



ITESO, Universidad Jesuita de Guadalajara

Presentan:

Equipo 4:

Edgar Alan Torres Tovar

Axel Roberto Orozco Hernandez

Profesor: Leonardo Sandoval Gonzalez

Fecha: 19 de Agosto del 2024

Materia: Fundamentos de Sistemas Operativos

1. ¿Cuáles son los Requerimientos del Sistema Operativo para la multiprogramación?

Gestión de memoria eficiente, soporte para la planificación de procesos, aislamiento de procesos, manejo de interrupciones y sincronización de recursos.

2. Discute con tu compañeros de equipo y saca una conclusión de cuáles son las ventajas de la multiprogramación sobre la monoprogramación

Conclusión: La multiprogramación mejora la utilización de la CPU, reduce el tiempo de espera y maximiza el rendimiento al ejecutar múltiples programas simultáneamente, a diferencia de la monoprogramación que solo permite un programa a la vez.

3. ¿Cuál es el objetivo del controlador de DMA?

Transferir los datos directamente entre la memoria y los dispositivos de E/S sin involucrar al CPU, liberándolo para otras tareas.

4. ¿Por qué es importante que un dispositivo pueda interrumpir al procesador?

Permite respuesta inmediata a eventos importantes, mejorando la eficiencia y el manejo de dispositivos en tiempo real.

5. ¿Qué entiendes por concepto de caché y cómo puedes describir la técnica *caching*?

Caché es una memoria rápida que almacena datos frecuentemente usados para acelerar el acceso. Caching es la idea de almacenar datos en cache para mejorar el rendimiento del sistema.

6. ¿Por qué los CPUs actuales proveen diferentes niveles de protección?

Para defender el sistema de fallos y abusos, separando los privilegios entre usuario y del sistema.

7. En los videos se menciona que el CPU provee al menos dos niveles de protección, ¿Qué se logra con más de dos niveles de protección?

Se logra un control más granular y se puede restringir aún más el acceso a recursos críticos, aumentando la seguridad y estabilidad del sistema.

8. ¿Qué es lo que debe protegerse a través de la protección que provee el CPU?

La integridad del sistema operativo, la memoria, los archivos, y los dispositivos de hardware.

9. ¿Por qué el sistema operativo provee llamadas al sistema?

Para ofrecer una interfaz controlada entre las aplicaciones y el hardware, garantizando seguridad y estabilidad.

10. ¿Por qué podemos decir que el sistema operativo es un administrador de recursos?

Por que gestiona y asigna eficientemente los recursos del sistema como CPU, memoria y dispositivos E/S.

11. ¿Por qué podemos ver al sistema operativo como una máquina ampliada?

Abstrae la complejidad del hardware, ofreciendo una interfaz mas facil de usar para los desarrolladores .

12. ¿En qué es responsable el sistema operativo en cuanto a los procesos?

Gestionar la creacion, ejecucion y finalizacion de procesos , asi como la asignacion de recursos.

13. ¿Qué es lo que hace o provee el hardware para proteger el buen uso de los dispositivos de entrada y salida?

Uso de modos de operación y privilegios, asi como el control de acceso mediante interrupciones y controladores.

14. ¿Qué aprendiste?

Aprendimos los conceptos y mecanismos criticos del sistema operativo para la gestion eficiente de recursos, la proteccion del sistema y la integracion con el hardware.

Bibliografias:

Spasojevic, A. (2024, 10 abril). *¿Qué es la multiprogramación?* phoenixNAP IT Glossary.

<https://phoenixnap.mx/glosario/multiprogramaci%C3%B3n>

Studocu. (s. f.). *T1-SO1-Ventajas y desventajas de SO multiprogramación - Ventajas y desventajas de los sistemas* - Studocu.

<https://www.studocu.com/pe/document/universidad-tecnologica-del-peru/sistemas-operativos/t1-so1-ventajas-y-desventajas-de-so-multiprogramacion/24989058>

GeeksforGeeks. (2023, 18 julio). *Direct Memory Access (DMA) Controller in Computer Architecture*. GeeksforGeeks. <https://www.geeksforgeeks.org/direct-memory-access-dma-controller-in-computer-architecture/>

Introducción a Caching: Turbocharging Your Application Workloads. (s. f.).

[Video]. Amazon Web Services, Inc. <https://aws.amazon.com/es/caching/>

Ljubuncic, I. (2015). Getting very geeky – application and kernel cores, kernel debugger.

En *Elsevier eBooks* (pp. 137-210). <https://doi.org/10.1016/b978-0-12-801019-8.00006-4>

Video del impartido por el profesor Elvira. Youtube.

<https://www.youtube.com/playlist?list=PLhMtLV713WEc7rWq2nY2-a26Vxb90JS4X>