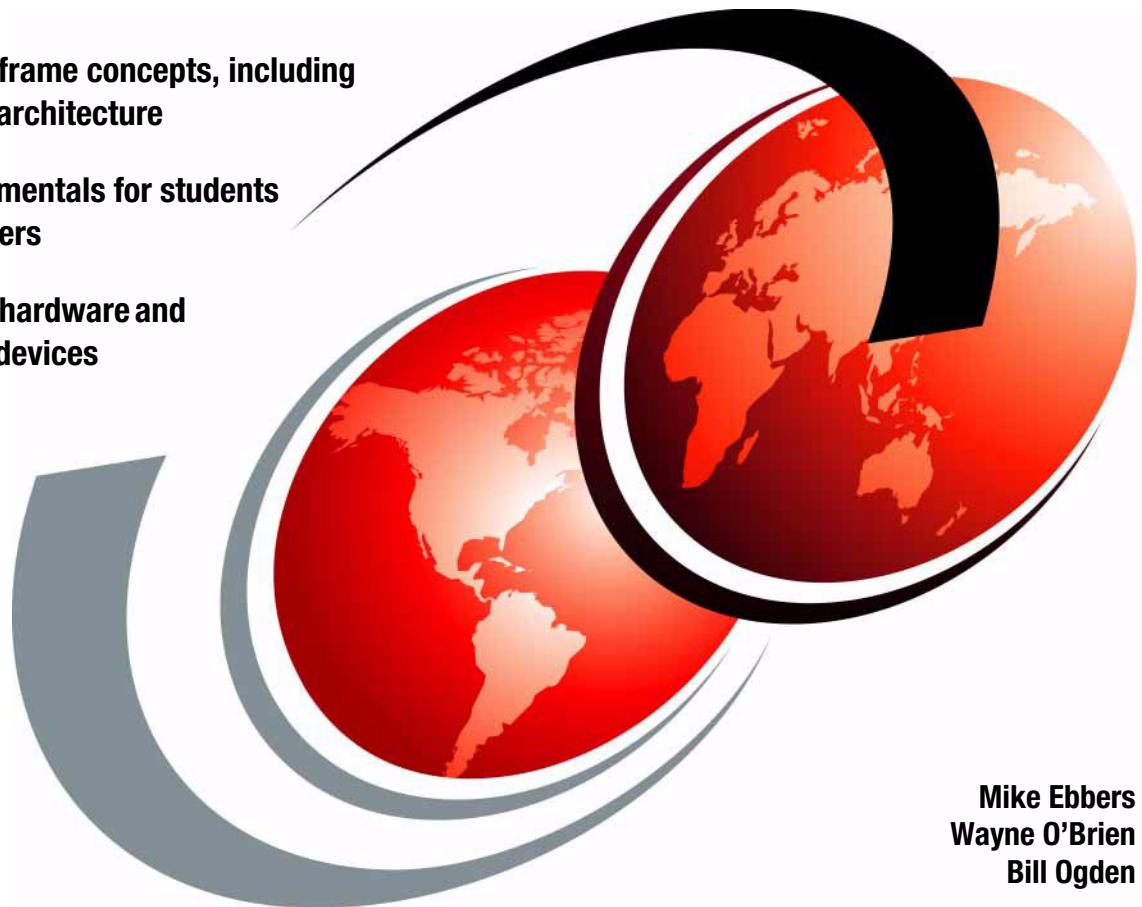


Introduction to the New Mainframe: z/OS Basics

Basic mainframe concepts, including
usage and architecture

z/OS fundamentals for students
and beginners

Mainframe hardware and
peripheral devices



Mike Ebbers
Wayne O'Brien
Bill Ogden

Part 1

Introduction to z/OS and the mainframe environment

Welcome to mainframe computing! We begin this text with an overview of the mainframe computer and its place in today's information technology (IT) organization. We explore the reasons why public and private enterprises throughout the world rely on the mainframe as the foundation of large-scale computing. We discuss the types of workloads that are commonly associated with the mainframe, such as batch jobs and online or interactive transactions, and the unique manner in which this work is processed by a widely used mainframe operating system—z/OS.

Throughout this text, we pay special attention to the people who use mainframes and to the role of the New Mainframe in today's business world.

Introduction to the new mainframe

Objective: As a technical professional in the world of mainframe computing, you will need to understand how mainframe computers support your company's IT infrastructure and business goals. You will also need to know the job titles of the various members of your company's mainframe support team.

After completing this chapter, you will be able to:

- ▶ List ways in which the mainframe of today challenges the traditional thinking about centralized computing versus distributed computing.
- ▶ Explain how businesses make use of mainframe processing power, the typical uses of mainframes, and how mainframe computing differs from other types of computing.
- ▶ Outline the major types of workloads for which mainframes are best suited.
- ▶ Name five jobs or responsibilities that are related to mainframe computing.
- ▶ Identify four mainframe operating systems.

1.1 The new mainframe

e-business

The transaction of business over an electronic medium such as the Internet.

Today, mainframe computers play a central role in the daily operations of most of the world's largest corporations, including many Fortune 1000 companies. While other forms of computing are used extensively in various business capacities, the mainframe occupies a coveted place in today's e-business environment. In banking, finance, health care, insurance, public utilities, government, and a multitude of other public and private enterprises, the mainframe computer continues to form the foundation of modern business.

The long-term success of mainframe computers is without precedent in the information technology (IT) field. Periodic upheavals shake world economies and continuous—often wrenching—change in the Information Age has claimed many once-compelling innovations as victims in the relentless march of progress. As emerging technologies leap into the public eye, many are just as suddenly rendered obsolete by some even newer advancement. Yet today, as in every decade since the 1960s, mainframe computers and the mainframe *style* of computing dominate the landscape of large-scale business computing.

Why has this one form of computing taken hold so strongly among so many of the world's corporations? In this chapter, we look at the reasons why mainframe computers continue to be the popular choice for large-scale business computing.

1.2 The S/360: A turning point in mainframe history

When did mainframe computers come into being? The origin of mainframe computers dates back to the 1950s, if not earlier. In those days, mainframe computers were not just the largest computers; they were the *only* computers and few businesses could afford them.

System/360

The first general purpose computer, introduced in 1964.

Mainframe development occurred in a series of *generations* starting in the 1950s. First generation systems, such as the IBM 705 in 1954 and the IBM 1401 in 1959, were a far cry from the enormously powerful machines that were to follow, but they clearly had characteristics of mainframe computers. These computers were sold as business machines and served then—as now—as the central data repository in a corporation's data processing center.¹

In the 1960s, the course of computing history changed dramatically when mainframe manufacturers began to standardize the hardware and software they offered to customers. The introduction of the IBM System/360™ (or S/360™) in 1964 signaled the start of the third generation: the first general purpose

¹ According to the IBM product brochure, typical customer uses for a 1401 were "payroll, railroad freight car accounting, public utility customer accounting, inventory control, and accounts receivable."

computers. Earlier systems such as the 1401 were dedicated as either commercial or scientific computers. The revolutionary S/360 could perform both types of computing, as long as the customer, a software company, or a consultant provided the programs to do so. In fact, the name S/360 refers to the architecture's wide scope: 360 degrees to cover the entire circle of possible uses.

The S/360 was also the first of these computers to use *microcode* to implement many of its machine instructions, as opposed to having all of its machine instructions hard-wired into its circuitry. Microcode (or *firmware*, as it is sometimes called) consists of stored microinstructions, not available to users, that provide a functional layer between hardware and software. The advantage of microcoding is flexibility, where any correction or new function can be implemented by just changing the existing microcode, rather than replacing the computer.

With standardized mainframe computers to run their workloads, customers could, in turn, write business applications that didn't need specialized hardware or software. Moreover, customers were free to upgrade to newer and more powerful processors without concern for compatibility problems with their existing applications. The first wave of customer business applications were mostly written in Assembler, COBOL, FORTRAN, or PL/1, and a substantial number of these older programs are still in use today.

