Chapter 12

System Management

At a Glance

Instructor's Manual Table of Contents

Lecture Notes

Overview

In earlier chapters, each component of the operating system was examined in isolation. However, in a real-life operating system, the pieces do not work in isolation. Instead, each component depends on the other components for smooth and seamless operation.

This chapter shows how the different components of an operating system work together. It explains the tradeoffs system designers consider when attempting an improvement for the system's overall efficiency. The chapter begins with a discussion regarding the need to keep the operating system patched correctly. This is followed by explanations of how the designer might improve the performance of one component, the various aspects surrounding the cost of the improvement, and how these changes might affect the performance of the remainder of the system. Next, various methods used to measure and monitor system performance including positive and negative feedback loops are presented. The chapter concludes with a discussion regarding the need to account for the overall operating system.

Learning Objectives

After completing this chapter, the student should be able to describe:

- The fundamentals of patch management
- The tradeoffs to be considered when attempting to improve overall system performance
- The roles of system measurement tools such as positive and negative feedback loops
- Two system monitoring techniques
- The importance of sound accounting practices by system administrators

Teaching Tips

Patch Management

1. Provide students with an overview of patch management. Discuss its importance and necessity. Point out three primary reasons for software patches.

2. Discuss why patching computing systems correctly has become a challenge. Mention that automatic patch management was among the eight technologies used most at security and law enforcement organizations in 2004. Use Figure 12.1 on page 376 to illustrate this ranking.

3. Mention the positive impacts that result from successfully patching a computer system, and discuss accountability of this task by noting that different organizations delegate responsibility for the task to different individuals.

Teaching Tip	Refer to the following Web site for detailed information on Microsoft patch management: www.microsoft.com/technet/security/guidance/patchmanagement.mspx

Patching Fundamentals

1. Outline the essential steps that need to be taken before successful patch installation, using the numbered points on page 377 as a guide. Be sure to stress that the operating system is never patched without a recent data backup in hand in case of failure and a need to recover.

2. Discuss in more detail the first task of identifying the required patch. Explain that this primarily concerns identifying the criticality category of the patch. Point out that this information is available from the vendor.

3. Point out the difference between critical and non-critical patches in terms of urgency of their installation.

4. Discuss in more detail the second task of verifying patch integrity. Explain that this is primarily concerned with validating the source and the integrity of a patch once it is received. Explain how patch integrity verification is ensured and validated.

5. Discuss in more detail the third task of patch testing. Point out that it is important to test the new patch on a sample system or an isolated machine to verify its worth before installation on a live system.

6. Explain the important checkpoints in the testing process, such as whether the system reboots after the patch is installed, whether the patched software performs its assigned tasks, and so on. Emphasize that contingency testing should be considered too at this time in case of patch installation failure.

7. Discuss in more detail the fourth task of patch deployment. Discuss the complexities involved in patch deployment for a multiuser environment, comparing that process to a single-user environment.

8. Review the various steps involved in patch deployment in a multiuser environment.

9. Discuss in more detail the fifth task of auditing the system. Outline various steps that need to be taken to confirm that the resulting system meets expectations. Point out that the process should include documentation of the changes made to the system and the success or failure of each stage of the process.

10. Be sure to mention that it is necessary to keep a log of all system changes for future reference, and to get feedback from users to verify the deployment's success.

Teaching Tip	Never patch your operating system without a recent data backup in hand.

Software Options

1. Discuss two ways in which patches can be managed: manually and automatically using software.

2. Point out the difference between two types of patch deployment software (agentless software and agent-based software). Discuss the advantages and disadvantages of both.

Timing the Patch Cycle

1. Discuss the timing of patch cycles. For example, critical system patches must be applied immediately; however, less-critical patches may be scheduled at the convenience of the systems group.

2. Discuss the advantages of routine patch cycles.

Evaluating an Operating System

1. Provide students with a brief overview of how to evaluate an operating system. Outline the various factors that one should have knowledge of in order to evaluate an operating system.

2. Point out that an operating system's strengths and weaknesses need to be weighed in relation to: (1) who will be using the operating system, (2) on what hardware, and (3) for what purpose.

Cooperation Among Components

1. Explain, using an example such as the one provided on page 380, that the performance of any one resource depends on the performance of the other resources in the system.

2. Outline different factors that need to be considered before making any system improvement. These factors might include evaluating the needs of the system's resources, requirements, managers, and users.

3. Point out that whenever changes are made to a system, one set of problems is often traded for another. Be sure to mention that the key is to consider the performance of the entire system and not just the individual components.

