Table W4.1 Operating Systems (page 98)

Mobile Device Operating Systems
- Microsoft Windows CE
- Microsoft Pocket PC
- Microsoft Windows NT Embedded 4.0
- Palm Operating System
- Embedded Linux

Desktop and Notebook Computer Operating Systems
- MS-DOS
- Windows 95, Windows 98
- Windows Millennium Edition (ME)
- Windows NT
- Windows 2000
- Windows XP
- IBM's OS/2
- UNIX
- Linux
- Mac OS X
- JavaOS

Departmental Server Operating Systems
- UNIX
- Linux
- Windows 2000
- Windows XP
- Novell NetWare

Enterprise Server Operating Systems
- IBM's OS/400
- IBM's z/OS (formerly OS/390)

Supercomputer Operating Systems
- Cray Unicos
- IBM's AIX

MOBILE DEVICE OPERATING SYSTEMS

(page 98)

The market for mobile device operating systems includes software for a variety of devices, such as handheld computers, set-top boxes, subnotebook PCs, mobile telephones, and factory-floor equipment. In the past, each device used its own operating system (although the software may have been written by an independent software vendor that licensed it to the device manufacturer). The buyer had no choice but to use the operating system the maker put into it. This situation created a fragmented market for mobile device operating systems. It also prevented a large market for applications from developing, in sharp contrast to the PC industry. The mobile device operating system market has begun to consolidate around a handful of high-volume operating systems, including:

- Microsoft's Windows CE, including the Pocket PC variant of CE
- Palm OS from Palm
- Symbian's EPOC operating system
- Embedded Linux, an OS originally developed for servers and desktop computers

For discussion of these operating systems, see the material below.

Microsoft's Windows CE, Pocket PC, and Windows NT Embedded 4.0

Microsoft offers three operating systems for embedded and mobile devices. These operating systems allow developers to create customized versions of the OS that let them omit unneeded capabilities and functions but include those necessary to exploit a particular device. All are aimed at 32-bit devices only.

The main differences among the three systems are that Windows NT Embedded is more powerful and is designed to build server appliances and systems that need to provide or consume services typically associated with a server, such as printer and file sharing, log-in and authentication, and Web page serving. Thus, the NT product would be used for such devices as set-top boxes, Web-server appliances, and possibly ATMs or Web-pad devices. Windows CE is an appropriate product for PDAs, Web-access devices, video-game platforms, mobile phones, and intelligent appliances. (Pocket PC is a variant of Windows CE.) Microsoft plans to replace Windows CE, Pocket PC, and Windows NT Embedded with Windows XP.

Windows CE

Windows CE is Microsoft's information appliance operating system that has been incorporated into entertainment devices, subnotebook products, handheld PCs, palm-size PCs, cordless and mobile phones, Windows terminals, some set-top boxes, and an automobile information and entertainment system (AutoPC). Windows CE is the modular embedded OS for the new generation of highly connected 32-bit devices that seamlessly integrate with both Windows and the Internet. The standard Windows CE software package for handheld PCs includes scaled-down versions (known as pocket versions) of Microsoft Word, Excel, and PowerPoint; Pocket Internet Explorer; an e-mail client; a personal information manager (PIM) that synchronizes with Microsoft Schedule+; and Outlook.

Using these pocket applications to edit documents originating on a PC and then sending them back to the PC often causes loss of most formatting and is therefore somewhat problematic. The package has a program that automatically detects when the handheld PC is docked with a Windows computer and initiates synchronization and replication between Information Manager and Microsoft Schedule+.

Future versions of Microsoft's embedded operating systems for handhelds will contain voice-recognition and pen software. Another technology in future versions will be "Tap and Talk," which lets the user point at a place on the screen with a stylus or pen and then speak to supply input for that area of the screen.

Pocket PC

In early 2000, Microsoft introduced a version of its Windows mobile operating system called Pocket PC. The Pocket PC operating system is a version of Windows CE 3.0 with features and an interface designed specifically for PDAs and handheld computers. (Pocket PC is a specific implementation of Windows CE optimized for these devices.) Pocket PC solves many problems that held back the success of previous versions of Windows CE, such as an overly complicated interface, slow performance, and poor power management that shortened battery life. Pocket PC supports natural handwriting recognition with a variety of programs. A utility named Transcriber takes words or phrases entered anywhere on the screen and translates them into text. Analysts say that using fuzzy logic with an integrated dictionary yields high accuracy.

Microsoft added new features to Pocket PC—devices with Pocket PC contain miniversions of Internet Explorer (which offers full Web access), Word, Excel, Outlook, the Microsoft Reader, Windows Media Player, and Money. One cannot usually alter documents using these versions and expect formatting to be preserved—indeed some allow no editing—but existing documents and attachments can be read.

Palm Operating System

Palm developed this operating system for its PalmPilot handheld, pen-input PDAs. Palm OS includes a graphical user interface and an interapplication drag-and-drop feature. Users must learn a stylized alphabet, called Graffiti, to make the device receive handwritten input. Palm OS is optimized for mobile computing and takes less memory and runs faster than Windows CE, Java, and EPOC. Many industry experts consider Palm OS the best for small devices, because of its speed and modest power requirements.

The Palm OS has many features:

- An agenda view where people can see both their schedule and a "to-do" list at the same time
- A shortcut command bar to access features such as cut, copy, paste, delete, and beam
- Tappable menus, so people can activate menus with a quick tap on the menu text at the top of their screens
- New privacy protections
- A faster way of creating Address Book entries, so that records of people from the same company or group can have common information duplicated more quickly
- Faster HotSync capabilities
- A "sleep" button that allows the user to delay an appointment alert from the calendar while continuing to use other product features
- Infrared HotSync capabilities that eliminate the need for a docking cradle

Symbian's EPOC

The Symbian operating system, EPOC-32, has been powering Psion handheld devices for several years. It is a full 32-bit operating system written for ARM (Advanced RISC Microprocessor) processors, which constitute the majority of processors used in mobile phones.

