Tecnológico de Costa Rica

Escuela de Ingeniería en Computación
Principios de Sistemas Operativos

15 de agosto de 2017

Samantha Arburola León
2013101697

Operating System Concepts

Chapter 2 - Practice Exercises

2.12 The services and functions provided by an operating system can be divided into two main categories. Briefly describe the two categories, and discuss how they differ.

First one is to enforce protection between different processes running concurrently in the system; here processes are not allowed to corrupt files associated with other users, also not allowed to access devices directly without operating system intervention.

The second is to provide new functionality that is not supported directly by the underlying hardware. Virtual memory and file systems are two such examples of new services provided by an operating system

2.13 Describe three general methods for passing parameters to the operating system.

a. Pass parameters in registers
b. Registers pass starting addresses of blocks of parameters
c. Parameters can be placed, or pushed, onto the stack by the program, and popped off the stack by the operating system

2.14 Describe how you could obtain a statistical profile of the amount of time spent by a program executing different sections of its code. Discuss the importance of obtaining such a statistical profile.

A statistical profile of which pieces of code were active should be consistent with the time spent by the program in different sections of its code. Once such a statistical profile has been obtained, the programmer could optimize those sections of code that are consuming more of the CPU resources

2.15 What are the five major activities of an operating system with regard to file management?

1. Create and delete files
2. Create and delete directories
3. Support of primitives for manipulating files and directories
4. Mapping of files into secondary storage
5. BackUp of files on stable storage

2.16 What are the advantages and disadvantages of using the same system call interface for manipulating both files and devices?

Advantages :

- add a new device driver by implementing the hardware-specific code to support this abstract file interface.
- can be written to access devices and files in the same manner
- device driver code, which can be written to support a well-defined API.

Disadvantage:

- it might be difficult to capture the functionality of certain devices within the context of the file access API,
- loss of functionality or a loss of performance
- overcome by the use of the ioctl operation that provides a general-purpose interface for processes to invoke operations on devices.

2.17 Would it be possible for the user to develop a new command interpreter using the system-call interface provided by the operating system?

The command interpreter allows an user to create and manage processes and also determine ways by which they communicate, it could be accessed by an user-level program using the system calls

2.18 What are the two models of interprocess communication? What are the strengths and weaknesses of the two approaches?

Message-passing model: is useful for exchanging smaller amounts of data, also easier to implement than is shared memory for intercomputer communication

Shared-memory model: allows maximum speed and convenience of communication, since it can be done at memory transfer speeds when it takes place within a computer but it compromises on protection and synchronization between the processes sharing memory

2.19 Why is the separation of mechanism and policy desirable?

Must be separate to ensure that systems are easy to modify and the policy may be changed at will while the mechanism stays unchanged, so it provides a more flexible system.

2.20 It is sometimes difficult to achieve a layered approach if two components of the operating system are dependent on each other. Identify a scenario in which it is unclear how to layer two system components that require tight coupling of their functionalities.

When an update is executed, it use virtual memory for download the updates files before apply them on the file system, so memory has to be coordinate for make that process. A mini-example of this behavior can be see when you *Copy & Paste* an element, it is saved in a temporary memory before change its location to a file.

2.21 What is the main advantage of the microkernel approach to system design? How do user programs and system services interact in a microkernel architecture? What are the disadvantages of using the microkernel approach?

The **main advantage** of microkernel is facility for extend the operative system, then adding a new service does not require modifying the kernel, it is more secure as more operations are done in user mode than in kernel mode, and a simpler kernel design and functionality typically results in a more reliable operating system.

User programs and system services **interact** with it using interprocess communication mechanisms such as messaging.

The **main disadvantage** is associated with interprocess communication and the frequent use of the operating system's messaging functions in order to enable the user process and the system service to interact with each other

2.22 What are the advantages of using loadable kernel modules?

The **advantage** of using loadable kernel modules is that functionality can be added to and removed from the kernel while it is running. There is no need to either recompile or reboot the kernel.

2.23 How are iOS and Android similar? How are they different?

Similarities

- Both are based on existing kernels (Linux and Mac OS X)
- Both have architecture that uses software stacks.
- Both provide frameworks for developers.

Differences

- iOS is closed-source, and Android is open-source.
- iOS applications are developed in Objective-C, Android in Java.
- Android uses a virtual machine, and iOS executes code natively.

2.24 Explain why Java programs running on Android systems do not use the standard

Java API and virtual machine.
Because the API and virtual machine are designed for desktop and server systems, not mobile devices. Google developed an API and virtual machine for mobile devices

2.25 The experimental Synthesis operating system has an assembler incorporated in the kernel. To optimize system-call performance, the kernel assembles routines within kernel space to minimize the path that the system call must take through the kernel. This approach is the antithesis of the layered approach, in which the path through the kernel is extended to make building the operating system easier. Discuss the pros and cons of the Synthesis approach to kernel design and system-performance optimization.

Synthesis is impressive due to the performance it achieves through on the compilation. Unfortunately, it is difficult to debug problems within the kernel due to the fluidity of the code, such compilation is system specific, making Synthesis difficult to port.