and pop-up menus. Specifically, the click and drag concept was developed by Jeff Raskin.

Unlike the IBM PC which used 8 KB of system ROM for power-on self test and basic input/output chores, the Mac ROM was significantly larger at 64 KB and held key OS code. Andy Hertzfeld was responsible for most of the original coding. He was able to conserve some of the ROM space by interweaving some of the assembly language code.

In addition to coding the ROM, he also coded the kernel, the Macintosh Toolbox, and some of the desktop accessories as well. The icons of the operating system which represented folders and application software were designed by Susan Kare who later designed the icons for Microsoft Windows 3.0.

Apple was very strong in advertising this new machine. After it was created, they actually bought out all thirty-nine pages of advertisement space in Newsweek Magazine's November/December, 1984 edition. It worked incredibly well and the investment paid off as Macs began flying off the shelves.

The first version of Mac OS along with subsequent updates were different from other operating systems in that this OS didn't use command line interface but rather user friendly interface. Many people think that Windows was the first to employ GUI, but Mac had them beat.

Updates to the OS mostly focused on changes to the "finder" which is an application for file management which also displays the desktop. Prior to version 5, the finder could only run one application at a time. When version 5 was released, it contained multi-finder which could run several applications at once.

Time was given to background applications only when the foreground running applications gave it up in co-operative multitasking, but in fact most of them did via a clever change in the operating system's event handling.

System 5 also brought Color Quick Draw to the Mac II. This significantly altered the extent and design of the underlying graphics architecture but it is a credit to Apple that most users, and perhaps more importantly existing code, were largely unaware of this.

System Software 5 was also the first MAC operating system to be given a unified system software version number as opposed to the numbers used for the system and finder files.

In 1991, System 7 was released. It was the second major upgrade to the Mac OS adding a significant user interface overhaul, new applications, stability improvement, and many new features.

The most visible change was a new full-color user interface. Although this feature made for a visually appealing interface, it was

optional. On machines not capable of displaying color or those with their display preferences set to monochrome, the interface defaulted back to the black and white of previous versions. Only some interface elements were colorized: scrollbars had a new look but push buttons remained in black and white.

The biggest feature added in system 7 included the built-in co-operative multitasking. In system 6, this function was optional through the multi finder. System 7 also introduced aliases which are similar to shortcuts that were introduced in later versions of Windows.

System extensions were enhanced by being moved to their own subfolder. A subfolder in the system folder was also created for the control panels. A smaller update – dubbed system 7.5 – included the extensions manager, a previously third party program which simplified the process of enabling and disabling extensions.

System 7 moved the Mac to true 32-bit memory addressing necessary for the every-increasing amounts of RAM available. Earlier systems used the lower 24 bits for addressing and the upper 8 bits for flags. This was an effective solution for earlier Mac models with very limited amounts of RAM, but it became a liability later. Virtual memory support was also added as a separate, optional feature.

The Apple menu, home only to desk accessories in system 6 was made more general purpose: the user could now make often-used folders and applications – or anything else they desired – appear in the menu by placing aliases to them in an “Apple Menu Items” subfolder of the system folder.

The trash folder, under system 6 and earlier, would empty itself automatically when shutting down the computer or, if multi-finder were not running, when launching an application. System 7 re-implemented the trash as a special hidden folder allowing files to remain in it across reboots until the user deliberately chose the “Empty Trash” command.

There were some other “high level” additions in system 7. Many people felt that Apple dropped the ball on some of these additions and accused the company of not fully thinking through these updates. Microsoft was accused of the same thing with earlier versions of Windows as well.

One of the most confusing aspects of the Mac OS was the reliance on countless System Enablers to support new hardware which would prove to plague the Mac OS all the way to version 8 after which iMac introduced its “New World” architecture. Although the iMac itself requires a system enabler with OS 8 as other Macs released at that time, Macs released after the iMac do not require a system enabler.

Another problem encountered was that various system update extensions with inconsistent version numbering schemes. Overall stability and performance of the Mac OS gradually worsened during this time which introduced Power PC support and 68K emulation.

When version OS 7.6 was released, the stability of the operating system was much better. People began to fully embrace the Mac OS and their legitimacy returned as a popular operating system.

System 7 also saw the introduction of an interactive help application, and the addition of “Stickies” which were basically virtual Post-It notes. Many Mac users still have OS 7 on their Apples.

Two other versions would follow in OS 8 and OS 9 each improving on the previous version. Apple continued to develop updates to their operating system making it more stable and capable of more tasks working efficiently to bring Mac into the 21st century.

The most recent version and one that is used on new systems today is Mac OS X. This version provides a stable operating environment for the Mac PC and offers more flexibility than other systems. The graphics are updated with lots of color and a flashier look.

UNIX

The UNIX operating system was developed in the 60's and 70's by a group of AT & T employees at Bell Labs. Unix is used by both servers and workstations and is the basis for a wide variety of other operating systems.

The UNIX system was designed to be portable, multi-tasking and multi-user in a time-sharing configuration. There are various concepts that are unique to all UNIX systems. These concepts include:

- The use of plain text for storing data
- A hierarchical file system
- Treating devices and certain types of inter-process communication as files
- The use of a large number of small programs that can be strung together through a command line interpreter using "pipes" as opposed to a single monolithic program with the same functionality.

The operating system under UNIX consists of many of the utilities listed above along with the master control program which is called the "kernel". The kernel helps start and stop programs, handle the file system, take care of other common high level tasks that most programs share and schedule access to hardware to avoid conflicts if

two programs try to access the same resource or device simultaneously.