MOBILE DEVICE OPERATING ENVIRONMENTS (page 98)

Hewlett-Packard's Chai is an integrated suite of software products, development tools, and support services. Chai is intended to help device manufacturers design information appliances. With Chai, a manufacturer can select components from the Chai Appliance Platform and drop them into a device. Any class of Java-enabled devices can be developed, including handheld phones, digital set-top boxes, high-speed network routers, factory automation controllers, and intelligent products.

Qualcomm's BREW (Binary Runtime Environment for Wireless) platform allows developers to create applications that operate on all handsets with Qualcomm CDMA chip sets. BREW sits between the chip system software and the application, making the phone's functionality available to the application without requiring the developer to have the chip system source code or even a direct relationship with a handset manufacturer. BREW allows rapid development of a variety of small, transportable handset applications that users can download over carrier networks onto any BREW-enabled phone. BREW enables consumers to download, update, and delete applications from their handsets, thereby bringing to wireless handsets the same choice and control that users have on their desktop computers. BREW supports e-mail and instant messaging, real-time navigation services, group chat, group games, and news and information services tailored to the individual.

Sun's Java Embedded Environment is Sun Microsystems' operating environment for embedded platforms and is based on the company's Java technology.

DESKTOP AND NOTEBOOK COMPUTER OPERATING SYSTEMS

(page 100)

The History of MS-DOS and Windows

Designed for the Client

MS-DOS

Windows 1.0 through Windows 3.1 (operating environments)

Windows 95

Windows 98

Windows Millennium Edition (ME)

Designed for Workstations and Servers

Windows NT 1.0 through Windows NT 4.0

Windows NT 5.0 (renamed Windows 2000)

Windows 2000
- Professional
- Standard Edition Server
- Advanced Server
- Datacenter Server

Windows XP
- Home Edition
- Professional (32-bit)
- Professional (64-bit)

Blackcomb

Windows Family Overview

Each of the current Windows desktop products is part of one of the two major Windows product families: Windows 95/98/Millennium, sometimes referred to collectively as "Windows 9x," and Windows NT/2000.

Windows 9x is software that has its origins in MS-DOS and the early versions of Windows. Although Windows 95 was a new development effort rather than an update to earlier versions of Windows, it retained many attributes of the earlier versions. For example, it lacked preemptive multitasking capability and protected memory and still contained some 16-bit code. Windows 98 and Windows Millennium Edition (typically referred to as "Windows ME") are based on Windows 95 and have inherited these attributes.

Windows Millennium Edition (Windows ME) is the first major update to the Windows 95 product family since Windows 98 was released in June 1998. Windows ME offered improvements for home computing in the areas of PC reliability, digital media, home networking, and the online experience. Highlights of Windows ME include:

- In the area of PC reliability, Windows ME provides such features as System Restore, which enables users to roll back to a previous PC configuration that worked properly.
- System File Protection, which prevents the accidental or unauthorized overwriting of crucial system files.
- AutoUpdate, which downloads Microsoft updates automatically without user intervention.
- Faster bootup.
- Digital media. Features in this area include:
  - Windows Image Acquisition (WIA), which helps users transfer digital images from digital scanners, cameras, and other WIA-compliant devices
  - Windows Movie Maker, which includes tools for editing, saving, and sharing digital home videos
  - Windows Media Player
  - DirectPlay Voice Chat, which allows gamers to chat across the Internet while playing an interactive game
- Home networking:
  - Home Networking Wizard
  - Improved Internet Connection Sharing (ICS) technology
  - Simplified USB networking and universal plug-and-play technology
- Online experience:
  - Windows ME improvements include enhanced browsing, shopping, Web publishing, and chatting, as well as Internet Explorer 5.5.

Windows NT and its follow-on product **Windows 2000**, on the other hand, were written from the beginning to be full multitasking operating systems based exclusively on 32-bit code. As a result, Windows NT/2000 is able to run multiple programs better, less prone to systems crashes, and more secure than the Windows 9x products. Windows 2000 Professional desktop OS and its planned upgrades, Windows XP Professional and Windows XP Home Edition, all shipped during the second half of 2001.

There are two major target market segments for desktop Windows: the consumer and the business user. During the latter part of the 1990s, MS attempted to position Windows NT as its preferred solution for the business user. In reality, however, the vast majority of business users ran Windows 95 or 98 on their desktops. Notebook computer users did not have much choice because Windows NT was not well-suited to running on notebooks—it lacked the power management and plug-and-play features notebooks need. With the release of *Windows 2000 Professional* (the desktop version of Windows 2000), Microsoft had a more attractive product to offer the mainstream business customer (including notebook users). With the release of *Windows XP*, the next-generation version of Windows 2000, Microsoft is in the process of abandoning the Windows 9x product family.

Windows 2000 Professional (32-bit) is a major upgrade of Windows NT. Windows 2000 was originally known as Windows NT 5.0. Four editions of Windows 2000 shipped in 2000: Professional, Standard Edition Server, Advanced Server, and Datacenter Server. Microsoft considers the Windows 2000 family its most vital operating system upgrade.