In the decades since the 1960s, mainframe computers have steadily grown to achieve enormous processing capabilities. The New Mainframe has an unrivaled ability to serve end users by the tens of thousands, manage petabytes of data, and reconfigure hardware and software resources to accommodate changes in workload—all from a single point of control.

1.3 An evolving architecture

An *architecture* is a set of defined terms and rules that are used as instructions to build products. In computer science, an architecture describes the organizational structure of a system. An architecture can be recursively decomposed into parts that interact through interfaces, relationships that connect parts, and constraints for assembling parts. Parts that interact through interfaces include classes, components, and subsystems.

Architecture
describes the
organizational
structure of a
system.

Starting with the first large machines, which arrived on the scene in the 1960s and became known as “Big Iron” (in contrast to smaller departmental systems), each new generation of mainframe computers has included improvements in one or more of the following areas of the architecture:²

- ▶ More and faster processors

- ▶ More physical memory and greater memory addressing capability
- ▶ Dynamic capabilities for upgrading both hardware and software
- ▶ Increased automation of hardware error checking and recovery
- ▶ Enhanced devices for input/output (I/O) and more and faster paths (*channels*) between I/O devices and processors
- ▶ More sophisticated I/O attachments, such as LAN adapters with extensive inboard processing
- ▶ A greater ability to divide the resources of one machine into multiple, logically independent and isolated systems, each running its own operating system
- ▶ Advanced clustering technologies, such as Parallel Sysplex, and the ability to share data among multiple systems.

Despite the continual change, mainframe computers remain the most stable, secure, and compatible of all computing platforms. The latest models can handle the most advanced and demanding customer workloads, yet continue to run applications that were written in the 1970s or earlier.

How can a technology change so much, yet remain so stable? It can by evolving to meet new challenges. In the early 1990s, the client/server model of computing, with its distributed nodes of less powerful computers, emerged to challenge the dominance of mainframe computers. Industry pundits predicted a swift end for the mainframe computer and called it a “dinosaur.” In response, mainframe designers did what they have always done when confronted with changing times and a growing list of user requirements: they designed new mainframe computers to meet the demand. With a tip of the hat to the dinosaur naysayers, IBM, as the leading manufacturer of mainframe computers, code-named its then-current machine T-Rex.

With the expanded functions and added tiers of data processing capabilities such as Web-serving, autonomies, disaster recovery, and grid computing, the mainframe computer is poised to ride the next wave of growth in the IT industry. Mainframe manufacturers such as IBM are once again reporting annual sales growth in the double digits.

And the evolution continues. While the mainframe computer has retained its traditional, central role in the IT organization, that role is now defined to include being the primary hub in the largest distributed networks. In fact, the Internet

“I predict that the last mainframe will be unplugged on March 15, 1996.”
—Stewart Alsop, Infoworld, March 1991

² Since the introduction of the S/360 in 1964, IBM has significantly extended the platform roughly every ten years: System/370™ in 1970, System/370 Extended Architecture (370-XA) in 1983, Enterprise Systems Architecture/390® (ESA/390) in 1990, and z/Architecture™ in 2000. For more information about earlier mainframe hardware systems, see Appendix A, “A brief look at IBM mainframe history” on page 637.

itself is based largely on numerous, interconnected mainframe computers serving as major hubs and routers.

As the image of the mainframe computer continues to evolve, you might ask: is the mainframe computer a self-contained computing environment, or is it one part of the puzzle in distributed computing? The answer is that The New Mainframe is both: a self-contained processing center, powerful enough to process the largest and most diverse workloads in one secure “footprint,” and one that is just as effective when implemented as the primary server in a corporation’s distributed server farm. In effect, the mainframe computer is the definitive server in the client/server model of computing.

1.4 Mainframes in our midst

Despite the predominance of mainframes in the business world, these machines are largely invisible to the general public, the academic community, and indeed many experienced IT professionals. Instead, other forms of computing attract more attention, at least in terms of visibility and public awareness. That this is so is perhaps not surprising. After all, who among us needs direct access to a mainframe? And, if we did, where would we find one to access? The truth, however, is that we are *all* mainframe users, whether we realize it or not (more on this later).

Most of us with some personal computer (PC) literacy and sufficient funds can purchase a notebook computer and quickly put it to good use—running software, browsing Web sites, and perhaps even writing papers for college professors to grade. With somewhat greater effort and technical prowess, we can delve more deeply into the various facilities of a typical Intel®-based workstation and learn its capabilities through direct, hands-on experience—with or without help from any of a multitude of readily available information sources in print or on the Web.

Mainframes, however, tend to be hidden from the public eye. They do their jobs dependably—indeed, with almost total reliability—and are highly resistant to most forms of insidious abuse that afflict PCs, such as e-mail-borne viruses and Trojan Horses. By performing stably, quietly, and with negligible downtime, mainframes are the example by which all other computers are judged. But at the same time, this lack of attention tends to allow them to fade into the background.

Furthermore, in a typical customer installation, the mainframe shares space with many other hardware devices: external storage devices, hardware network routers, channel controllers, and automated tape library “robots,” to name a few. The New Mainframe is physically no larger than many of these devices and generally does not stand out from the crowd of peripheral devices.

So, how can we explore the mainframe's capabilities in the real world? How can we learn to interact with the mainframe, learn its capabilities, and understand its importance to the business world? Major corporations are eager to hire new mainframe professionals, but there's a catch: Some previous experience would help.

Would we even know a mainframe if we saw one, given that these machines have evolved to flourish in the twenty-first century IT organization? What we need is an experienced guide to lead us on a *dinosaur safari*, which is where this textbook comes in!

1.5 What is a mainframe?

First, let's tackle the terminology. Today, computer manufacturers don't always use the term *mainframe* to refer to mainframe computers. Instead, most have taken to calling any commercial-use computer—large or small—a *server*, with the mainframe simply being the largest type of server in use today. IBM, for example, refers to its latest mainframe as the IBM System z9™ server. We use the term mainframe in this text to mean computers that can support thousands of applications and input/output devices to simultaneously serve thousands of users.

Servers are proliferating. A business might have a large server collection that includes transaction servers, database servers, e-mail servers and Web servers. Very large collections of servers are sometimes called *server farms* (in fact, some data centers cover areas measured in *acres*). The hardware required to perform a server function can range from little more than a cluster of rack-mounted personal computers to the most powerful mainframes manufactured today.

Server farm

A very large collection of servers.

A mainframe is the central data repository, or *hub*, in a corporation's data processing center, linked to users through less powerful devices such as workstations or terminals. The presence of a mainframe often implies a centralized form of computing, as opposed to a distributed form of computing. Centralizing the data in a single mainframe repository saves customers from having to manage updates to more than one copy of their business data, which increases the likelihood that the data is current.

The distinction between centralized and distributed computing, however, is rapidly blurring as smaller machines continue to gain in processing power and mainframes become ever more flexible and multi-purpose. Market pressures require that today's businesses continually reevaluate their IT strategies to find better ways of supporting a changing marketplace. As a result, mainframes are now frequently used in combination with networks of smaller servers in a multitude of configurations. The ability to dynamically reconfigure a mainframe's

hardware and software resources (such as processors, memory, and device connections), while applications continue running, further underscores the flexible, evolving nature of the modern mainframe.

While mainframe hardware has become harder to pigeon-hole, so, too, have the operating systems that run on mainframes. Years ago, in fact, the terms defined each other: a mainframe was any hardware system that ran a major IBM operating system.³ This meaning has been blurred in recent years because these operating systems can be run on very small systems.

Platform

A computer architecture (hardware and software).

Computer manufacturers and IT professionals often use the term *platform* to refer to the hardware and software that are associated with a particular computer architecture. For example, a mainframe computer and its operating system (and their predecessors⁴) are considered a platform; UNIX on a Reduced Instruction Set Computer (RISC) system is considered a platform somewhat independently of exactly which RISC machine is involved; personal computers can be seen as several different platforms, depending on which operating system is being used.