Besides the main kernel, UNIX systems also had micro-kernels which tried to reverse the growing size of kernels and return to a system in which most tasks were completed by smaller utilities.

In an era where a “normal” computer consisted of a hard disk for storage and a data terminal for input and output, the UNIX file model worked quite well as most input/output was linear. However, modern systems include networking and other new devices.

Describing a graphical user interface driven by mouse control in an “event driven” fashion didn’t work well under the old model. Work on systems supporting these new devices in the 1980’s led to facilities for non-blocking input/output forms of inter-process communications other than just pipes, as well as moving functionality such as network protocols out of the kernel.

Just as with other operating systems, the programming was updated periodically to add other features and to streamline processes that the system would run. Ironically, the importance of the UNIX system is quite far-reaching. In fact, some experts call it the most important system you’ll never use.

UNIX is mostly used by Internet servers and database servers. It is a very efficient multi-user, multi-tasking operating system traditionally used by large companies and educational institutions.

It is scalable from a small system right up to a mainframe-class system (all you need to do is add extra hardware), which makes it suitable for anyone looking for a low cost, reliable operating system.

For programmers it has a wonderful set of built-in utilities, a programmable shell (command/user interface) and a straight forward structure that makes it very easy to quickly produce quite complex programs. For end users, UNIX has a friendly graphical interface (called X Windows) and many business applications and games.

As we said, UNIX is used as a basis for other operating systems. One of those is Linux.

LINUX

The first Linux systems were completed in 1992 by combining system utilities and libraries from the GNU project which is another operating system we'll address next.

Predominantly known for its use in servers, Linux is used as an operating system for a wider variety of computer hardware than any other operating system including desktop computers, super computers, mainframes, and embedded devices such as cell phones. Linux is packaged for different uses in Linux distributions which contain the kernel along with a variety of other software packaged tailored to its intended use.

Linux alleges that people regard the system as suitable mostly for computer experts because mainstream computer magazine reports cannot explain what Linux is in a meaningful way as they lack real-life experience using it. Furthermore, the frictional cost of switching operating systems and lack of support for certain hardware and application programs designed for Microsoft Windows have been two factors that have inhibited adoption.

However, as of early 2007, significant progress in hardware compatibility has been made, and it is becoming increasingly common for hardware to work "out of the box" with many Linux distributions. Proponents and analysts attribute the relative success of Linux to its security, reliability, low cost, and freedom from vendor lock-in.

The primary difference between Linux and other contemporary operating systems is that the Linux kernel and other components are open source software. That means that users have permission to

study, change, and improve the software. They can then redistribute it in modified or unmodified form. This is usually done in a public and collaborative manner.

Linux is not the only such operating system, although it is the most well-known and widely used one. Some open source licenses are based on the principle of “copy left”, a kind of reciprocity: any work derived from a copy left piece of software must also be copy left itself. One of the advantages of open source is that it allows for rapid software bug detection and elimination which is important for correcting security exploits.

Another advantage of Linux as an operating system is inter-operability. That means, it can run software from other companies such as Mac and Windows. This makes it hugely advantageous to the open market as inter-operability in an operating system is rather uncommon as of late.

People have actually taken on the promotion of Linux in what might be considered almost a cult-like following. In many cities and regions, local associations known as Linux Users Groups. They seek to promote Linux and, by extension, the notion and reality of free software. They actually hold meetings and provide free demonstrations, training, technical support, and operating system installation to new users.

There are also many Internet communities that seek to provide support to Linux users and developers. Most distributions and open source projects have a chat room on the freenode IRC network. These chat rooms are open to anyone with an IRC client. Online forums are

another means for support with notable examples being www.linuxquestions.org.

Every established free software project and Linux distribution has one or more mailing lists. Commonly there will be a specific topic such as usage or development for a given list.

Although Linux is generally available free of charge, several large corporations have established business models that involve selling, supporting, and contributing to Linux and free software. The free software licenses on which Linux is based explicitly accommodate and encourage commercialization.

As of late, lively discussions among computer enthusiasts have arisen over which is the best operating system to use: Windows or Linux? In the past, free software products have been criticized for not going far enough to insure ease of use. However, some experts have declared that Linux is nearly equal to Windows for ease of use as well as compatibility with other programs.

Many Windows applications can be run on the Linux operating system. While there are not many games or applications that are available with Linux, there are still others that can run easily on the software.

GNU

Like Linux, GNU (pronounced guh-noo) is also a free software operating system. Its name is a recursive acronym for “GNU’s not Unix” which was chosen because while it is Unix based, it is freeware and contains no Unix code. As of 2007, GNU is being actively developed although a complete system has not yet been released.

The gentleman responsible for developing GNU is Richard Stallman, a former employee at MIT. He believed computer users should be free, as most were in the 1960s and 1970s; free to study the source code of the software they use, free to share the software with other people, free to modify the behavior of the software, and free to publish their modified versions of the software. This philosophy was published in March 1985 as the GNU Manifesto.

Much of the needed software had to be written from scratch, but existing compatible free software components were used. Most of GNU has been written by volunteers; some in their spare time, some paid by companies, educational institutions, and other non-profit organizations.

In 1992, the operating system was more or less finished except for the kernel. The GNU project had a microkernel, and to add the necessary Unix-kernel-like functionality to their microkernel, they were developing a project called “Hurd”. However, “Hurd” was still very incomplete.

That year, Linus Torvalds released his Unix-like kernel Linux as free software. The combination of the Linux kernel and the GNU system made for a whole, Unix-like free software operating system.