Windows 2000 is a more manageable operating system than previous desktop versions of Windows and provides centralized, policy-based management, broad support for laptops and mobile devices, and improved performance. Windows 2000 also features the following:

- Active Directory—IntelliMirror copies desktop settings automatically to a networked Windows 2000 server. Active Directory allows users to access applications and configurations from any Windows 2000 Professional desktop on the network even if the network connection fails, and makes it easier for administrators to troubleshoot and repair Windows 2000 desktops.
- Improvements for notebook use:
  - Mobile features include Hibernate, which turns off the computer and restores programs and settings when the user reactivates.
  - Offline access to files and folders and offline viewing of Web documents, which allows users to disconnect from the network and work and view Web pages as if still connected.
  - A synchronization manager for files and documents that are being used offline so they can be kept consistent with the copies stored on the network.
  - A smart battery meter, which gives a more accurate view of battery life and lets users reduce power to specific functions to extend battery strength.
  - Hot docking and easier remote configuration features.
- Direct X 7.0 improves video performance.
- Distributed File System: This feature permits multiple disk drives to be grouped into a single, virtual disk drive, allowing users to find files more easily without searching multiple drives.
- File Allocation Table 32 (FAT32) file system:
  - The FAT32 file system improves Windows 2000 Professional's compatibility with Windows 98/Millennium Edition.
  - This feature is provided in addition to the NT file system and the original 16-bit FAT format.
- Enhanced security:
  - Windows 2000 Professional provides file-level data encryption and implements Kerberos, a platform-independent security mechanism for user authentication and access control in a networked computer environment.
- Microsoft management console provides users with the ability to conduct single-interface management.
- Advanced networking hardware:
  - Software support for new networking hardware, including digital subscriber line (DSL) modems, cable modems, wireless technologies, and improved support for peripherals connected by means of universal serial bus (USB), FireWire, and infrared.
- Internet capabilities:
  - Windows 2000 Professional's Internet features include Internet Explorer 5.01, optional Internet Information Services (IIS) 5.0, Dynamic Hypertext Markup Language (DHTML) Behaviors and Extensible Markup Language (XML) support for developers, and Internet connection-sharing technologies.
- Faster multitasking and faster performance.
- Full 32-bit architecture:
  - The presence of full 32-bit architecture allows users to run more programs and perform more tasks simultaneously than with Windows 95 or Windows 98.
  - It also provides 25 percent faster performance than with Windows 9x on systems with 64 megabytes or more of memory.
- Scalable memory and processor support: Supports up to 4 gigabytes of RAM and up to two processors in a symmetric multiprocessor (SMP) configuration.

Windows XP Home Edition and Windows XP Professional (32-bit) is the first upgrade to Windows 2000, originally known by code name of "Whistler." Windows XP has three versions: a 32-bit consumer version, a 32-bit business version, and a 64-bit business version.

Windows XP will be the first product descended from Windows NT that is designed for both business and consumer users, and the first version of Windows that integrates features of Microsoft's .NET platform. Windows XP Home edition is aimed at home users and gamers. Windows XP Home Edition is viewed primarily as a 32-bit consumer replacement for Windows ME. Microsoft emphasizes Internet- and network-oriented features, such as a redesigned Weblike user interface with user-selectable Visual Styles; an emphasis on digital media; the Internet Explorer 6.0 browser; and basic support for Microsoft's forthcoming .NET platform. For example, Windows XP will enable users to save files to a Microsoft Internet storage site. Windows XP will offer the same features as Windows ME and also feature simplified security/log-in, an enhanced home-networking wizard, and the ability for home users to use multiple configuration profiles simultaneously.

Windows XP Professional has a 32-bit version and a 64-bit version. Windows XP Professional is considered a minor upgrade to Windows 2000 Professional. However, many of the features being introduced in Windows XP Professional will be of value to business users. The user-selectable visual design will enable task-based computing, with a task bar that provides a summary function via a pop-up that lists each e-mail message, document, or spreadsheet currently open. Windows XP will also feature the inclusion of smart agents, enhanced memory management, and better support for PC companion devices such as Web pads, personal digital assistants, and mobile phones. It also will provide support for Microsoft's Visual Studio.NET programming environment. This support will allow developers to design applications that exploit the .NET features of the client.

Windows XP Professional (64-bit) is designed to accompany Intel's Pentium 4 microprocessor. This upgrade will feature large memory support, features designed to provide high availability, support for new floating-point numerics, and multimedia-specific instructions. The 64-bit Windows is targeted at demanding applications, including e-commerce, data mining, memory-intensive high-end graphics, complex mathematics, and high-performance multimedia. The 64-bit operating system maintains backward compatibility with current 32-bit applications.

Following Windows XP, Microsoft will release its first fully .NET-enabled Windows operating system—code named **Blackcomb**—which can be considered version 3.0 of Windows 2000. On the client side, Blackcomb will provide services of interest to both consumers and business users. For example, Blackcomb will feature what are known as natural interfaces, including speech recognition and handwriting support.

Blackcomb will also feature seamless integration with XML-based Web services currently under construction by Microsoft and other firms. Over time, these Web services, such as the authentication and digital-rights-management elements of Microsoft's forthcoming Hailstorm services, will be able to tap into the Windows 2000 Active Directory, essentially melding Internet services with Microsoft's back-end directory services. Blackcomb will also be modularized to support non-PC devices. Microsoft is working on services for Blackcomb: a new storage system, a series of PC-based and non-PC-based user interfaces, building-block Web services, and a new programming environment, C#, that will support non-Windows environments such as Solaris and Linux and is designed to compete with Sun's Java.

Linux

Linux is based on UNIX and, like the Apache Web server, is open source freeware that can be downloaded at no cost from the Internet. Linux's designer, Linus Torvalds, and other freeware developers continue to enhance the core Linux operating system kernel.

As part of the GNU Network Object Model Environment (GNOME) Foundation, HP, IBM, and Sun are committed to developing a set of standard APIs to promote Linux as a desktop alternative to Windows, and they have embraced the GNOME interface as a means of doing so. Leading system and software vendors announced initiatives in 2000 designed to advance Linux on the desktop to compete with Microsoft Windows, which currently controls about 90 percent of desktops worldwide.

