

## Operating System Concepts

### Practice Exercises

#### 1.1 What are the three main purposes of an operating system?

- Ejecutar programas para que un usuario pueda ejecutar programas en el hardware de la computadora de una manera conveniente y eficiente.
- Para asignar los recursos de la computadora según sea necesario para resolver un problema dado; lo más justo y eficiente posible
- Controlador de la ejecución de los programas de usuario para evitar errores y uso indebido de la computadora, y función y control de dispositivos I/O

#### 1.2 We have stressed the need for an operating system to make efficient use of the computing hardware.

**When is it appropriate for the operating system to forsake this principle and to “waste” resources? Why is such a system not really wasteful?**

Los sistemas de un solo usuario deben maximizar el uso del sistema para el usuario, donde los gráficos pueden “desperdiciar” ciclos de CPU, pero garantizar una buena la interacción del usuario

#### 1.3 What is the main difficulty that a programmer must overcome in writing an operating system for a real-time environment?

Mantenerse dentro de las limitaciones de tiempo, porque si el sistema no completa una tarea en un período de tiempo puede causar un breakdown de todo el sistema que se está ejecutando. Por lo tanto el escritor debe estar seguro de que sus schemes de programación no permiten que el tiempo de respuesta exceda la restricción de tiempo

#### 1.4 Keeping in mind the various definitions of operating system, consider whether the operating system should include applications such as web browsers and mail programs. Argue both that it should and that it should not, and support your answers.

##### Should

Es probable que esté en mejores condiciones de aprovechar las características del kernel, dando ventaja en cuanto a rendimiento

##### Should Not

Vulnerabilidad de seguridad

Las aplicaciones (y sus requerimientos no funcionales) no forman parte del sistema

Conducen a un sistema “bloated”

### 1.5 How does the distinction between kernel mode and user mode function as a rudimentary form of protection (security) system?

#### Kernel Mode

Algunas instrucciones y los dispositivos de hardware pueden ser ejecutadas sólo cuando la CPU está en este modo; en cambio

#### User Mode

La capacidad es muy limitada cuando se ejecuta en modo de usuario, reforzando así la protección de recursos críticos

### 1.6 Which of the following instructions should be privileged?

a. Set value of timer.	Si
b. Read the clock.	No
c. Clear memory.	Si
d. Issue a trap instruction.	No
e. Turn off interrupts.	Si
f. Modify entries in device-status table.	Si
g. Switch from user to kernel mode.	No
h. Access I/O device.	Si

### 1.7 Some early computers protected the operating system by placing it in a memory partition that could not be modified by either the user job or the operating system itself. Describe two difficulties that you think could arise with such a scheme

Los datos requeridos tendrían que ser *almacenados en o pasados a través de* la memoria desprotegida lo que bastaría para ser accesibles a los usuarios no autorizados.

### 1.8 Some CPUs provide for more than two modes of operation. What are two possible uses of these multiple modes?

1. Se podrían usar varios modos para proporcionar una política de seguridad más fina.
2. Proporcionar diferentes distinciones dentro del código del kernel

### 1.9 Timers could be used to compute the current time. Provide a short description of how this could be accomplished.

Un programa podría utilizar un enfoque para calcular el tiempo actual utilizando interrupciones del temporizador. El programa podría establecer un temporizador durante algún tiempo en el futuro y suspenderse, cuando es despertado por la interrupción, podría actualizar su estado local, que está utilizando para seguir la pista del número de interrupciones que ha recibido hasta ahora. Habilitando la repetición de este proceso de establecer continuamente interrupciones de temporizador y actualizar su estado local cuando las interrupciones son realmente elevadas

### **1.10 Give two reasons why caches are useful.**

Los cachés son útiles cuando dos o más componentes necesitan intercambiar datos y los componentes realizan transferencias a diferentes velocidades

#### **What problems do they solve?**

Resuelven el problema de transferencia proporcionando un búfer de velocidad intermedia entre los componentes

#### **What problems do they cause? If a cache can be made as large as the device for which it is caching (for instance, a cache as large as a disk), why not make it that large and eliminate the device?**

Si el dispositivo rápido encuentra los datos que necesita en la memoria caché, no necesita esperar al dispositivo más lento. Los datos de la memoria caché deben mantenerse coherentes con los datos de los componentes. Si un componente tiene un cambio de valor de datos y el dato también está en la memoria caché, también debe actualizarse el caché. Esto es especialmente un problema en sistemas multiprocesador donde más de un proceso puede tener acceso a un dato. Un componente puede ser eliminado por un caché de igual tamaño, pero sólo si:

- a. el caché y el componente tienen capacidad equivalente de ahorro de estado
- b. el caché es asequible, porque el almacenamiento más rápido tiende a ser más caro.

### **1.11 Distinguish between the client–server and peer-to-peer models of distributed systems**

El modelo cliente-servidor distingue firmemente las funciones del cliente y del servidor. Bajo este modelo, el cliente solicita servicios que son proporcionados por el servidor. El modelo peer-to-peer no tiene roles tan estrictos. Por ende, todos los nodos del sistema se consideran pares y por lo tanto pueden actuar como clientes o servidores, o ambos. Un nodo puede solicitar un servicio de otro compañero, o el nodo puede de hecho proporcionar dicho servicio a otros compañeros del sistema.