Linux-based variants of the GNU system became the most common way in which people use GNU.

As of 2005, Hurd is in slow development, and is now the official kernel of the GNU system.

OTHER OPERATING SYSTEMS

While we have covered the main operating systems above, there are still other minor systems to touch on. We won't go into great detail on these systems, but they are still worth mentioning.

Amiga computers have their own Amiga OS. It has unique hardware in Amiga DOS which is a disk operating system. Their windowing interface is called Intuition, and the graphical user interface is referred to as Workbench. While Amiga OS isn't as popular now as it used to be, there are still some Amiga computers out there running this operating system.

Solaris is a computer operating system developed by Sun Microsystems. It is one of the largest open source projects in the computer community. It continues to grow in features, members, and applications. It is an independent operating system much as Unix is and is available for use on many computer systems.

Digital audio players run on free software called Rockbox. It was released under the GNU General Public License. This is a relatively new operating system and was first implemented on Archos Studio player because of owner frustration with limitations in the manufacturer-supplied user interface and device operations. It is present or available for many different players including the new ones with multi-color video capabilities.

Apple Computers miniature MP3 players run on a version of Linux called ipod linux. Newer players like the iPod shuffle are supported by its own version of operating system as the Linux system isn't capable of supporting some of the new graphical abilities.

Most computer routers run on a Cisco operating system. While the specific OS may vary slightly between versions, the basic set-up is essentially the same.

There have been many, many operating systems that have been created over the years. They have all evolved over the years to accommodate new technologies, and they will continue that evolution as our computers and electronic devices change.

Windows Vista has been the most recent OS to be released. Mac will soon be releasing another version of Mac OS X as well dubbed "Leopard". As the open source OS market continues to grow and evolve, we are likely to see many more of these popping up as well.

The competition can be fierce. As of late, some users have put pressure on Dell to pre-load all of its PCs and laptops with Linux operating system. Whether or not this is a good move remains to be seen. Many, many computer users are used to Windows, and learning a new system could be detrimental to one of the largest computer manufacturers in the world.

Another movement has been taken on to have new computers come without an operating system but with a coupon for a free OS of their choosing. That way, users can decide for themselves which system they want to use. Microsoft isn't keen on this idea, as you can imagine, but we do have a free enterprise system in our country. While they do have an almost corner on the market, some believe that these other companies should be able to go after their "piece of the pie" as well.

If you think you want to change your current operating system or if you have a computer without an operating system, you'll need to know how to install the new OS.

INSTALLING AN OPERATING SYSTEM

You might think that installing an operating system would be a simple procedure. And you'd be right. However, you really should know what you're doing before you undertake this process. Even if you already know how to do it, it's always nice to have a refresher course.

The first thing you should before installing a new operating system is to back up your existing data. These programs basically take a picture image of everything on your hard drive. With them, you can restore your entire system even if your new OS doesn't leave a trace of your old OS.

To effectively and safely back up your system, it's always a good idea to get some type of software program that will accomplish this. You could try doing it yourself, but these programs make it much, much easier. Consider one of the following:

- **Acronis True Image 10 Home**

  Acronis True Image 10 Home creates the exact copy of your hard disk and allows you to instantly restore the entire machine including operating system, applications, and all the data in the event of a fatal system crash or virus attack — no reinstallations required!

  With the new version you also can easily back up your music, video, and digital photos, as well as Outlook e-mails, contacts, calendar, tasks, and user settings with just a few mouse clicks!

  Copy your entire PC, including the operating system, applications, user settings, and all data using patented disk imaging technology; backup your music, video, and digital photos; backup your outlook e-mails, contacts, calendar, tasks, etc.; and restore all settings for Microsoft Office, iTunes, Media

Player, and dozens of popular applications.

A free trial copy of this software can be downloaded at http://www.acronis.com/homecomputing/products/trueimage/

- **Norton Ghost**

Made by Symantec Corporation, the same people who make and distribute Norton Anti-Virus, Norton Ghost is awarding winning back-up software. You should know, however, that this software is only for Microsoft Windows XP and 2000 operating systems.

Like True Image, Ghost will save everything on your computer safely and efficiently. It will back up everything on your system including music, pictures, applications, settings, etc. in one easy step. It can also recover your system and data even when you can't restart your operating system.

You can download a free trial ware version of Norton Ghost at http://www.symantec.com/home_homeoffice/products/overview.jsp?pcid=br&pvid=ghost10.

- **SOS Online Backup**

With SOS Online Backup, you can start small, protecting a handful of really important files, and scale all the way up to 100GB. It keeps previous versions of files forever. And continuous backup means your files are always backed up.

It handles open files and performs continuous and scheduled backup. SOS backs up only differences for changed files and stores unlimited versions. It also has online access and file sharing.

A free trial version can be downloaded at http://www.sosonlinebackup.com/

- **Ghost for Linux**

If you are currently using Linux for your operating system and want to switch to another one, this is the program for you. Like other backup software programs, Ghost for Linux is a cloning

tool and was created by Symantec, the folks who brought us Norton Anti-Virus.

Your drive will be cloned using the click and clone tool. It has been said to be easy to use and very efficient.

A downloadable version of Ghost for Linux can be found at http://linux.softpedia.com/get/System/Hardware/Ghost-for-Linux-053.shtml

Once you have all of your files backed up and you are sure the disk contains all of your information, you'll then be ready to install your system. This is the easiest process of all.