A number of graphical user interfaces are available for use with Linux. The two best known are the K Desktop Environment (KDE) and GNOME, both of which are based on the X Windows System. KDE 2.0 incorporates an office productivity suite and an array of new technologies, such as Konqueror, a full-featured Web browser and file manager, and Koffice, an integrated office suite. The GNOME Foundation's GNOME 1.4 was released in mid-2001.

The GNOME Foundation is sponsoring various projects and has adopted Sun's StarOffice productivity suite as the core of the office productivity software for GNOME Office. Sun, which acquired Star Division's StarOffice in 1999, has made the office productivity suite available to the open source community.

Another GNOME Foundation member, Eazel (founded by a group of former Macintosh engineers from Apple Computer), have developed a Linux GUI and file manager product called Nautilus 1.0 that offers a Windows-like interface, file management services, and integrated Web services for the Linux desktop. Nautilus is a core component of the GNOME 1.4 desktop project. The GNOME 1.4 desktop will be bundled with Red Hat and other Linux distributions. Dell and Sun have committed to bundling Nautilus on workstations in 2001. Sun, for example, will distribute Eazel's Nautilus software on the upcoming GNOME 2.0 desktop user environment for Solaris.

The Apple Macintosh Operating System

The current desktop version of Apple's flagship **Macintosh operating system** is Mac OS X, released in 2001. Mac OS X, a desktop operating system, is built on an open source, UNIX-based core operating system called Darwin. It features a new Aqua user interface, advanced graphics, true memory protection, preemptive multitasking, and SMP when running on the dual-processor Power Mac G4 systems.

The new "liquidlike" Aqua user interface offers color, depth, translucence, and fluid motion with new controls that resemble polished gems and buttons that indicate active or nonactive status by glowing and dimming. The GUI also offers new ease-of-use features, including customizable toolbars, navigation aids, the ability to save documents in Adobe Systems' Portable Document Format (PDF), and a new Dock feature that makes it easier for users to organize applications, documents, and windows.

The Dock feature contains many new user-friendly features, including pop-up menus, customizable toolbars, navigation aids, and the ability to print PDF files. The Dock tray holds folders, applications, documents, storage devices, minimized windows, digital images, links to Web sites, and e-mail.

Another major feature for end users, the redesigned Finder, offers simple navigation and large buttons to applications, documents, home directories, and Internet favorites in a single window. The File Viewer gives users three options for viewing the file system, including the icon and list views of Mac OS 9. By double-clicking on items in either the icon or list view, a view of the new folder appears in place of the older folder to reduce screen clutter and provide a single window view.

Mac OS X also features a new Quartz two-dimensional graphics engine, OpenGL for three-dimensional graphics and gaming, and QuickTime for streaming video and audio. Quartz provides on-the-fly rendering of documents stored in PDF and dramatically renders sharper graphic elements. OpenGL has become the industry's most widely used and supported 2-D and 3-D graphics API.

Mac OS X also includes features for improved integration with UNIX systems. It extends support to UNIX File System, the standard volume format of most UNIX operating systems, and to POSIX file system semantics, which makes it easier for developers to bring server applications to the new operating system.

Mac OS X also includes clients for the Apple File Protocol, Network File System (NFS, the dominant file-sharing protocol in the UNIX world), and Web-based Distributed Authoring and Versioning (WebDAV, an emerging Internet standard for accessing and sharing files over the Web). Mac OS X incorporates a version of the Apache Web server for file sharing and support for the Java 2 Platform, Standard Edition.

Mac OS X is a major upgrade of Mac OS 9. Mac OS X provides these major improvements in performance and usability via the following:

- Protected memory that allocates unique address space for each application or process on the computer
- A Virtual Memory Manager that automatically allocates the amount of memory needed by an application
- Built-in support for Point-to-Point Protocol (PPP) for remote access as well as AppleTalk support for interoperability with existing Macintosh networks
- Ability to embed and manipulate PDF data in any Mac OS X application
- QuickTime support
- Larger icons (up to 128 $\times$ 128 pixels)
- Improved screen space management
- New e-mail application
- Easier switching between applications
- Column views for easier navigation of rich file systems

PARTITIONING

(page 100)

Partitioning, also called *domain support*, is a method of segmenting a server's resources to allow the processing of multiple workloads on a single system. That is, users can deploy a mixture of applications on separate partitions, each with its own copy of the operating system. First developed by IBM for the mainframe more than 25 years ago, partitioning is becoming ubiquitous on high-end RISC servers running UNIX and has made its debut on some of IBM's midrange servers as well as high-end Intel servers.

As many large corporations, data centers, and service providers attempt to reduce the costs associated with running large numbers of servers, partitioning is becoming more widespread. Partitioning offers the ability to segment system resources—CPU, memory, and I/O—with varying degrees of isolation between the partitions. In some cases, such as IBM's zSeries mainframes, customers can run a combination of operating systems, such as OS/390, z/OS, and Linux, in separate partitions. For instance, a customer can run a technical/engineering application on one partition running OS/390 or z/OS, and a series of e-commerce and data-warehousing applications on Linux partitions. There are distinctions in the manner in which partitions are implemented, the degree to which the application workloads are isolated, and how system resources are allocated to each partition.

Soft versus Hard Partitioning

Partitions may be implemented in software, hardware, or a combination of both. The benefit of software partitioning is that it enables finer partitioning of a single server than a hardware approach. The downside is that it cannot isolate application workloads in separate partitions as neatly, or in as fault-tolerant a manner, as hard partitions. *Physical,* or *hard, partitions* can isolate application workloads in a more fault-tolerant manner. With physical partitioning, the hardware enforces separation of partitions so they work as if they were separate servers. If a failure occurs in one partition, the remaining partitions can continue to function. Additionally, hard partitions give customers the ability to run multiple operating systems on a single server. The negative side is that fewer partitions can be created per system, thus reducing the number of workloads a single server can take on. IBM's logical partitions (LPARs), which are a combined software and hardware approach, have been a feature of IBM's S/390 architecture since 1988.