So, let's return to our question now: "What is a mainframe?" Today, the term mainframe can best be used to describe a *style* of operation, applications, and operating system facilities. To start with a working definition, "a mainframe is what businesses use to host the commercial databases, transaction servers, and applications that require a greater degree of security and availability than is commonly found on smaller-scale machines."

Mainframe

A large computer system that is used to host the databases, transaction servers, and applications that require a great degree of security and availability.



Early mainframe systems were housed in enormous, room-sized metal boxes or frames, which is probably how the term mainframe originated. The early mainframe required large amounts of electrical power and air-conditioning, and the room was filled mainly with I/O devices. Also, a typical customer site had several mainframes installed, with most of the I/O devices connected to all of the mainframes. During their largest period, in terms of physical size, a typical mainframe occupied 2,000 to 10,000 square feet (600 to 3000 square meters). Some installations were even larger than this.

³ The name was also traditionally applied to large computer systems that were produced by other vendors.

⁴ IBM System/390® (S/390®) refers to a specific series of machines, which have been superseded by the IBM zSeries machines. Nevertheless, many S/390 systems are still in use. Therefore, keep in mind that although we discuss the zSeries systems in this course, almost everything discussed also applies to S/390 machines. One major exception is 64-bit addressing, which is used only with zSeries.



Starting around 1990, mainframe processors and most of their I/O devices became physically smaller, while their functionality and capacity continued to grow. Mainframe systems today are much smaller than earlier systems—about the size of a large refrigerator.

In some cases, it is now possible to run a mainframe operating system on a PC that emulates a mainframe. Such emulators are useful for developing and testing business applications before moving them to a mainframe production system.

Clearly, the term mainframe has expanded beyond merely describing the physical characteristics of a system. Instead, the word typically applies to some combination of the following attributes:

- ▶ Compatibility with mainframe operating systems, applications, and data.
- ▶ Centralized control of resources.
- ▶ Hardware and operating systems that can share access to disk drives with other systems, with automatic locking and protection against destructive simultaneous use of disk data.
- ▶ A *style* of operation, often involving dedicated operations staff who use detailed *operations procedure books* and highly organized procedures for backups, recovery, training, and disaster recovery at an alternative location.
- ▶ Hardware and operating systems that routinely work with hundreds or thousands of simultaneous I/O operations.
- ▶ Clustering technologies that allow the customer to operate multiple copies of the operating system as a single system. This configuration, known as Parallel Sysplex, is analogous in concept to a UNIX cluster, but allows systems to be added or removed as needed, while applications continue to run. This flexibility allows mainframe customers to introduce new applications, or discontinue the use of existing applications, in response to changes in business activity.
- ▶ Additional data and resource sharing capabilities. In a Parallel Sysplex, for example, it is possible for users across multiple systems to access the same databases concurrently, with database access controlled at the *record level*.

As the performance and cost of such hardware resources as central processing unit (CPU) power and external storage media improve, and the number and types of devices that can be attached to the CPU) increase, the operating system software can more fully take advantage of the improved hardware. Also, continuing improvements in software functionality help drive the development of each new generation of hardware systems.

1.6 Who uses mainframe computers?

So, who uses mainframes? Just about *everyone* has used a mainframe computer at one point or another. If you ever used an automated teller machine (ATM) to interact with your bank account, you used a mainframe.

Today, mainframe computers play a central role in the daily operations of most of the world's largest corporations. While other forms of computing are used extensively in business in various capacities, the mainframe occupies a coveted place in today's e-business environment. In banking, finance, health care, insurance, utilities, government, and a multitude of other public and private enterprises, the mainframe computer continues to be the foundation of modern business.

Until the mid-1990s, mainframes provided the *only* acceptable means of handling the data processing requirements of a large business. These requirements were then (and are often now) based on large and complex batch jobs, such as payroll and general ledger processing.

The mainframe owes much of its popularity and longevity to its inherent reliability and stability, a result of careful and steady technological advances that have been made since the introduction of the System/360 in 1964. No other computer architecture can claim as much continuous, evolutionary improvement, while maintaining compatibility with previous releases.

Because of these design strengths, the mainframe is often used by IT organizations to host the most important, *mission-critical* applications. These applications typically include customer order processing, financial transactions, production and inventory control, payroll, as well as many other types of work.

One common impression of a mainframe's user interface is the 80x24-character "green screen" terminal, named for the old cathode ray tube (CRT) monitors from years ago that glowed green. In reality, mainframe interfaces today look much the same as those for personal computers or UNIX systems. When a business application is accessed through a Web browser, there is often a mainframe computer performing crucial functions "behind the scene."

Many of today's busiest Web sites store their production databases on a mainframe host. New mainframe hardware and software products are ideal for Web transactions because they are designed to allow huge numbers of users and applications to rapidly and simultaneously access the same data without interfering with each other. This security, scalability, and reliability is critical to the efficient and secure operation of contemporary information processing.

Corporations use mainframes for applications that depend on scalability and reliability. For example, a banking institution could use a mainframe to host the

database of its customer accounts, for which transactions can be submitted from any of thousands of ATM locations worldwide.

Businesses today rely on the mainframe to:

- ▶ Perform large-scale transaction processing (thousands of transactions per second)⁵
- ▶ Support thousands of users and application programs concurrently accessing numerous resources
- ▶ Manage terabytes of information in databases
- ▶ Handle large-bandwidth communication

The roads of the information superhighway often lead to a mainframe.

1.7 Factors contributing to mainframe use

The reasons for mainframe use are many, but most generally fall into one or more of the following categories:

- ▶ Reliability, availability, and serviceability
- ▶ Security
- ▶ Scalability
- ▶ Continuing compatibility
- ▶ Evolving architecture

Let's look at each of these categories in more detail.

1.7.1 Reliability, availability, and serviceability

The *reliability*, *availability*, and *serviceability* (or "RAS") of a computer system have always been important factors in data processing. When we say that a particular computer system "exhibits RAS characteristics," we mean that its design places a high priority on the system remaining in service at all times. Ideally, RAS is a central design feature of all aspects of a computer system, including the applications.

RAS has become accepted as a collective term for many characteristics of hardware and software that are prized by mainframe users. The terms are defined as follows:

Reliability The system's hardware components have extensive self-checking and self-recovery capabilities. The system's

⁵ IBM's latest series of mainframe computers, the IBM System z9 109 (also known as the z9-109) can process a staggering *one billion* transactions per day.

Availability

The ability to recover from the failure of a component without impacting the rest of the running system.

Availability

software reliability is a result of extensive testing and the ability to make quick updates for detected problems.

The system can recover from a failed component without impacting the rest of the running system. This applies to hardware recovery (the automatic replacing of failed elements with spares) and software recovery (the layers of error recovery that are provided by the operating system).

Serviceability

The system can determine why a failure occurred. This allows for the replacement of hardware and software elements while impacting as little of the operational system as possible. This term also implies well-defined units of replacement, either hardware or software.

A computer system is available when its applications are available. An available system is one that is reliable; that is, it rarely requires downtime for upgrades or repairs. And, if the system is brought down by an error condition, it must be serviceable; that is, easy to fix within a relatively short period of time.

Mean time between failure (MTBF) refers to the availability of a computer system. The New Mainframe and its associated software have evolved to the point that customers often experience months or even *years* of system availability between system downtimes. Moreover, when the system is unavailable because of an unplanned failure or a scheduled upgrade, this period is typically very short. The remarkable availability of the system in processing the organization's mission-critical applications is vital in today's 24-hour, global economy. Along with the hardware, mainframe operating systems exhibit RAS through such features as storage protection and a controlled maintenance process.

Beyond RAS, a state-of-the-art mainframe system might be said to provide *high availability* and *fault tolerance*. Redundant hardware components in critical paths, enhanced storage protection, a controlled maintenance process, and system software designed for unlimited availability all help to ensure a consistent, highly available environment for business applications in the event that a system component fails. Such an approach allows the system designer to minimize the risk of having a *single point of failure* undermine the overall RAS of a computer system.