If you have purchased an OS such as Windows, it will come with an installation disk. What you'll need to do is put the disk in your disk drive and then shut down your computer. Turn it back on, and the OS will begin installing from the boot. Follow any on-screen prompts and answer accordingly.

If you have downloaded your new OS from a freeware site such as those that offer Linux for free, you will need to save the program to your computer's "My Documents" area. Once it has downloaded, go to "My Documents" and double click on the file. You will probably then be directed to answer questions as to how to proceed.

Be aware that the installation process will probably take some time to complete. You need to monitor the progress, so expect to spend at least an hour during the install answering questions and watching it install.

Once your new system is installed, you should spend some time going through its features and performing tasks that you would normally do. Make sure that all your previous functions work. Keep in mind that if you have had certain programs and applications on your computer prior to the re-install, you will have to put them back on after the install.

For example, if you had a favorite game on the computer before, you will have lost it with the re-boot. You will have to re-install the game if you want to play it.

That's it for installing an operating system. It really is quite a simple process, but it can be time consuming, so be prepared before you proceed!

Now let's take a look at some of the functions contained in an operating system and what they mean. Sometimes computer terminology can be a bit confusing, so it's helpful to know what certain functions are all about!

DEFINING THE PROCESSES

Cache

When shopping for a computer, often the word “cache” will come up. There are two types of caches when it comes to modern computers: L1 and L2. Some now even have L3 caches. Caching is a very important process when it comes to your PC.

There are memory caches, hardware and software disk caches, page caches and more. Virtual memory is even a form of caching. Let’s look at what caching is and why it is so important.

Caching is a technology based on the memory subsystem of your computer. The main purpose of a cache is to accelerate your computer while keeping the price of the computer low. Caching allows you to do your computer tasks more rapidly.

To understand the basic idea behind a cache system, we can use a simple analogy using a librarian to demonstrate the caching process. Think of a librarian behind the desk. He or she is there to give you the books you ask for.

To keep it simple, let’s assume that you can’t get the books yourself, you have to ask the librarian for the book you want to read and he or she gets it for from you from shelving in a storeroom. This first example is a librarian without a cache.

The first person arrives and asks for the book Great Expectations. The librarian goes to the storeroom, gets the book, returns to the counter, and gives the book to the customer. Later, the borrower comes back to return the book. The librarian takes the book and returns it to the storeroom returning to the counter to wait for the next customer.

The next customer comes in and also asks for Great Expectations. The librarian has to return to the storeroom to get the same book he had already handled and give it to the client. So basically, the librarian has to make a complete round trip to fetch every book – even very popular ones that are requested frequently.

This isn't a very efficient system for our librarian, is it? However, there is a way to improve on this system. We can add a cache on the librarian.

To illustrate a cache, let's give the librarian a backpack into which he or she will be able to store, say, ten books. That would mean the librarian has a 10 book cache. In this backpack, he or she will put the books the customers return to him up to a maximum of ten. Now, let's go back and visit the first scenario with our cached librarian.

At the beginning of the day, the librarian's cache is empty. The first person arrives and asks for Great Expectations. So the librarian goes to the storeroom and gives it to the customer. When the customer return with the book, instead of going back to the storeroom, the librarian puts the book into the backpack making sure it isn't full first.

Another person arrives and asks for Great Expectations. Before going to the storeroom the librarian checks to see if the book is in the backpack already. Lo and behold, it is! Now all he or she has to do is take the book from the backpack and give it to the client. No extra energy is expended by the librarian, and the customer doesn't have to wait for that trip to the storeroom.

Let's assume that the customer asks for a title that's not in the backpack? In this case, the librarian is less efficient with a cache because he or she must take the time to look for the book in the backpack first.

That is why one of the challenges of cache design is to minimize the impact of cache searches. Modern hardware has reduced this time delay to practically zero. The time it takes for the librarian to look in the cache is much less than having to run to the storeroom, so time is saved automatically with a cache. The cache is small (just ten books) so the time it takes to notice a miss is only a tiny fraction of the time it takes to walk to the storeroom.

From this example you can see several important facts about caching:

- Cache technology is the use of a faster but smaller memory type to accelerate a slower but larger memory type.

- When using a cache, you must check the cache to see if an item is in there. If it is there, it's called a cache hit. If not, it is

called a cache miss and the computer must wait for a round trip from the larger, slower memory area.

- A cache has some maximum size that is much smaller than the larger storage area.

- It is possible to have multiple layers of cache. With our librarian example, the smaller but faster memory type is the backpack, and the storeroom represents the larger and slower memory type. This is a one-level cache.

  There might be another layer of cache consisting of a shelf that can hold 100 books behind the counter. The librarian can check the backpack, then the shelf and then the storeroom. This would be a two-level cache.

A computer is a machine in which we measure time in very small increments. When the microprocessor accesses the main memory (RAM), it does it in about 60 nanoseconds (60 billionths of a second). That's pretty fast, but it is much slower than the typical microprocessor. Microprocessors can have cycle times as short as 2 nanoseconds, so to a microprocessor 60 nanoseconds seems like an eternity.

What if we build a special memory bank in the motherboard, small but very fast (around 30 nanoseconds)? That's already two times faster than the main memory access. That's called a level 2 cache or

an L2 cache.

What if we build an even smaller but faster memory system directly into the microprocessor's chip? That way, this memory will be accessed at the speed of the microprocessor and not the speed of the memory bus. That's an L1 cache, which on a 233-megahertz (MHz) Pentium is 3.5 times faster than the L2 cache, which is two times faster than the access to main memory.