Dynamic versus Static Partitioning

There also is a distinction in the manner in which resources—CPUs, memory, and input/output—can be reallocated to various physical partitions in a single server. *Dynamic partitioning* refers to the ability to move resources from one partition to the other while the system is still running. Dynamic partitioning allows users to reassign resources between partitions on the fly without taking the system offline, or in an uninterrupted manner. *Static partitioning,* on the other hand, requires operators to halt the operation of the affected partition, reconfigure the machine as they want it, and reboot.

Vendors are enhancing dynamic partitioning by providing *automatic dynamic partitioning,* which allows the system to automatically reallocate resources between partitions based on application needs. In this case, the system monitors application needs within a partition and adjusts resources accordingly, without operator intervention.

Hard Partitioning versus Workload Management

Vendors are also extending workload management features to high-end systems to handle variable workloads. In this software-based model, the workload is controlled dynamically by the operating system. The benefits are that the model enables better resource allocation and prioritization of workloads. However, the downside is that it allows for only one operating system image, which is a single point of failure. In contrast, a hard-partitioned system allows for multiple operating system images on a single SMP server. IBM's zSeries partitioning has been improved with the introduction of Intelligent Resource Director, which dynamically and automatically can balance CPU power or I/O bandwidth to the priority LPARs on the server. In this model, the priority of an LPAR is determined by the Workload Manager, which has all business policies defined and continually monitors the server at ten-second intervals.

IBM'S ENTERPRISE OPERATING SYSTEM: z/OS (previously OS/390)

(page 100)

Introduced in 1974, Multiple Virtual Storage (MVS) was a batch-processing-oriented, mainframe operating system that managed large (by the standards of the time) amounts of memory and disk space. Since that time, IBM has enhanced and rewritten MVS to offer online operations, provided by subsystems such as the Customer Information Control System (CICS), Time Sharing Option (TSO), and other system software.

In 1981, the MVS family was enhanced by MVS/Extended Architecture (MVS/XA) to include 16-bit addressing, 2 GB of virtual memory, and other new features of IBM's System 370 XA architecture. In 1994, MVS/Enterprise System Architecture (MVS/ESA) enhanced the operating system for larger-scale mainframes by including 31-bit addressing and 16 terabytes of virtual memory with IBM's ESA/390 architecture. MVS also has been enhanced for lower-cost CMOS mainframe hardware and can be used to run UNIX applications, including Linux, e-commerce and e-business applications, and client/server deployments. During the mid-1990s, IBM changed the name MVS to **OS/390** to reflect the fact that it no longer was limited to traditional MVS functionality. IBM also included UNIX user and programming interfaces and could be used to run both traditional MVS applications and UNIX applications at the same time.

OS/390 runs on S/390 architecture systems from Amdahl, Hitachi, and IBM, and is designed to handle very large systems with many gigabytes of data. In September 2000, IBM introduced its z/Architecture, a new 64-bit mainframe architecture developed under the code name "Freeway" that replaces the 32-bit architecture first debuted in 1994. The first system implementing the new architecture is the **eServer zSeries 900**.

Would Your Organization Benefit from These Groupware Features?

Manager's Checklist W4.1

Common Groupware Features

(page 106)

- Electronic mail and messaging
- File and document storage/sharing
- Shared database access
- Calendaring and scheduling coordination
- Desktop videoconferencing
- Collaborative presentation capabilities
- Workflow/document routing
- Managed security/file access
- LAN or Internet-based access
- Meeting-support software

Manager's Checklist W4.2

Software Evaluation Criteria

(page 107)

Does the Software Meet Our Needs in Terms of the Following Criteria?

	Yes	No		Yes	No
• Ease of development	☐	☐	• Output options	☐	☐
• Learning (for developers and for users)	☐	☐	• Environments and hardware	☐	☐
• Reporting capability	☐	☐	• Security	☐	☐
• General functionality	☐	☐	• Documentation	☐	☐
• Data handling	☐	☐	• Maintenance	☐	☐
• Graphic presentation	☐	☐	• Vendor support	☐	☐
• Performance	☐	☐	• Cost	☐	☐

MIDDLEWARE AND COMPONENT SOFTWARE

(page 117)

The Internet is the ultimate distributed system. Internet developers cannot be sure what types of hardware and software their customers will be using when they seek access to company servers. Internet applications designed to let one company interact with other companies can be complex because of the variety of hardware and software with which they must be able to work. Furthermore, building a complete, end-to-end integrated system is a complex task, regardless of the tools available. Most applications do not support integration sufficiently and require workarounds. More sophisticated applications need **middleware** to link application modules developed in different computer languages and running on heterogeneous platforms. To accomplish this linkage, companies must use distributed middleware systems.

Components are modules of code that can serve as the building blocks for even larger programs. *Distributed components* offer two advantages over traditional one- or two-tier programs in helping to integrate disparate systems. First, they provide common interfaces for the applications that need to be linked to build Web applications. Second, because it shifts the labor from writing code to assembling prebuilt modules, the use of components can significantly reduce the cost and time involved in developing applications. Many companies already rely on the use of small components for the rapid development of user interface screens and for tasks such as linking applications to databases. Emphasis, however, remains on larger components and portfolio assembly. Larger components can offer functionality similar to that of off-the-shelf software, and portfolio assembly involves integrating multiple applications from different vendors into a suite of the best ones.

Scaling up from component-based software, middleware integrates code written for heterogeneous systems. The most popular of the standards for distributed component middleware are Microsoft's Distributed Component Object Model, Sun's Enterprise JavaBeans, and the Object Management Group's Common Object Request Broker Architecture. An increasingly popular middleware option is the Extensible Markup Language (XML) coupled with a transport protocol like Simple Object Access Protocol (SOAP).