1.7.2 Security

One of a firm's most valuable resources is its data: customer lists, accounting data, employee information, and so on. This critical data needs to be securely managed and controlled, and, simultaneously, made available to those users authorized to see it. The mainframe computer has extensive capabilities to simultaneously share, but still protect, the firm's data among multiple users.

In an IT environment, data security is defined as protection against unauthorized access, transfer, modification, or destruction, whether accidental or intentional. To protect data and to maintain the resources necessary to meet the security objectives, customers typically add a sophisticated security manager product to their mainframe operating system. The customer's security administrator often bears the overall responsibility for using the available technology to transform the company's security policy into a usable plan.

A secure computer system prevents users from accessing or changing any objects on the system, including user data, except through system-provided interfaces that enforce authority rules. The New Mainframe can provide a very secure system for processing large numbers of heterogeneous applications that access critical data. In this text, we discuss one example of a mainframe security system in Chapter 18, "Security on z/OS" on page 587.

1.7.3 Scalability

It has been said that the only constant is *change*. Nowhere is that statement truer than in the IT industry. In business, positive results can often trigger a growth in IT infrastructure to cope with increased demand. The degree to which the IT organization can add capacity without disruption to normal business processes or without incurring excessive overhead (nonproductive processing) is largely determined by the *scalability* of the particular computing platform.

Scalability

The ability of a system to retain performance levels when adding processors, memory, and storage.

By scalability, we mean the ability of the hardware, software, or a distributed system to continue to function well as it is changed in size or volume; for example, the ability to retain performance levels when adding processors, memory, and storage. A scalable system can efficiently adapt to work, with larger or smaller networks performing tasks of varying complexity.

As a company grows in employees, customers, and business partners, it usually needs to add computing resources to support business growth. One approach is to add more processors of the same size, with the resulting overhead in managing this more complex setup. Alternatively, a company can consolidate its many smaller processors into fewer, larger systems. Using a mainframe system, many companies have significantly lowered their total cost of ownership (TCO), which includes not only the cost of the machine (its hardware and software), but the cost to run it.

Mainframes exhibit scalability characteristics in both hardware and software, with the ability to run multiple copies of the operating system software as a single entity called a system complex, or *sysplex*. We further explore mainframe clustering technology and its uses in 2.8, "What is a Parallel Sysplex?" on page 56.

1.7.4 Continuing compatibility

Mainframe customers tend to have a very large financial investment in their applications and data. Some applications have been developed and refined over decades. Some applications were written many years ago, while others may have been written “yesterday.” The ability of an application to work in the system or its ability to work with other devices or programs is called *compatibility*.

The need to support applications of varying ages imposes a strict compatibility demand on mainframe hardware and software, which have been upgraded many times since the first System/360 mainframe computer was shipped in 1964. Applications *must* continue to work properly. Thus, much of the design work for new hardware and system software revolves around this compatibility requirement.

Compatibility

The ability of a system both to run software requiring new hardware instructions and to run older software requiring the original hardware instructions.

The overriding need for compatibility is also the primary reason why many aspects of the system work as they do, for example, the syntax restrictions of the job control language (JCL) that is used to control batch jobs. Any new design enhancements made to JCL must preserve compatibility with older jobs so that they can continue to run without modification. The desire and need for continuing compatibility is one of the defining characteristics of mainframe computing.

Absolute compatibility across decades of changes and enhancements is not possible, of course, but the designers of mainframe hardware and software make it a top priority. When an incompatibility is unavoidable, the designers typically warn users *at least a year* in advance that software changes might be needed.

1.8 Typical mainframe workloads

Most mainframe workloads fall into one of two categories: batch processing or online transaction processing, which includes Web-based applications (Figure 1-1).

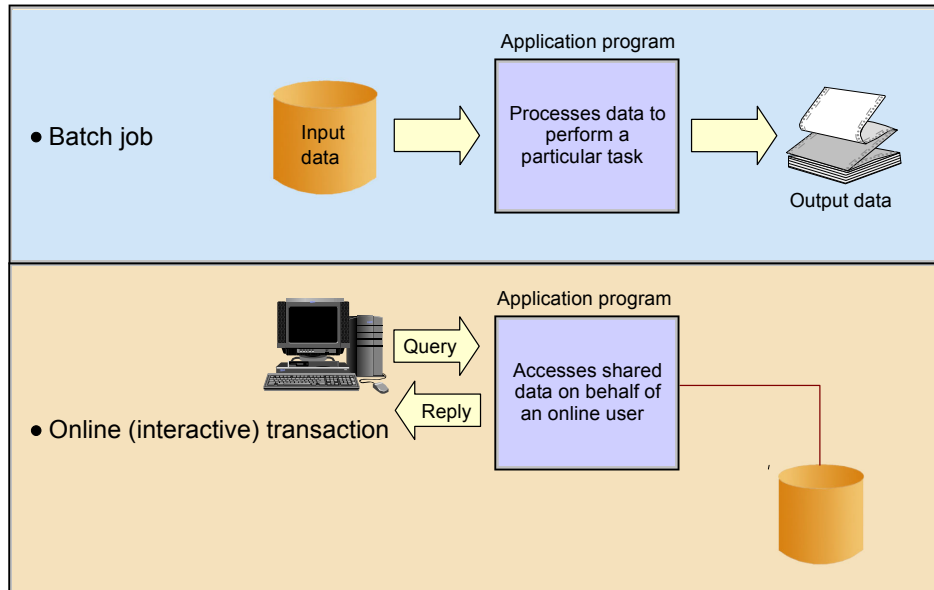


Figure 1-1 Typical mainframe workloads

These workloads are discussed in several chapters in this text; the following sections provide an overview.

1.8.1 Batch processing

One key advantage of mainframe systems is their ability to process terabytes of data from high-speed storage devices and produce valuable output. For example, mainframe systems make it possible for banks and other financial institutions to perform end-of-quarter processing and produce reports that are necessary to customers (for example, quarterly stock statements or pension statements) or to the government (for example, financial results). With mainframe systems, retail stores can generate and consolidate nightly sales reports for review by regional sales managers.

Batch processing
The running of jobs on the mainframe without user interaction.

The applications that produce these statements are *batch* applications; that is, they are processed on the mainframe without user interaction. A *batch job* is submitted on the computer, reads and processes data in bulk—perhaps terabytes of data—and produces output, such as customer billing statements. An equivalent concept can be found in a UNIX script file or a Windows® command file, but a z/OS batch job might process millions of records.

While batch processing is possible on distributed systems, it is not as commonplace as it is on mainframes because distributed systems often lack:

- ▶ Sufficient data storage
- ▶ Available processor capacity, or *cycles*
- ▶ Sysplex-wide management of system resources and job scheduling

Mainframe operating systems are typically equipped with sophisticated job scheduling software that allows data center staff to submit, manage, and track the execution and output of batch jobs⁶.

Batch processes typically have the following characteristics:

- ▶ Large amounts of input data are processed and stored (perhaps terabytes or more), large numbers of records are accessed, and a large volume of output is produced.
- ▶ Immediate response time is usually not a requirement. However, batch jobs often must complete within a “batch window,” a period of less-intensive online activity, as prescribed by a *service level agreement (SLA)*.
- ▶ Information is generated about large numbers of users or data entities (for example, customer orders or a retailer’s stock on hand).
- ▶ A scheduled batch process can consist of the execution of hundreds or thousands of jobs in a pre-established sequence.

During batch processing, multiple types of work can be generated. Consolidated information such as profitability of investment funds, scheduled database backups, processing of daily orders, and updating of inventories are common examples. Figure 1-2 shows a number of batch jobs running in a typical mainframe environment.