Some microprocessors have two levels of cache built right into the chip. In this case, the motherboard cache -- the cache that exists between the microprocessor and main system memory -- becomes level 3, or L3 cache.

There are a lot of subsystems in a computer; you can put cache between many of them to improve performance. Here's an example. We have the microprocessor (the fastest thing in the computer). Then there's the L1 cache that caches the L2 cache that caches the main memory which can be used (and is often used) as a cache for even slower peripherals like hard disks and CD-ROMs. The hard disks are also used to cache an even slower medium -- your Internet connection.

Your Internet connection is the slowest link in your computer. So your browser (Internet Explorer, Netscape, etc.) uses the hard disk to store HTML pages, putting them into a special folder on your disk.

The first time you ask for an HTML page, your browser renders it

and a copy of it is also stored on your disk. The next time you request access to this page, your browser checks if the date of the file on the Internet is newer than the one cached. If the date is the same, your browser uses the one on your hard disk instead of downloading it from Internet. In this case, the smaller but faster memory system is your hard disk and the larger and slower one is the Internet.

Cache can also be built directly on peripherals. Modern hard disks come with fast memory, around 512 kilobytes, hardwired to the hard disk. The computer doesn't directly use this memory -- the hard-disk controller does.

For the computer, these memory chips are the disk itself. The computer asks for data from the hard disk. The hard-disk controller checks into this memory prior to moving the mechanical parts of the hard disk. This is very slow compared to memory. If it finds the data that the computer asked for in the cache, it will return the data stored in the cache without actually accessing data on the disk itself, saving a lot of time.

Here's an experiment you can try. Your computer caches your floppy drive with main memory, and you can actually see it happening. Access a large file from your floppy -- for example, open a 300-kilobyte text file in a text editor.

The first time, you will see the light on your floppy turning on, and you will wait. The floppy disk is extremely slow, so it will take 20 seconds to load the file. Now, close the editor and open the same file

again. The second time (don't wait 30 minutes or do a lot of disk access between the two tries) you won't see the light turning on, and you won't wait.

The operating system checked into its memory cache for the floppy disk and found what it was looking for. So instead of waiting 20 seconds, the data was found in a memory subsystem much faster than when you first tried it. One access to the floppy disk takes 120 milliseconds, while one access to the main memory takes around 60 nanoseconds -- that's a lot faster. You could have run the same test on your hard disk, but it's more evident on the floppy drive because it's so slow.

To give you the big picture of it all, here's a list of a normal caching system:

- **L1 cache** - Memory accesses at full microprocessor speed (10 nanoseconds, 4 kilobytes to 16 kilobytes in size)
- **L2 cache** - Memory access of type SRAM (around 20 to 30 nanoseconds, 128 kilobytes to 512 kilobytes in size)
- **Main memory** - Memory access of type RAM (around 60 nanoseconds, 32 megabytes to 128 megabytes in size)
- **Hard disk** - Mechanical, slow (around 12 milliseconds, 1 gigabyte to 10 gigabytes in size)

- **Internet** - Incredibly slow (between 1 second and 3 days, unlimited size)

As you can see, the L1 cache caches the L2 cache, which caches the main memory, which can be used to cache the disk subsystems, and so on.

One common question asked at this point is, "Why not make all of the computer's memory run at the same speed as the L1 cache, so no caching would be required?" That would work, but it would be incredibly expensive. The idea behind caching is to use a small amount of expensive memory to speed up a large amount of slower, less-expensive memory.

In designing a computer, the goal is to allow the microprocessor to run at its full speed as inexpensively as possible. A 500-MHz chip goes through 500 million cycles in one second (one cycle every two nanoseconds). Without L1 and L2 caches, an access to the main memory takes 60 nanoseconds, or about 30 wasted cycles accessing memory.

When you think about it, it is kind of incredible that such relatively tiny amounts of memory can maximize the use of much larger amounts of memory. Think about a 256-kilobyte L2 cache that caches 64 megabytes of RAM. In this case, 256,000 bytes efficiently caches 64,000,000 bytes. Why does that work?

In computer science, there is a theoretical concept called locality of reference. It means that in a fairly large program, only small portions are ever used at any one time. As strange as it may seem, locality of reference works for the huge majority of programs. Even if the executable is 10 megabytes in size, only a handful of bytes from that program are in use at any one time, and their rate of repetition is very high.

Virtual Memory

Virtual memory is a common part of most operating systems on desktop computers. It has become so common because it provides a big benefit for users at a very low cost.

Most computers today have something like 32 or 64 megabytes of RAM available for the CPU to use. Unfortunately, that amount of RAM is not enough to run all of the programs that most users expect to run at once.

For example, if you load the operating system, an e-mail program, a Web browser and word processor into RAM simultaneously, 32 megabytes is not enough to hold it all. If there were no such thing as virtual memory, then once you filled up the available RAM your computer would have to say, "Sorry, you can not load any more applications. Please close another application to load a new one." With virtual memory, what the computer can do is look at RAM for areas that have not been used recently and copy them onto the hard disk.

This frees up space in RAM to load the new application.

Because this copying happens automatically, you don't even know it is happening, and it makes your computer feel like is has unlimited RAM space even though it only has 32 megabytes installed. Because hard disk space is so much cheaper than RAM chips, it also has a nice economic benefit.

The read/write speed of a hard drive is much slower than RAM, and the technology of a hard drive is not geared toward accessing small pieces of data at a time. If your system has to rely too heavily on virtual memory, you will notice a significant performance drop. The key is to have enough RAM to handle everything you tend to work on simultaneously -- then, the only time you "feel" the slowness of virtual memory is is when there's a slight pause when you're changing tasks. When that's the case, virtual memory is perfect.