Role of Memory Management

1. Explain briefly the role of memory management. Point out that there is a tradeoff between memory use and CPU overhead. That is, as memory management algorithms grow more complex, the CPU overhead increases and overall performance can suffer.

Role of Processor Management

1. Provide students with an overview of processor management. Point out that multiprogramming requires synchronization between the Memory Manager, the Processor Manager, and the I/O devices.

2. Discuss the tradeoffs to consider in processor management. For example, better use of the CPU versus increased overhead, slower response time, and decreased throughput. Outline several problems related to multiprogramming, using the bullet points on page 381 as a guide.

Role of Device Management

1. Provide students with an overview of device management. Remind students of the ways to improve I/O device utilization, such as buffering, blocking, and rescheduling I/O requests to optimize access times.

2. Discuss the tradeoffs involved in using the device management technique of blocking.

3. Discuss the tradeoffs involved in using the device management technique of buffering.

4. Discuss the tradeoffs involved in using the device management technique of rescheduling I/O requests. Provide examples as shown in Table 12.2 and Figures 12.1 and 12.2 (on pages 382 and 383) to illustrate the tradeoffs involved in rescheduling I/O requests.

5. Point out that rescheduling is not always warranted. Also, mention that when the system is configured, the reordering algorithm is always either on or always off. It cannot be changed by the systems operator without reconfiguration, so the initial setting (on or off) must be determined by evaluating the system on the average.

Teaching Tip	Emphasize that the speeds of both the CPU and the I/O device must be weighed against the time it would take to execute the reordering algorithm.

Role of File Management

1. Provide students with an overview of file management. Explain the two important file management considerations: file organization and volume directory location. Illustrate these concepts using examples as provided on pages 383-384, such as the overhead involved in storing the files noncontiguously and the effect of the location of the volume directory on retrieval time.

2. Point out that different storage allocation schemes offer different degrees of flexibility, but the tradeoff for increased file flexibility is increased CPU overhead.

3. Be sure to point out that file management is closely related to the device on which the files are stored and that designers must consider both issues at the same time when evaluating or modifying computer systems.

4. Use Table 12.3 on page 384 to illustrate how file management is related to the device where the files stored.

Role of Network Management

1. Provide students with an overview of network management, discussing the various functions of the Network Manager.

Teaching Tip	Click on the following link for detailed information regarding the System Management processes within the IBM AIX operating system: http://publib.boulder.ibm.com/infocenter/systems/index.jsp?topic=/com.ibm.aix.baseadmn/doc/baseadmndita/admnconc-kickoff.htm

Measuring System Performance

1. Define the concept of total system performance. Point out that system efficiency is not easily measured. List three major components that affect system efficiency. Briefly discuss the difficulties in measuring system performance.

Measurement Tools

1. List different system performance measures that most designers and analysts rely upon, such as throughput, capacity, response time, turnaround time, resource utilization, availability, and reliability.

2. Discuss the key performance measures above and provide examples for each of them.

3. Explain the concepts of availability and reliability. Make sure students clearly understand terms such as mean time between failures (MTBF) and mean time to repair (MTTR). Provide examples to calculate each of these measures, using the formulas provided in the text.

4. Be sure to mention that these measures of performance cannot be taken in isolation from the workload being handled by the system. Overall, system performance varies from time to time, so it is important to define the actual working environment before generalizing.

Feedback Loops

1. Provide students with a brief overview of a feedback loop, which is a mechanism to monitor a system's resource utilization so adjustments can be made.

2. Using examples, discuss two types of feedback loops: negative and positive feedback loops. Use the flowcharts in Figures 12.3 and 12.4 on pages 388 and 389 to clarify the concepts. Point out their key features and applications.

Monitoring

1. Briefly discuss hardware and software monitors, pointing out their pros and cons. Provide examples for each of them.

2. Point out the differences in system performance monitoring between early systems and today's systems.

3. Discuss a variety of ways to carry out these measurements, especially using production programs and simulation models.

4. Explain the concept of benchmarks. Outline different factors on which benchmark results are highly dependent. Use Table 12.4 on page 391 to explain typical benchmark results.

Teaching Tip	For recent benchmark results and details of the testing conditions, refer to the following Web site: www.tpc.org

Quick Quiz 1

1. Which of the following is correct about device management?
   a. Blocking results in reducing overhead.
   b. Rescheduling requests always results in minimizing I/O times.
   c. Buffering is not an overhead function.
   d. Rescheduling requests is an overhead function.