In Figure 1-2, consider the following elements at work in the scheduled batch process:

1. At night, numerous batch jobs running programs and utilities are processed. These jobs consolidate the results of the online transactions that take place during the day.
2. The batch jobs generate reports of business statistics.
3. Backups of critical files and databases are made before and after the batch window.
4. Reports with business statistics are sent to a specific area for analysis the next day.

⁶ In the early days of the mainframe, punched cards were often used to enter jobs into the system for execution. “Key-punch operators” used card punches to enter data, and decks of cards (or batches) were produced. These were fed into card readers, which read the jobs and data into the system. As you can imagine, this process was cumbersome and error-prone. Nowadays, it is possible to transfer the equivalent of punched card data to the mainframe in a PC text file. We discuss various ways of introducing work into the mainframe in Chapter 7, “Batch processing and JES” on page 245.

5. Reports with exceptions are sent to the branch offices.
6. Monthly account balance reports are generated and sent to all bank customers.
7. Reports with processing summaries are sent to the partner credit card company.

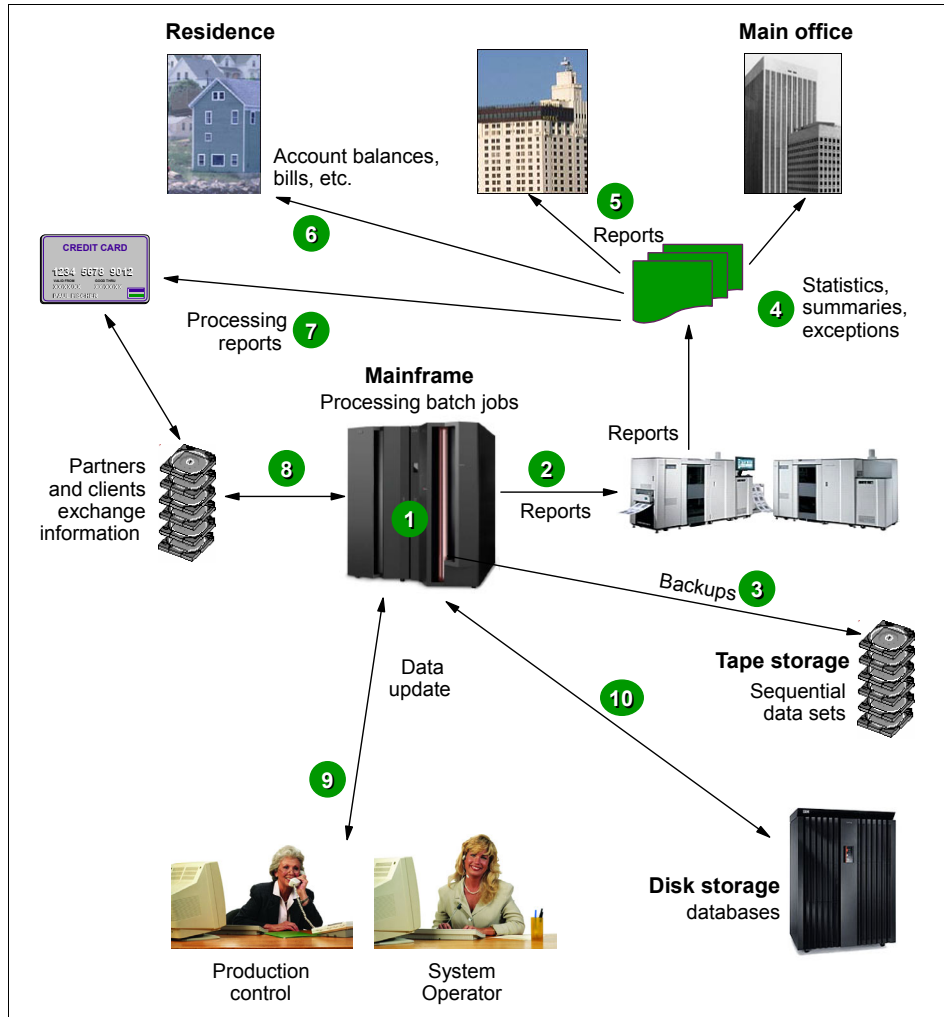


Figure 1-2 Typical batch use

8. A credit card transaction report is received from the partner company.
9. In the production control department, the operations area is monitoring the messages on the system console and the execution of the jobs.

10. Jobs and transactions are reading or updating the database (the same one that is used by online transactions) and many files are written to tape.

1.8.2 Online transaction processing

Transaction processing that occurs interactively with the end user is referred to as *online transaction processing* or OLTP. Typically, mainframes serve a vast number of *transaction systems*. These systems are often mission-critical applications that businesses depend on for their core functions. Transaction systems must be able to support an unpredictable number of concurrent users and transaction types. Most transactions are executed in short time periods—fractions of a second in some cases.

One of the main characteristics of a transaction system is that the interactions between the user and the system are very short. The user will perform a complete business transaction through short interactions, with immediate response time required for each interaction. These systems are currently supporting mission-critical applications; therefore, continuous availability, high performance, and data protection and integrity are required.

Online transactions are familiar to most people. Examples include:

- ▶ ATM machine transactions such as deposits, withdrawals, inquiries, and transfers
- ▶ Supermarket payments with debit or credit cards
- ▶ Purchase of merchandise over the Internet

Online transaction processing (OLTP)

Transaction processing that occurs interactively with the end user.

For example, inside a bank branch office or on the Internet, customers are using online services when checking an account balance or directing fund balances.

In fact, an online system performs many of the same functions as an operating system:

- ▶ Managing and dispatching tasks
- ▶ Controlling user access authority to system resources
- ▶ Managing the use of memory
- ▶ Managing and controlling simultaneous access to data files
- ▶ Providing device independence

Some industry uses of mainframe-based online systems include:

- ▶ Banks - ATMs, teller systems for customer service
- ▶ Insurance - Agent systems for policy management and claims processing
- ▶ Travel and transport - Airline reservation systems
- ▶ Manufacturing - Inventory control, production scheduling
- ▶ Government - Tax processing, license issuance and management

How might the end users in these industries interact with their mainframe systems? Multiple factors can influence the design of a company's transaction processing system, including:

- ▶ Number of users interacting with the system at any one time.
- ▶ Number of transactions per second (TPS).
- ▶ Availability requirements of the application. For example, must the application be available 24 hours a day, seven days a week, or can it be brought down briefly one night each week?

Before personal computers and intelligent workstations became popular, the most common way to communicate with online mainframe applications was with 3270 terminals. These devices were sometimes known as “dumb” terminals, but they had enough intelligence to collect and display a full screen of data rather than interacting with the computer for each keystroke, saving processor cycles. The characters were green on a black screen, so the mainframe applications were nicknamed “green screen” applications.

Based on these factors, user interactions vary from installation to installation. With applications now being designed, many installations are reworking their existing mainframe applications to include Web browser-based interfaces for users. This work sometimes requires new application development, but can often be done with vendor software purchased to “re-face” the application. Here, the end user often does not realize that there is a mainframe behind the scenes.

In this text, there is no need to describe the process of interacting with the mainframe through a Web browser, as it is exactly the same as any interaction a user would have through the Web. The only difference is the machine at the other end!

Online transactions usually have the following characteristics:

- ▶ A small amount of input data, a few stored records accessed and processed, and a small amount of data as output
- ▶ Immediate response time, usually less than one second
- ▶ Large numbers of users involved in large numbers of transactions
- ▶ Round-the-clock availability of the transactional interface to the user
- ▶ Assurance of security for transactions and user data

In a bank branch office, for example, customers use online services when checking an account balance or making an investment.

Figure 1-3 shows a series of common online transactions using a mainframe.