Development can be simplified if a company chooses to use a distributed component system with a component-based application server. Component Object Model or

Enterprise JavaBeans application servers provide both basic support for a distributed component model and a wide range of integrated servers and utilities that the developers otherwise would have to create. For similar reasons, vendors are introducing Extensible Markup Language servers.

Middleware comprises two general subsets: the application server or component environment, and messaging. Systems that maintain links between client/server systems or between applications in conventional distributed systems are examples of the former. The latter applies to all types of messaging software, including XML. Middleware is any system or collection of utilities that links different elements together, whether on a single machine or over a network.

When applications were maintained on mainframes, the operating system and its utilities tracked modules and passed messages with ease. Managing the communications between the computers and the database server was also relatively straightforward in later client/server systems that simply linked computers to a single database running on a database server.

Specific interfaces between the client and server for each application, which maintained the software modules that the servers needed to reach, made sense for mainframes and early client/server systems. Today, application developers rely on middleware systems that are independent of specific applications or components that might want to talk to one another. This independence is necessary because the developer can no longer anticipate the location, language, or platform used at each of the nodes in the distributed system.

With the rise of the Internet, dealing with the problem of disparate systems has become more urgent. Companies are building applications to be linked by the Internet to clients and to other servers throughout the world. The company building the application cannot know which kind of platform the client is running or where it is located. This problem will only become more complex as a variety of mobile wireless devices begin to access company sites via the Internet.

In the world of networked computing, middleware must link a variety of platforms from an unknown number of locations. The middleware that keeps track of the locations of the software modules that need to link to each other and that manages the actual exchange of information is rapidly increasing in sophistication. In short, middleware is being used to solve the two distinct problems of heterogeneity and resource discovery.

Types of Middleware

A major challenge in building applications that can be partitioned over distributed systems is the creation of a uniform scheme for passing information among program modules and for accessing data from multiple sources when programs and data may reside on different platforms. Middleware provides developers with a uniform interface through which their programs can access other applications. Each application interacts only with the middleware, which performs the necessary translations to communicate with the appropriate databases, operating systems, and applications. The middleware thus presents the application developer with a single, consistent interface (the illusion of a single underlying server) that masks the complexity of the actual computing infrastructure.

Several types of middleware are often used together, because no single middleware approach is general enough for the full range of applications. Therefore, systems are not uniform and single-server, but are collections of subsystems that use a variety of middleware to interoperate. This diversity removes the pressure for an entire enterprise to use a single solution and encourages business data standards for integrating middleware systems. In other cases, heterogeneity is unavoidable, so bridging integration solutions are necessary.

In addition to providing connectivity, middleware often provides special services for an application when it is desirable to isolate those services rather than build them directly into the application. Isolation is beneficial if a service is used by multiple applications (saving money by providing a service all applications can share) or if the software to implement those services is purchased from a middleware vendor. Some of these services function independently, and their association with true middleware is merely historical. (Some common examples are Common Object Request Broker Architecture's collections, properties, query, relationship, and time services. Although all these are application independent, none require intimate integration with the middleware infrastructure.) Other services, however, are intertwined with the middleware's connectivity services (such as security, high availability, and load balancing).

Of greatest interest to developers of distributed systems is middleware for remote procedure calls, messaging, transaction processing, and object and component management, as well as the dominant types of distributed middleware, such as enterprise application integration (EAI), distributed components, and XML.

Remote Procedure Call (RPC) Middleware. The RPC middleware model is based on a synchronous approach to communication. When a procedure has initiated a call to another procedure, it waits for the response. If a call from procedure A to procedure B initiates subsequent calls from procedure B to procedure C, etc., each procedure involved in the sequence waits as the call is transmitted out and back. (In *synchronous* communications, data are transmitted at regular intervals and in one direction. In *asynchronous* communications, data can be transmitted at any time and in any direction.)

The primary drawback to RPC middleware is that its programming model is at a low level of abstraction. Consequently, applications using RPCs are tedious to write. RPCs allow point-to-point communication between machines, but they do little to mask the details of implementation from developers. Each RPC in an exchange, as well as control flow embedding the RPCs, must be explicitly coded, which slows development and results in code that is more difficult to change than it would have been if components and objects had been used. Another disadvantage of this type of middleware is that it works for applications that use few RPCs, but in high-volume systems or in cases where the function call will take a long time to complete, an asynchronous approach works better.

Message-Oriented (MOM) Middleware. MOM is a popular asynchronous model for exchanging messages between applications or modules. A MOM environment incorporates a queue. The initiating module sends a message to the queue, then goes on with other processing. The queue tries to send the message to the target module, continuing to do so until the target module is available and accepts the message. Once the target module has created a response, that response is also sent to the queue, which then forwards it to the originator, again continuing to forward the message until it is accepted.

Queues are especially effective if the originator or the target modules receive an overwhelming number of messages. In such cases, the queue mechanism holds the messages and forwards them only when the target module can process them. In that way, MOM prevents the source from blocking and prevents the target from being overwhelmed. MOM also can add at least a second target process to read from the same queue, increasing the rate at which the messages are processed.

A common approach is to use commercial MOM software, which often provides functionality beyond simple message passing. For example, most MOM systems incorporate some way of keeping track of where modules are so that the target can be more or less transparent to the sender. Similarly, if a MOM system is to pass messages between modules written in different languages, it requires a neutral Interface Description Language (IDL) mechanism to provide a common language for the two modules. MOM systems have been developed to handle asynchronous calls between procedures, between components, or between modules or applications.