   Answer: d

2. Reliability is a function of which of the item(s) listed below?
   a. MTBF
   b. MTTR
   c. Throughput
   d. All of the above
   Answer: A

3. (True or False) The location of a volume's directory does not affect retrieval time.
   Answer: False

Accounting

1. Provide students with an overview of accounting. Discuss the complexities involved in accounting for a multiuser environment as compared to the accounting for a single-user environment.

2. Discuss the requirements for distributing computer costs in a multiuser environment.

3. Point out that pricing policies vary from system to system. List the different factors that are taken into consideration when setting up a pricing policy.

4. Using examples, explain that pricing policies are often used as a way to achieve specific operational goals. Discuss the motives behind varying the price of system services, offering reduced rates during off hours, or offering pricing incentives.

5. Discuss the advantages and disadvantages of maintaining billing records online.

Teaching Tip	The accounting function may seem mundane; however, from a practical standpoint, it may contain the most important elements of the system.

Quick Quiz 2

1. Which of the following formulas correctly calculates "availability"?
   a. MTBF/(MTBF+MTTR)
   b. MTTR/(MTTR+MTBF)
   c. (MTBF+MTTR)/MTBF
   d. (MTBF+MTTR)/MTTR
   Answer: a

2. A(n) _____ is an executable program that repairs errors or omissions in another program or piece of software.
   Answer: patch

3. (True or False) Agentless deployment software is attractive for administrators of large, complex networks.
   Answer: True

Class Discussion Topics

1. Have students discuss their thoughts on the tradeoffs involved when attempting to improve overall system performance. Ask them to provide five examples where an increase in the performance of one resource results in compromising the performance of another resource in the system.

2. Have students discuss their experience of Microsoft's patch update process. Do they feel that since Microsoft started an automated process to update software, it has resulted in reducing the number of computer attacks significantly? Why or why not?

Additional Projects

1. Have students research online to find details on Microsoft's software updates services, which are aimed at providing security. Ask students to compile a list of key features of their services.

2. Have students research online to find out ways to install more than one update or patch in a manner that is both efficient and less time-consuming in an environment of large, complex networks.

Additional Resources

1. Intel Corporation:
   www.intel.com/

2. Freescale Corporation:
   www.freescale.com/

3. IBM Corporation:
   www.ibm.com/

4. Microsoft Corporation:
   www.microsoft.com

5. Microsoft Security Guidance Kit:
   www.microsoft.com/security/guidance

6. Microsoft's Digital Signature:
   www.microsoft.com/technet/security/bulletin/pgp.mspx

7. Microsoft Software Update Services: www.microsoft.com/windowsserversystem/sus/default.mspx

Key Terms

- **Availability:** a resource measurement tool that indicates the likelihood that the resource will be ready when a user needs it. It's influenced by mean time between failures and mean time to repair.
- **Benchmarks:** measurement tool used to objectively measure and evaluate a system's performance by running a set of jobs representative of the work normally done by a computer system.
- **Capacity:** the maximum throughput level of any one of the system's components.
- **Feedback loop:** a mechanism to monitor the system's resource utilization so adjustments can be made.
- **Mean time between failures (MTBF):** a resource measurement tool; the average time that a unit is operational before it breaks down.
- **Mean time to repair (MTTR):** a resource measurement tool; the average time needed to fix a failed unit and put it back in service.
- **Negative feedback loop:** a mechanism to monitor the system's resources and, when it becomes too congested, to signal the appropriate manager to slow down the arrival rate of the processes.
- **Patch:** executable software that repairs errors or omissions in another program or piece of software.
- **Patch management:** the timely installation of software patches to make repairs and keep the operating system software current.
- **Positive feedback loop:** a mechanism used to monitor the system. When the system becomes underutilized, the feedback causes the arrival rate to increase.
- **Reliability:** a standard that measures the probability that a unit will not fail during a given time period. It's a function of mean time between failures.
- **Resource utilization:** a measure of how much each unit is contributing to the overall operation of the system.
- **Response time:** a measure of an interactive system's efficiency that tracks the speed with which the system will respond to a user's command.
- **Service pack:** a term used by some vendors to describe an update to customer software to repair existing problems and/or deliver enhancements.
- **Throughput:** a composite measure of a system's efficiency that counts the number of jobs served in a given unit of time.
- **Turnaround time:** a measure of a system's efficiency that tracks the time required to execute a job and return output to the user.