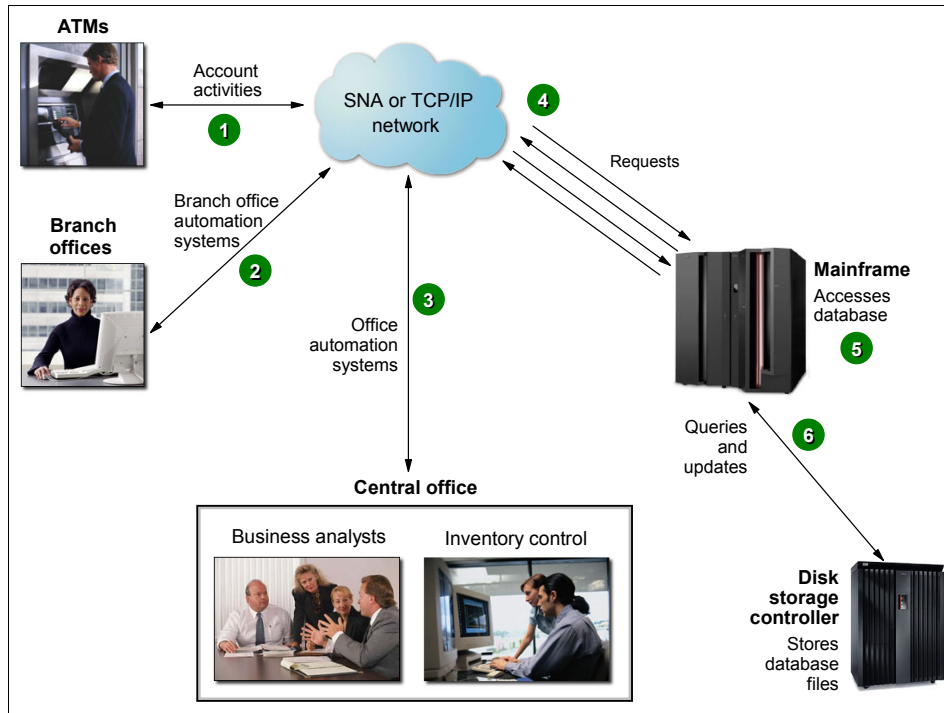


Figure 1-3 Typical online use

1. A customer uses an ATM, which presents a user-friendly interface for various functions: Withdrawal, query account balance, deposit, transfer, or cash advance from a credit card account.
2. Elsewhere in the same private network, a bank employee in a branch office performs operations such as consulting, fund applications, and money ordering.
3. At the bank's central office, business analysts tune transactions for improved performance. Other staff use specialized online systems for office automation to perform customer relationship management, budget planning, and stock control.
4. All requests are directed to the mainframe computer for processing.
5. Programs running on the mainframe computer perform updates and inquiries to the database management system (for example, DB2).
6. Specialized disk storage systems store the database files.

1.9 Roles in the mainframe world

Mainframe systems are designed to be used by large numbers of people. Most of those who interact with mainframes are end users—people who use the applications that are hosted on the system. However, because of the large number of end users, applications running on the system, and the sophistication and complexity of the system software that supports the users and applications, a variety of roles are needed to operate and support the system.

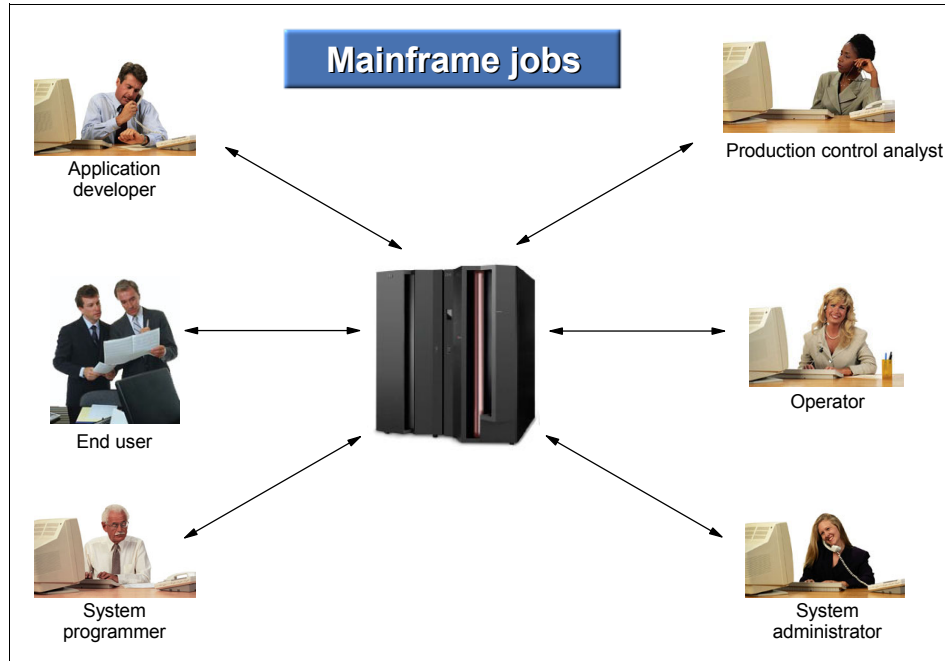


Figure 1-4 Who's who in the mainframe world

In the IT field, these roles are referred to by a number of different titles; this text uses the following:

- ▶ System programmers
- ▶ System administrators
- ▶ Application designers and programmers
- ▶ System operators
- ▶ Production control analysts

In a distributed systems environment, many of the same roles are needed as in the mainframe environment. However, the job responsibilities are often not as well-defined. Since the 1960s, mainframe roles have evolved and expanded to provide an environment in which the system software and applications can

function smoothly and effectively and serve many thousands of users efficiently. While it may seem that the size of the mainframe support staff is large and unwieldy, the numbers become comparatively small when one considers the number of users supported, the number of transactions run, and the high business value of the work that is performed on the mainframe.

This text is concerned mainly with the system programmer and application programmer roles in the mainframe environment. There are, however, several other important jobs involved in the “care and feeding” of the mainframe, and we touch on some of these roles to give you a better idea of what’s going on behind the scene.

Mainframe activities, such as the following, often require cooperation among the various roles:

- ▶ Installing and configuring system software
- ▶ Designing and coding new applications to run on the mainframe
- ▶ Introduction and management of new workloads on the system, such as batch jobs and online transaction processing
- ▶ Operation and maintenance of the mainframe software and hardware

In the following sections, we describe each role in more detail.

1.9.1 Who is the system programmer?

In a mainframe IT organization, the *system programmer* (or *systems programmer*) plays a central role. The system programmer installs, customizes, and maintains the operating system, and also installs or upgrades products that run on the system. The system programmer might be presented with the latest version of the operating system to upgrade the existing systems. Or, the installation might be as simple as upgrading a single program, such as a sort application.

System programmer
The person who installs, customizes, and maintains the operating system.

The system programmer performs such tasks as the following:

- ▶ Planning hardware and software system upgrades and changes in configuration
- ▶ Training system operators and application programmers
- ▶ Automating operations
- ▶ Capacity planning
- ▶ Running installation jobs and scripts
- ▶ Performing installation-specific customization tasks

- ▶ Integration-testing the new products with existing applications and user procedures
- ▶ System-wide performance tuning to meet required levels of service

The system programmer must be skilled at debugging problems with system software. These problems are often captured in a copy of the computer's memory contents called a *dump*, which the system produces in response to a failing software product, user job, or transaction. Armed with a dump and specialized debugging tools, the system programmer can determine where the components have failed. When the error has occurred in a software product, the system programmer works directly with the software vendor's support representatives to discover whether the problem's cause is known and whether a patch is available.

System programmers are needed to install and maintain the *middleware* on the mainframe, such as database management systems, online transaction processing systems and Web servers. Middleware is a software "layer" between the operating system and the end user or end user application. It supplies major functions that are not provided by the operating system. Major middleware products such as DB2, CICS, and IMS™ can be as complex as the operating system itself, if not more so.

1.9.2 Who is the system administrator?

The distinction between system programmer and system administrator varies widely among mainframe sites. In smaller IT organizations, where one person might be called upon to perform several roles, the terms may be used interchangeably.

System administrator

The person who maintains the critical business data that resides on the mainframe.