When it is not the case, the operating system has to constantly swap information back and forth between RAM and the hard disk. This is called thrashing, and it can make your computer feel incredibly slow.

The area of the hard disk that stores the RAM image is called a page file. It holds pages of RAM on the hard disk, and the operating system moves data back and forth between the page file and RAM. On a Windows machine, page files have a .SWP extension.

Windows 98 is an example of a typical operating system that has virtual memory. Windows 98 has an intelligent virtual memory

manager that uses a default setting to help Windows allocate hard drive space for virtual memory as needed. For most circumstances, this should meet your needs, but you may want to manually configure virtual memory, especially if you have more than one physical hard drive or speed-critical applications.

System Resources

Many people can get confused about running out of “memory” when they get the message that the system resources are out of memory. In many cases, an "out of memory" message is misleading, since your whole system really did not run out of memory. What this really means is that certain systems in your computer are running low on memory.

Windows maintains an area of memory for operating system resources. The maximum size of this area is 128K, in two 64K areas. Windows uses this area of memory to store fonts, bitmaps, drop-down menu lists and other on-screen information used by each application.

When any program begins running, it uses up some space in the "system resources" area in memory. But, as you exit, some programs do not give back system resources they were temporarily using. Eventually the system will crash as it runs out of memory. The crash happens sometimes if you start and close many programs, even the same ones, without a periodic reboot. This is what Microsoft calls a resource leak or memory leak.

When you tell your system to exit a program, the program is supposed to give back the resources (memory) it was using. However, programs are written by humans and mistakes can happen. The program may not give back all of the resources to the operating system. This failing to "give back" is the "memory leak," eventually leading to a message that your computer is low on resources. Memory leaks can also be caused by programs that automatically load every time you boot your system.

The system resources problem is something you might have to live with until the misbehaving application is found. If you are sure a certain application is causing the problem, be sure to contact the software vendor.

The best preventive maintenance is to periodically reboot your system.

RAM

Random access memory (RAM) is the best known form of computer memory. RAM is considered "random access" because you can access any memory cell directly if you know the row and column that intersect at that cell.

The opposite of RAM is serial access memory (SAM). SAM stores data as a series of memory cells that can only be accessed

sequentially (like a cassette tape).

If the data is not in the current location, each memory cell is checked until the needed data is found. SAM works very well for memory buffers, where the data is normally stored in the order in which it will be used (a good example is the texture buffer memory on a video card). RAM data, on the other hand, can be accessed in any order.

Similar to a microprocessor, a memory chip is an integrated circuit (IC) made of millions of transistors and capacitors. In the most common form of computer memory, dynamic random access memory (DRAM), a transistor and a capacitor are paired to create a memory cell, which represents a single bit of data.

The capacitor holds the bit of information -- a 0 or a 1. The transistor acts as a switch that lets the control circuitry on the memory chip read the capacitor or change its state.

A capacitor is like a small bucket that is able to store electrons. To store a 1 in the memory cell, the bucket is filled with electrons. To store a 0, it is emptied. The problem with the capacitor's bucket is that it has a leak. In a matter of a few milliseconds a full bucket becomes empty.

Therefore, for dynamic memory to work, either the CPU or the memory controller has to come along and recharge all of the capacitors holding a 1 before they discharge. To do this, the memory

controller reads the memory and then writes it right back. This refresh operation happens automatically thousands of times per second.

This refresh operation is where dynamic RAM gets its name. Dynamic RAM has to be dynamically refreshed all of the time or it forgets what it is holding. The downside of all of this refreshing is that it takes time and slows down the memory.

Memory cells are etched onto a silicon wafer in an array of columns (bit lines) and rows (word lines). The intersection of a bit line and word line constitutes the address of the memory cell.

DRAM works by sending a charge through the appropriate column (CAS) to activate the transistor at each bit in the column. When writing, the row lines contain the state the capacitor should take on. When reading the sense-amplifier determines the level of charge in the capacitor.

If it is more than 50 percent, it reads it as a 1; otherwise it reads it as a 0. The counter tracks the refresh sequence based on which rows have been accessed in what order. The length of time necessary to do all this is so short that it is expressed in nanoseconds (billionths of a second). A memory chip rating of 70ns means that it takes 70 nanoseconds to completely read and recharge each cell.

Memory cells alone would be worthless without some way to get information in and out of them. So the memory cells have a whole support infrastructure of other specialized circuits. These circuits

perform functions such as:

- Identifying each row and column (row address select and column address select)
- Keeping track of the refresh sequence (counter)
- Reading and restoring the signal from a cell (sense amplifier)
- Telling a cell whether it should take a charge or not (write enable)

Other functions of the memory controller include a series of tasks that include identifying the type, speed and amount of memory and checking for errors.

Static RAM uses a completely different technology. In static RAM, a form of flip-flop holds each bit of memory. A flip-flop for a memory cell takes four or six transistors along with some wiring, but never has to be refreshed.

This makes static RAM significantly faster than dynamic RAM. However, because it has more parts, a static memory cell takes up a lot more space on a chip than a dynamic memory cell. Therefore, you get less memory per chip, and that makes static RAM a lot more expensive.

Static RAM is fast and expensive, and dynamic RAM is less expensive and slower. Static RAM is used to create the CPU's speed-sensitive cache. Dynamic RAM forms the larger system RAM space.