Transaction Processing Middleware. In database systems, a *transaction* represents a group of commands that must succeed or fail as a single unit. An example is an operation that transfers funds from one bank account to another. Both halves of the transaction must occur before the transaction can succeed. If funds are debited from one account, they must be credited to the other account. If one of these operations fails, all parts of the transaction must be restored to their original states. That is, a transaction processing system keeps a copy of the original state of the elements in the transaction and releases that copy *only when* all changes necessary to the transaction have been accomplished. Transaction processing systems are essential to financial systems and to any other applications that require transactional integrity.

Transaction processing monitors, also called transaction managers, are based on mainframe processing. Examples include the Customer Information Control System (CICS) and Information Management System, which have matured to provide excellent performance and reliability.

Modern distributed computing applications frequently use middleware to access relational database servers using Structured Query Language (SQL) (discussed in Chapter 5). Ideally, business software applications would be written as a series of transactions. All participating application components would adhere to the transaction-management paradigm. In reality, there are two basic approaches to transaction management:

- ***Without a separate transaction-processing monitor:*** This approach uses the transaction management capabilities of the relational database management system's server product.
- ***With a separate transaction-processing monitor:*** This approach is required when several heterogeneous databases are being updated in a single transaction. It also can improve performance in high-volume transactional applications.

Component Middleware

The three best-known component models are Microsoft's Component Object Model (COM), Sun's Enterprise JavaBeans (EJB) model, and the Object Management Group's (OMG's) CORBA Component Model (CCM). The component level comprises interfaces and an environment with services that keep track of components and pass messages between them. Each of these three component models (CORBA, COM, and JavaBeans) has been extended by the creation of a new, more advanced component model (the new CCM, COM+, and EJB, respectively). One of the advantages of component software is that its development is independent of how the component is written. The only attributes of a component that a developer needs to know are its interfaces.

OMG's CORBA. The CORBA system is transparent to the developer in that he or she does not need to know where the target object is located. It can reside on the same machine or on another platform with the same or different operating systems.

To use CORBA, developers wrap components written in a specific object-oriented language by creating IDL interfaces. The code that implements a CORBA environment can be object-oriented code, but it also can be written in a non-object-oriented language like C. At the CORBA or component level, one does not care what is inside the components. Thus, CORBA can also be used to wrap COBOL modules and integrate them.

Microsoft's COM. Microsoft's COM is another approach to components that defines an environment that allows a developer to create interfaces for code modules, then allows any code modules to send messages and obtain results from other COM modules. COM modules are not objects. They lack some of the characteristics that have traditionally been used to define objects, specifically inheritance. COM modules are components. They have interfaces defined in Microsoft IDL and an environment, which is built into the various Windows operating systems, that passes messages between COM components.

To pass messages to COM components on networked Windows platforms, a Windows developer uses DCOM, which is an RPC mechanism that moves messages between distributed COM components. COM components can be developed in non-object-oriented languages like C. Like CORBA, the key is that at the component level, the system is simply concerned with interfaces and passing messages and does not concern itself with what kind of code is inside the COM components.

Sun's Enterprise JavaBeans (EJB) Model. This component system was developed by Sun to provide its Java object-oriented language with modules that would be easy to reuse. Sun defined a specific type of Java object that it referred to as a *JavaBean*. JavaBeans have interfaces written in Java. Sun also specified utilities (other objects) that would manage message passing between JavaBeans. Java's Remote Method Invocation (RMI) enables a Java program running on one computer to access the objects and methods of another Java program running on a different computer (i.e., it is an RPC-like mechanism for accessing JavaBeans on distributed platforms).

Enterprise Application Integration (EAI) Middleware. EAI-packaged middleware is designed to allow developers to choose different levels of integration (e.g., data-level, application interface-level, or business process-level) between existing and new systems. As a result, many older systems that still perform valuable tasks are able to continue. Indeed, often the return on an EAI investment is measured by how little of the installed base needs to be changed to leverage a new system.

Another driver behind EAI deployment involves sharing information between disparate applications that never were designed to work together. *Business process automation* is the automation of tasks that an organization already might be doing manually by integrating two or more applications. For example, a business can print a sales report from an ERP system to cross-reference customer information for credit checks in another system. EAI would transform this task from a manual process to an automated one.

The single-application vendor solution promises the highest degree of integration among applications. However, it is unlikely that this approach can satisfy requirements for complete integration across a portfolio of applications because, according to various analysts, packaged applications address only about one-third of an enterprise's requirements. The remaining two-thirds are satisfied by custom applications. So, once integration spreads beyond the scope of a single application suite, planners, designers, and architects must rely on EAI middleware. EAI middleware addresses the uppermost three layers of services seen next.

Managers Checkist W4.3

Seven Levels of Application Integration

Business Integration	1. *Business process development*—business process design/modeling, real-time decision support, state management
Application Integration	2. *Business event processing*—automatic event notification, flow control, content routing, transactional integrity 3. *Application content transformation*—format translation, data semantics, validation, prebuilt templates
Application Connectivity	4. *Application bridges and gateways*—for legacy, Web, database, and packaged applications 5. *Application interaction styles*—publish/subscribe, publish/reply, file transfer, request/reply, conversational 6. *Message-handling services*—queuing, security, message management, administration 7. *Basic communications*—point-to-point, reliable broadcast, IP multicast, IIOP/ORB, database, Web, 3270 SNA

Distributed Component Middleware

Although components are often used as a necessary part of middleware, they are not required. Middleware systems have been built in a variety of ways, using different technologies. As the complexity and heterogeneity of the typical enterprise's portfolio of applications increase, however, distributed component middleware systems may play a role in reducing that complexity.