In larger IT organizations with multiple departments, the job responsibilities tend to be more clearly separated. System administrators perform more of the day-to-day tasks related to maintaining the critical business data that resides on the mainframe, while the system programmer focuses on maintaining the system itself. One reason for the separation of duties is to comply with auditing procedures, which often require that no one person in the IT organization be allowed to have unlimited access to sensitive data or resources. Examples of system administrators include the database administrator (DBA) and the security administrator.

While system programmer expertise lies mainly in the mainframe hardware and software areas, system administrators are more likely to have experience with the applications. They often interface directly with the application programmers and end users to make sure that the administrative aspects of the applications are met. These roles are not necessarily unique to the mainframe environment, but they are key to its smooth operation nonetheless.

In larger IT organizations, the system administrator maintains the system software environment for business purposes, including the day-to-day maintenance of systems to keep them running smoothly. For example, the database administrator must ensure the integrity of, and efficient access to, the data that is stored in the database management systems.

Other examples of common system administrator tasks can include:

- ▶ Installing software
- ▶ Adding and deleting users and maintaining user profiles
- ▶ Maintaining security resource access lists
- ▶ Managing storage devices and printers
- ▶ Managing networks and connectivity
- ▶ Monitoring system performance

In matters of problem determination, the system administrator generally relies on the software vendor support center personnel to diagnose problems, read dumps, and identify corrections for cases in which these tasks aren't performed by the system programmer.

1.9.3 Who are the application designers and programmers?

The application designer and application programmer (or application developer) design, build, test, and deliver mainframe applications for the company's end users and customers. Based on requirements gathered from business analysts and end users, the designer creates a design specification from which the programmer constructs an application. The process includes several iterations of code changes and compilation, application builds, and unit testing.

During the application development process, the designer and programmer must interact with other roles in the enterprise. For example, the programmer often works on a team of other programmers who are building code for related application program modules. When completed, each module is passed through a testing process that can include function, integration, and system-wide tests. Following the tests, the application programs must be acceptance tested by the user community to determine whether the code actually satisfies the original user requirement.

In addition to creating new application code, the programmer is responsible for maintaining and enhancing the company's existing mainframe applications. In fact, this is often the primary job for many of today's mainframe application programmers. While mainframe installations still create new programs with Common Business Oriented Language (COBOL) or PL/I, languages such as Java™ have become popular for building new applications on the mainframe, just as they have on distributed platforms.

Widespread development of mainframe programs written in high-level languages such as COBOL and PL/I continues at a brisk pace, despite rumors to the contrary. Many thousands of programs are in production on mainframe systems around the world, and these programs are critical to the day-to-day business of the corporations that use them. COBOL and other high-level language programmers are needed to maintain existing code and make updates and modifications to existing programs. Also, many corporations continue to build new application logic in COBOL and other traditional languages, and IBM continues to enhance their high-level language compilers to include new functions and features that allow those languages to continue to take advantage of newer technologies and data formats.

We will look at the roles of application designer and application programmer in more detail in Part 2 of this text.

1.9.4 Who is the system operator?

The system operator monitors and controls the operation of the mainframe hardware and software. The operator starts and stops system tasks, monitors the system consoles for unusual conditions, and works with the system programming and production control staff to ensure the health and normal operation of the systems.

System operator

The person who monitors and controls the operation of the mainframe hardware and software.

As applications are added to the mainframe, the system operator is responsible for ensuring that they run smoothly. New applications from the Applications Programming Department are typically delivered to the Operations Staff with a *run book* of instructions. A run book identifies the specific operational requirements of the application, which operators need to be aware of during job execution. Run book instructions might include, for example: application-specific console messages that require operator intervention, recommended operator responses to specific system events, and directions for modifying job flows to accommodate changes in business requirements⁷.

The operator is also responsible for starting and stopping the major subsystems, such as transaction processing systems, database systems, and the operating system itself. These *restart operations* are not nearly as commonplace as they once were, as the availability of the mainframe has improved dramatically over the years. However, the operator must still perform an orderly shutdown and startup of the system and its workloads, when it is required.

⁷ Console messages were once so voluminous that operators often had a difficult time determining whether a situation was really a problem. In recent years, tools to reduce the volume of messages and automate message responses to routine situations have made it easier for operators to concentrate on unusual events that might require human intervention.

In case of a failure or an unusual situation, the operator communicates with system programmers, who assist the operator in determining the proper course of action, and with the production control analyst, who works with the operator to make sure that production workloads are completing properly.

1.9.5 Who is the production control analyst?

Production control analyst

The person who ensures that batch workloads run to completion without error or delay.

The production control analyst is responsible for making sure that batch workloads run to completion—without error or delay. Some mainframe installations run interactive workloads for online users, followed by batch updates that run after the prime shift when the online systems are not running. While this execution model is still common, world-wide operations at many companies—with live, Internet-based access to production data—are finding the “daytime online/night time batch” model to be obsolete. Batch workloads continue to be a part of information processing, however, and skilled production control analysts play a key role.

A common complaint about mainframe systems is that they are inflexible and hard to work with, specifically in terms of implementing changes. The production control analyst often hears this type of complaint, but understands that the use of well-structured rules and procedures to control changes—a strength of the mainframe environment—helps to prevent outages. In fact, one reason that mainframes have attained a strong reputation for high levels of availability and performance is that there are controls on change and it is difficult to introduce change without proper procedures.

1.9.6 What role do vendors play?

A number of vendor roles are commonplace in the mainframe shop. Because most mainframe computers are sold by IBM, and the operating systems and primary online systems are also provided by IBM, most vendor contacts are IBM employees. However, independent software vendor (ISV) products are also used in the IBM mainframe environment, and customers use original equipment manufacturer (OEM) hardware, such as disk and tape storage devices, as well.

Typical vendor roles follow:

- ▶ *Hardware support or customer engineer*

Hardware vendors usually provide on-site support for hardware devices. The IBM hardware maintenance person is often referred to as the customer engineer (CE). The CE provides installation and repair service for the mainframe hardware and peripherals. The CE usually works directly with the operations teams when hardware fails or new hardware is being installed.

► *Software support*

A number of vendor roles exist to support software products on the mainframe⁸. IBM has a centralized *Support Center* that provides entitled and extra-charge support for software defects or usage assistance. There are also information technology specialists and architects who can be engaged to provide additional pre- and post-sales support for software products, depending upon the size of the enterprise and the particular customer situation.

► *Field technical sales support, systems engineer, or client representative*

For larger mainframe accounts, IBM and other vendors provide face-to-face sales support. The vendor representatives specialize in various types of hardware or software product families and call on the part of the customer organization that influences the product purchases. At IBM, the technical sales specialist is referred to as the *field technical sales support (FTSS)* person, or by the older term, *systems engineer (SE)*.

For larger mainframe accounts, IBM frequently assigns a client representative, who is attuned to the business issues of a particular industry sector, to work exclusively with a small number of customers. The client representative acts as the general “single point of contact” between the customer and the various organizations within IBM.

1.10 z/OS and other mainframe operating systems

Much of this text is concerned with teaching you the fundamentals of z/OS, which is IBM’s foremost mainframe operating system. We begin discussing z/OS concepts in Chapter 3, “z/OS overview” on page 81. It is useful for mainframe students, however, to have a working knowledge of other mainframe operating systems. One reason is that a given mainframe computer might run multiple operating systems. For example, the use of z/OS, z/VM®, and Linux® on the same mainframe is common.

Mainframe operating systems are sophisticated products with substantially different characteristics and purposes, and each could justify a separate book for a detailed introduction. Besides z/OS, four other operating systems dominate mainframe usage: z/VM, z/VSE™, Linux for zSeries, and z/TPF.

⁸ This text does not examine the marketing and pricing of mainframe software. However, the availability and pricing of middleware and other licensed programs is a critical factor affecting the growth and use of mainframes.