Memory chips in desktop computers originally used a pin configuration called dual inline package (DIP). This pin configuration could be soldered into holes on the computer's motherboard or plugged into a socket that was soldered on the motherboard. This method worked fine when computers typically operated on a couple of megabytes or less of RAM, but as the need for memory grew, the number of chips needing space on the motherboard increased.

The solution was to place the memory chips, along with all of the support components, on a separate printed circuit board (PCB) that could then be plugged into a special connector (memory bank) on the motherboard. Most of these chips use a small outline J-lead (SOJ) pin configuration, but quite a few manufacturers use the thin small outline package (TSOP) configuration as well.

The key difference between these newer pin types and the original DIP configuration is that SOJ and TSOP chips are surface-mounted to the PCB. In other words, the pins are soldered directly to the surface of the board, not inserted in holes or sockets.

Memory chips are normally only available as part of a card called a module. You've probably seen memory listed as 8x32 or 4x16. These numbers represent the number of the chips multiplied by the capacity of each individual chip, which is measured in megabits (Mb), or one

million bits.

Take the result and divide it by eight to get the number of megabytes on that module. For example, 4x32 means that the module has four 32-megabit chips. Multiply 4 by 32 and you get 128 megabits. Since we know that a byte has 8 bits, we need to divide our result of 128 by 8. Our result is 16 megabytes!

It's been said that you can never have enough money. The same holds true for RAM, especially if you do a lot of graphics-intensive work or gaming. Next to the CPU itself, RAM is the most important factor in computer performance. If you don't have enough, adding RAM can make more of a difference than getting a new CPU!

If your system responds slowly or accesses the hard drive constantly, then you need to add more RAM. If you are running Windows XP, Microsoft recommends 128MB as the minimum RAM requirement. At 64MB, you may experience frequent application problems.

For optimal performance with standard desktop applications, 256MB is recommended. If you are running Windows 95/98, you need a bare minimum of 32 MB, and your computer will work much better with 64 MB. Windows NT/2000 needs at least 64 MB, and it will take everything you can throw at it, so you'll probably want 128 MB or more.

Linux works happily on a system with only 4 MB of RAM. If you plan to add X-Windows or do much serious work, however, you'll

probably want 64 MB. Mac OS X systems should have a minimum of 128 MB, or for optimal performance, 512 MB.

The amount of RAM listed for each system above is estimated for normal usage -- accessing the Internet, word processing, standard home/office applications and light entertainment. If you do computer-aided design (CAD), 3-D modeling/animation or heavy data processing, or if you are a serious gamer, then you will probably need more RAM. You may also need more RAM if your computer acts as a server of some sort (Web pages, database, application, FTP or network).

Another question is how much VRAM you want on your video card. Almost all cards that you can buy today have at least 16 MB of RAM. This is normally enough to operate in a typical office environment. You should probably invest in a 32-MB or better graphics card if you want to do any of the following:

- Play realistic games
- Capture and edit video
- Create 3-D graphics
- Work in a high-resolution, full-color environment

- Design full-color illustrations

When shopping for video cards, remember that your monitor and computer must be capable of supporting the card you choose.

Computer Memory

You already know that the computer in front of you has memory. What you may not know is that most of the electronic items you use every day have some form of memory also. Here are just a few examples of the many items that use memory:

- Cell phones
- PDAs
- Game consoles
- Car radios
- VCRs
- TVs

Each of these devices uses different types of memory in different ways!

Although memory is technically any form of electronic storage, it is used most often to identify fast, temporary forms of storage. If your computer's CPU had to constantly access the hard drive to retrieve every piece of data it needs, it would operate very slowly. When the

information is kept in memory, the CPU can access it much more quickly. Most forms of memory are intended to store data temporarily.

The CPU accesses memory according to a distinct hierarchy. Whether it comes from permanent storage (the hard drive) or input (the keyboard) most data goes in random access memory (RAM) first. The CPU then stores pieces of data it will need to access, often in a cache, and maintains certain special instructions in the register.

All of the components in your computer, such as the CPU, the hard drive and the operating system, work together as a team, and memory is one of the most essential parts of this team. From the moment you turn your computer on until the time you shut it down, your CPU is constantly using memory. Let's take a look at a typical scenario:

- You turn the computer on.

- The computer loads data from read-only memory (ROM) and performs a power-on self-test (POST) to make sure all the major components are functioning properly. As part of this test, the memory controller checks all of the memory addresses with a quick read/write operation to ensure that there are no errors in the memory chips. Read/write means that data is written to a bit and then read from that bit.

- The computer loads the basic input/output system (BIOS) from ROM. The BIOS provides the most basic information

about storage devices, boot sequence, security, Plug and Play (auto device recognition) capability and a few other items.

- The computer loads the operating system (OS) from the hard drive into the system's RAM. Generally, the critical parts of the operating system are maintained in RAM as long as the computer is on. This allows the CPU to have immediate access to the operating system, which enhances the performance and functionality of the overall system.

- When you open an application, it is loaded into RAM. To conserve RAM usage, many applications load only the essential parts of the program initially and then load other pieces as needed.

- After an application is loaded, any files that are opened for use in that application are loaded into RAM.

- When you save a file and close the application, the file is written to the specified storage device, and then it and the application are purged from RAM.

In the list above, every time something is loaded or opened, it is placed into RAM. This simply means that it has been put in the computer's temporary storage area so that the CPU can access that information more easily.

The CPU requests the data it needs from RAM, processes it and writes new data back to RAM in a continuous cycle. In most

computers, this shuffling of data between the CPU and RAM happens millions of times every second.