A potential approach could be to convert all of a system's modules into independent components with standard interfaces. Then, instead of wiring the entire system together, a middleware system could be used to pass messages between the components. Because the components would be uncoupled, they could be added or eliminated without changing the entire system. However, the standard interfaces are not currently provided by most widely used component models. Most commercial middleware packages use the MOM model, and each package has a proprietary method for connecting an application to the middleware. Open standards, such as the Java Cryptography Architecture, are just beginning to come into use, and no major application vendors are supporting them. Current middleware connectors are makeshift, customized components that use the application's native API to get the interface to work. Some vendors are starting to standardize the connections between applications and middleware, as Siebel is doing with its XML-based EAI bridge.

Developers can reduce the complexity of developing distributed component systems in other ways. Many middleware systems implement protocols, such as APIs, and supply a few utilities, which are usually components themselves. If the information passing between two components running on different platforms is uncomplicated, then a minimal middleware package can be used.

However, many distributed applications and almost all e-commerce applications require more than simply passing data between two modules. For example, a distrib-

IT's About Business

MKT

(page 101)

Box W4.2: Surveillance software: How much is too much?

AT WORK

A survey found that 54 percent of companies said they monitored their employees' Internet connections, while 38 percent said they reviewed worker e-mail messages. Silent Watch surveillance software from Adavi (*adavi.com*) allows an employer to monitor dozens of computers from a single screen in real time, while recording every keystroke an employee makes, even if the data are deleted.

WHILE SHOPPING

Many large retailers have implemented surveillance systems that record and interpret customer movements, using IBM BlueEyes software that tracks customer movements and pupil, eyebrow, and mouth movement. The software allows computers to anticipate users' wants by gathering video data on eye movement and facial expression.

The American Civil Liberties Union notes that the software will be able to identify who shoppers are. Once identity is established, it can be cross-referenced to capture that shopper's income and buying preferences. Interestingly, purchasers of BlueEyes unanimously have requested that IBM not reveal their names to the press, or the locations where the software has been implemented.

AT SCHOOL

eSniff Solution from Vericept (*vericept.com*) allows schools to monitor students' use of school computers. ESniff detects packets of information that indicate that proscribed activities are occurring—that a student is visiting Web sites that offer, for example, sexual images, or copyright-violating music downloads. The software can also "sniff" e-mail messages for signs of out-of-bounds activities, like drug dealing. Suspect pages and messages are captured on a computer hard disk for review by school administrators. One high school reports that access of unauthorized Internet sites dropped by 98 percent after the school's administrators announced the implementation of the software.

AT HOME

A child was playing spelling games on Reader Rabbit, Mattel Interactive's popular educational software program. After she went to bed, her father logged on to the Internet to do some work. Minutes into his session, an Internet security program, or firewall, alerted him that Reader Rabbit was attempting to secretly send data from his computer to Mattel. Inside the child's software he uncovered "spyware," a type of program that embeds itself in a PC's hard drive. It can then relay information to and from over the Internet connection. Because the information being sent was encrypted, the father could not determine what exactly was being sent to Mattel. Mattel said that the spyware was originally designed to offer consumers additional product content and to communicate bug fixes. (Computer users concerned about potential spyware transmission can download the ZoneAlarm 2.6 firewall, available free at *zonealarm.com*.)

Sources: "More Employers Taking Advantage of New Cyber-Surveillance Software," *cnn.com*, July 11, 2000; "Schools Get Tool to Track Students' Internet Use," *nytimes.com*, May 21, 2001; "Privacy Worries Arise Over Spyware in Kids' Software," *U.S. News and World Report*, July 3, 2000, p. 55; "Behind BlueEyes," *MIT Technology Review*, May 2001, p. 32.

Questions

1. How would you feel about having yourself identified by surveillance software in any of the preceding situations? Would some be acceptable to you but others not acceptable? Why?
2. Do you feel that surveillance software is too intrusive? If not, do you feel that surveillance software has the potential for becoming too intrusive?
3. What are the advantages and disadvantages of surveillance software to companies?

uted application may need a transaction-processing model that ensures that updates to data are not finally committed until all aspects of the transaction are ready to be applied. Most Internet systems need the capability to scale from one to multiple servers. This task requires an application designed in such a way that multiple instances of it can run without producing invalid results. It also involves load balancing to ensure that the server is not overloaded during peak periods of activity.

Distributed Component Architectures. Software architectures describe the basic elements and relationships used in software development. Companies that embrace distributed component middleware are adopting a component architecture. In such cases, the key elements in the business environment are described in terms of components, and a middleware infrastructure is included to facilitate communication between the components. The three most popular component architectures are OMG's CORBA architecture, Microsoft's Distributed Internet Architecture (DNA), and Sun's J2EE architecture.

In our example (see Figure 4.10b), ACCOUNT would be a superclass of the CHECKING-ACCOUNT subclass and the SAVINGS-ACCOUNT subclass. Inheritance here means that the CHECKING-ACCOUNT subclass and SAVINGS-ACCOUNT subclass inherit data values such as customer name and account number from the ACCOUNT class (see Figure 4.10b). Also, the CHECKING-ACCOUNT subclass and the SAVINGS-ACCOUNT subclass inherit methods such as add deposit and calculate balance from the ACCOUNT class. It is important to note that the CHECKING-ACCOUNT subclass has an additional data value, charge per check, and an additional method, subtract check. Further, the SAVINGS-ACCOUNT subclass has an additional data value, interest rate, and an additional method, subtract withdrawal. These additional data values and methods serve to differentiate subclasses from their superclasses. A useful aspect of inheritance is that if necessary, one can change the data values or methods in a superclass, and the changes will be reflected in all subsequent subclasses.

(page 117)

Table W4.2 Common Modules of Enterprise Application Software

• Payroll	• Shipping/logistics
• General ledger	• Employee records
• Sales-order processing	• Tax accounting and planning
• Cash-flow analysis	• Manufacturing control
• Fixed-asset accounting	• Check processing
• Accounts payable	• Customer service
• Invoicing	• Sales force support
• Accounts receivable	