1.10.1 z/VM

z/Virtual Machine (z/VM) has two basic components: a *control program* (CP) and a single-user operating system, CMS. As a control program, z/VM is a *hypervisor* because it runs other operating systems in the virtual machines it creates. Any of the IBM mainframe operating systems such as z/OS, Linux for zSeries, z/VSE, and z/TPF can be run as *guest systems* in their own virtual machines, and z/VM can run any combination of guest systems.

The control program artificially creates multiple virtual machines from the real hardware resources. To end users, it appears as if they have dedicated use of the shared real resources. The shared real resources include printers, disk storage devices, and the CPU. The control program ensures data and application security among the guest systems. The real hardware can be shared among the guests, or dedicated to a single guest for performance reasons. The system programmer allocates the real devices among the guests. For most customers, the use of guest systems avoids the need for larger hardware configurations.

z/VM's other major component is the Conversational Monitor System or CMS. This component of z/VM runs in a virtual machine and provides both an interactive end user interface and the general z/VM application programming interface.

1.10.2 z/VSE

z/Virtual Storage Extended (z/VSE) is popular with users of smaller mainframe computers. Some of these customers eventually migrate to z/OS when they grow beyond the capabilities of z/VSE.

Compared to z/OS, the z/VSE operating system provides a smaller, less complex base for batch processing and transaction processing. The design and management structure of z/VSE is excellent for running routine production workloads consisting of multiple batch jobs (running in parallel) and extensive, traditional transaction processing. In practice, most z/VSE users also have the z/VM operating system and use this as a general terminal interface for z/VSE application development and system management.

z/VSE was originally known as Disk Operating System (DOS), and was the first disk-based operating system introduced for the System/360 mainframe computers. DOS was seen as a temporary measure until OS/360 would be ready. However, some mainframe customers liked its simplicity (and small size) and decided to remain with it after OS/360 became available. DOS became known as DOS/VS (when it started using virtual storage), then VSE/SP and later VSE/ESA™, and most recently z/VSE. The name VSE is often used collectively to refer to any of the more recent versions.

1.10.3 Linux for zSeries

Several (non-IBM) Linux distributions can be used on a mainframe. There are two generic names for these distributions:

- ▶ Linux for S/390 (uses 31-bit addressing and 32-bit registers)
- ▶ Linux for zSeries (uses 64-bit addressing and registers)

The phrase *Linux on zSeries* is used to refer to Linux running on an S/390 or zSeries system, when there is no specific need to refer explicitly to either the 31-bit version or the 64-bit version. We assume students are generally familiar with Linux and therefore we mention only those characteristics that are relevant for mainframe usage. These include the following:

Linux uses traditional count key data (CKD)⁹ disk devices and SAN-connected SCSI-type devices. Other mainframe operating systems can recognize these drives as Linux drives, but cannot use the data formats on the drives. That is, there is no sharing of data between Linux and other mainframe operating systems.

- ▶ Linux does not use 3270 display terminals, while all other mainframe operating systems use 3270s as their basic terminal architecture.¹⁰ Linux uses X Window System based terminals or X-Window System emulators on PCs; it also supports typical ASCII terminals, usually connected through the *telnet* protocol. The X-Window System is the standard for graphical interfaces in Linux. It is the middle layer between the hardware and the window manager.
- ▶ With the proper setup, a Linux system under z/VM can be quickly cloned to make another, separate Linux image. The z/VM emulated LAN can be used to connect multiple Linux images and to provide an external LAN route for them. Read-only file systems, such as a typical /usr file system, can be shared by Linux images.
- ▶ Linux on a mainframe operates with the ASCII character set, not the EBCDIC¹¹ form of stored data that is typically used on mainframes. Here, EBCDIC is used only when writing to such character-sensitive devices as displays and printers. The Linux drivers for these devices handle the character translation.

⁹ CKD devices are formatted such that the individual data pieces can be accessed directly by the read head of the disk.

¹⁰ There is a Linux driver for minimal 3270 operation, in very restrictive modes, but this is not commonly used. 3270 terminals were full-screen buffered non-intelligent terminals, with control units and data streams to maximize efficiency of data transmission.

¹¹ EBCDIC, which stands for extended binary coded decimal interchange code, is a coded character set of 256 8-bit characters that was developed for the representation of textual data. EBCDIC is not compatible with ASCII character coding. For a handy conversion table, see Appendix D, "EBCDIC - ASCII table" on page 665.

1.10.4 z/TPF

The z/Transaction Processing Facility (z/TPF) operating system is a special-purpose system that is used by companies with very high transaction volume, such as credit card companies and airline reservation systems. z/TPF was once known as Airline Control Program (ACP). It is still used by airlines and has been extended for other very large systems with high-speed, high-volume transaction processing requirements.

z/TPF can use multiple mainframes in a loosely-coupled environment to routinely handle tens of thousands of transactions per second, while experiencing uninterrupted availability that is measured in years. Very large terminal networks, including special-protocol networks used by portions of the reservation industry, are common.

1.11 Summary

Today, mainframe computers play a central role in the daily operations of most of the world's largest corporations, including many Fortune 1000 companies. While other forms of computing are used extensively in business in various capacities, the mainframe occupies a coveted place in today's e-business environment. In banking, finance, health care, insurance, utilities, government, and a multitude of other public and private enterprises, the mainframe computer continues to form the foundation of modern business.

The New Mainframe owes much of its popularity and longevity to its inherent reliability and stability, a result of continuous technological advances since the introduction of the IBM System/360 in 1964. No other computer architecture in existence can claim as much continuous, evolutionary improvement, while maintaining compatibility with existing applications.

The term *mainframe* has gradually moved from a physical description of IBM's larger computers to the categorization of a style of computing. One defining characteristic of the mainframe has been a continuing compatibility that spans decades.

The roles and responsibilities in a mainframe IT organization are wide and varied. It takes skilled staff to keep a mainframe computer running smoothly and reliably. It might seem that there are far more resources needed in a mainframe environment than for small, distributed systems. But, if roles are fully identified on the distributed systems side, a number of the same roles exist there as well.

Several operating systems are currently available for mainframes. This text concentrates on one of these, z/OS. However, mainframe students should be

aware of the existence of the other operating systems and understand their positions relative to z/OS.

Key terms in this chapter				
architecture	availability	batch processing	compatibility	e-business
mainframe	online transaction processing (OLTP)	platform	production control analyst	run book
scalability	scalability	system operator	system programmer	System/360

1.12 Questions for review

To help test your understanding of the material in this chapter, complete the following questions:

1. List ways in which the mainframe of today challenges the traditional thinking about centralized computing versus distributed computing.
2. Explain how businesses make use of mainframe processing power, and how mainframe computing differs from other types of computing.
3. List three strengths of mainframe computing, and outline the major types of workloads for which mainframes are best suited.
4. Name five jobs or responsibilities that are related to mainframe computing.
5. This chapter mentioned at least five operating systems that are used on the mainframe. Choose three of them and describe the main characteristics of each.

1.13 Topics for further discussion

1. What is a mainframe today? How did the term arise? Is it still appropriate?
2. Why is it important to maintain system compatibility for older applications? Why not simply change existing application programming interfaces whenever improved interfaces become available?
3. Describe how running a mainframe can be cost effective, given the large number of roles needed to run a mainframe system.

4. What characteristics, good or bad, exist in a mainframe processing environment because of the roles that are present in a mainframe shop? (Efficiency? Reliability? Scalability?)
5. Most mainframe shops have implemented rigorous systems management, security, and operational procedures. Have these same procedures been implemented in distributed system environments? Why or why not?
6. Can you find examples of mainframe use in your everyday experiences? Describe them and the extent to which mainframe processing is apparent to end users. Examples might include the following:
 - Popular Web sites that rely on mainframe technology as the back-end server to support online transactions and databases.
 - Mainframes used in your locality. These might include banks and financial centers, major retailers, transportation hubs, and the health and medical industries.
7. Can you find examples of distributed systems in everyday use? Could any of these systems be improved through the addition of a mainframe? How?