When an application is closed, it and any accompanying files are usually purged (deleted) from RAM to make room for new data. If the changed files are not saved to a permanent storage device before being purged, they are lost.

Fast, powerful CPUs need quick and easy access to large amounts of data in order to maximize their performance. If the CPU cannot get to the data it needs, it literally stops and waits for it.

Modern CPUs running at speeds of about 1 gigahertz can consume massive amounts of data -- potentially billions of bytes per second. The problem that computer designers face is that memory that can keep up with a 1-gigahertz CPU is extremely expensive -- much more expensive than anyone can afford in large quantities.

Computer designers have solved the cost problem by "tiering" memory -- using expensive memory in small quantities and then backing it up with larger quantities of less expensive memory.

The cheapest form of read/write memory in wide use today is the hard disk. Hard disks provide large quantities of inexpensive, permanent storage. You can buy hard disk space for pennies per megabyte, but it can take a good bit of time (approaching a second) to read a megabyte off a hard disk. Because storage space on a hard disk is so cheap and plentiful, it forms the final stage of a CPUs memory

hierarchy, called virtual memory.

The next level of the hierarchy is RAM. The bit size of a CPU tells you how many bytes of information it can access from RAM at the same time. For example, a 16-bit CPU can process 2 bytes at a time (1 byte = 8 bits, so 16 bits = 2 bytes), and a 64-bit CPU can process 8 bytes at a time.

Megahertz (MHz) is a measure of a CPU's processing speed, or clock cycle, in millions per second. So, a 32-bit 800-MHz Pentium III can potentially process 4 bytes simultaneously, 800 million times per second (possibly more based on pipelining)! The goal of the memory system is to meet those requirements.

A computer's system RAM alone is not fast enough to match the speed of the CPU. That is why you need a cache (discussed later). However, the faster RAM is the better. Most chips today operate with a cycle rate of 50 to 70 nanoseconds. The read/write speed is typically a function of the type of RAM used, such as DRAM, SDRAM, RAMBUS.

System RAM speed is controlled by bus width and bus speed. Bus width refers to the number of bits that can be sent to the CPU simultaneously, and bus speed refers to the number of times a group of bits can be sent each second. A bus cycle occurs every time data travels from memory to the CPU.

For example, a 100-MHz 32-bit bus is theoretically capable of sending 4 bytes (32 bits divided by 8 = 4 bytes) of data to the CPU

100 million times per second, while a 66-MHz 16-bit bus can send 2 bytes of data 66 million times per second. If you do the math, you'll find that simply changing the bus width from 16 bits to 32 bits and the speed from 66 MHz to 100 MHz in our example allows for three times as much data (400 million bytes versus 132 million bytes) passing through to the CPU every second.

In reality, RAM doesn't usually operate at optimum speed. Latency changes the equation radically. Latency refers to the number of clock cycles needed to read a bit of information. For example, RAM rated at 100 MHz is capable of sending a bit in 0.00000001 seconds, but may take 0.00000005 seconds to start the read process for the first bit. To compensate for latency, CPUs uses a special technique called burst mode.

Burst mode depends on the expectation that data requested by the CPU will be stored in sequential memory cells. The memory controller anticipates that whatever the CPU is working on will continue to come from this same series of memory addresses, so it reads several consecutive bits of data together.

This means that only the first bit is subject to the full effect of latency; reading successive bits takes significantly less time. The rated burst mode of memory is normally expressed as four numbers separated by dashes.

The first number tells you the number of clock cycles needed to begin a read operation; the second, third and fourth numbers tell you

how many cycles are needed to read each consecutive bit in the row, also known as the word line. For example: 5-1-1-1 tells you that it takes five cycles to read the first bit and one cycle for each bit after that. Obviously, the lower these numbers are, the better the performance of the memory.

Burst mode is often used in conjunction with pipelining, another means of minimizing the effects of latency. Pipelining organizes data retrieval into a sort of assembly-line process. The memory controller simultaneously reads one or more words from memory, sends the current word or words to the CPU and writes one or more words to memory cells. Used together, burst mode and pipelining can dramatically reduce the lag caused by latency.

So why wouldn't you buy the fastest, widest memory you can get? The speed and width of the memory's bus should match the system's bus. You can use memory designed to work at 100 MHz in a 66-MHz system, but it will run at the 66-MHz speed of the bus so there is no advantage, and 32-bit memory won't fit on a 16-bit bus.

Even with a wide and fast bus, it still takes longer for data to get from the memory card to the CPU than it takes for the CPU to actually process the data. That's where caches come in.

CONCLUSION

Computer operating systems are the basic building blocks for computer users. It is the “brain” that allows the computer to operate and can be quite important.

As we have shown you, there are many types of operating systems, and which one you choose is a personal decision. Each OS has its own specific advantages and disadvantages.

What you need to do is look at the features each has to offer and then pick the one that best suits your needs. You don’t have to be stuck with Windows any more if you don’t want to be. Now you have a choice!

When you have a better understanding of what operating systems can offer, you’ll be better informed to make your choice. While the terminology can be complicated, we hope that you now understand that each system is really built with the user in mind which is the most important component to operating system designers.

So next time you log on to your computer, remember that behind the scenes there are many, many things going on that you can’t see. Try to imagine your computer usage without these things. It is truly magical to be living in the computer age. Your operating system just makes it easier and more enjoyable.

There’s no big secret to an operating system, but now, hopefully, you know more than you did before – now that operating systems have been uncovered!

The following websites were referenced in researching this book:

www.microsoft.com

www.wikipedia.org

www.howstuffworks.com

www.webopedia.com