

## Repaso: Sistemas Operativos 18 de feb. de 25

### Pregunta 1:

¿Cuál es la diferencia entre el modo usuario y el modo kernel en una CPU?

### Pregunta 2:

¿Qué es la memoria virtual y cómo funciona?

### Pregunta 3:

¿Qué es una llamada al sistema y cuál es su propósito?

### Pregunta 4:

¿Qué es la jerarquía de memoria y por qué es importante?

### Pregunta 5:

¿Cuál de las siguientes estructuras de sistemas operativos es la más tradicional y está compuesta por un solo bloque de código que se ejecuta en modo kernel?

- a) Estructura en Capas
- b) Microkernel
- c) Estructura Monolítica
- d) Exokernel

### Pregunta 6:

¿Qué ventaja principal ofrece la estructura en capas de un sistema operativo?

- a) Mayor seguridad y estabilidad
- b) Facilita el mantenimiento y la modificación del sistema
- c) Mayor rendimiento debido a la comunicación directa con el hardware
- d) Acceso directo al hardware para los programas de usuario

### Pregunta 7:

¿Cuál de los siguientes sistemas operativos es un ejemplo clásico de un sistema monolítico?

- a) Windows NT
- b) MS-DOS
- c) Mach
- d) MINIX

### Pregunta 8:

¿Qué característica define a un microkernel?

- a) Contiene todas las funciones del sistema operativo en un solo bloque de código.
- b) Solo contiene las funciones esenciales, como la gestión de procesos y la comunicación entre procesos.

- c) Proporciona acceso directo al hardware para los programas de usuario.
- d) Divide el sistema operativo en capas o niveles.

**Pregunta 9:**

¿Qué sistema operativo fue desarrollado por Andrew S. Tanenbaum con fines educativos y sirvió de inspiración para el desarrollo de Linux?

- a) Mach
- b) MINIX
- c) UNIX
- d) Windows NT

**Pregunta 10:**

¿Qué generación de computadoras introdujo el uso de transistores y el procesamiento por lotes?

- a) Primera generación
  - b) Segunda generación
  - c) Tercera generación
  - d) Cuarta generación
- 

**Pregunta 11:**

¿Qué es un proceso y cuáles son sus estados principales?

**Pregunta 12:**

¿Cuál es la diferencia entre la fragmentación externa y la fragmentación interna?

**Pregunta 13:**

¿Qué es un sistema de archivos y cuáles son sus componentes principales?

**Pregunta 14:**

¿Qué es la planificación de procesos y cuáles son algunos algoritmos comunes?

**Pregunta 15:**

¿Qué es la comunicación entre procesos (IPC) y cuáles son algunos métodos comunes?

**Pregunta 16:**

¿Qué es un sistema distribuido y cuáles son sus características principales?

**Pregunta 17:**

¿Qué mecanismos de seguridad utiliza un sistema operativo para proteger los recursos del sistema?

**Pregunta 18:**

¿Qué es la paginación y cómo funciona en la gestión de memoria?

**Pregunta 19:**

¿Qué es un archivo FIFO y para qué se utiliza?

**Pregunta 20:**

¿Qué es un sistema de archivos NTFS y en qué se diferencia de FAT32?

## **Simulación**

### **Introducción**

En un sistema operativo multitarea, la planificación de procesos es fundamental para administrar la ejecución de múltiples tareas de manera eficiente. Un **proceso** es una instancia de un programa en ejecución, que requiere recursos como CPU, memoria y dispositivos de entrada/salida para su correcta ejecución. La gestión de estos procesos se realiza mediante algoritmos de planificación que determinan el orden de ejecución y el tiempo que cada proceso puede usar la CPU.

### **Objetivo de la Práctica**

El objetivo de esta práctica es analizar diferentes algoritmos de planificación de procesos y observar cómo afectan los tiempos de ejecución y espera de los procesos en un sistema operativo. Se implementarán los siguientes algoritmos de planificación:

1. **FCFS (First-Come, First-Served)**
2. **SJF (Shortest Job First) sin desalojo**
3. **Round Robin con un quantum de tiempo fijo**

Se analizarán métricas clave como el **tiempo de finalización** y el **tiempo de espera** para comparar la eficiencia de cada algoritmo.

---

### **Definición de Conceptos Clave**

#### **1. Tipos de Procesos**

En la planificación de procesos, se pueden clasificar los procesos según su comportamiento y requerimientos de ejecución:

- **Procesos CPU-Bound:** Son aquellos que requieren un uso intensivo del procesador, con pocos accesos a dispositivos de entrada/salida.
- **Procesos I/O-Bound:** Son procesos que dependen de operaciones de entrada/salida y pasan gran parte del tiempo esperando que se completen estas operaciones.
- **Procesos Interactivos:** Son aquellos que requieren una rápida respuesta del sistema, como los utilizados en interfaces gráficas o servicios en línea.
- **Procesos en Segundo Plano (Background Processes):** Se ejecutan sin intervención del usuario y suelen tener menor prioridad.

## 2. Algoritmos de Planificación de Procesos

### FCFS (First-Come, First-Served)

- Se basa en el principio de que los procesos se ejecutan en el orden en que llegan a la cola de planificación.
- No realiza desalojo de procesos; una vez que un proceso inicia su ejecución, debe completarse antes de que otro comience.
- Puede provocar el **problema de la espera prolongada (convoy effect)**, donde un proceso corto espera mucho tiempo si un proceso largo llegó antes.

### SJF (Shortest Job First) sin desalojo

- Prioriza los procesos con menor tiempo de ejecución.
- Mejora el tiempo de espera promedio respecto a FCFS, ya que los procesos cortos se ejecutan primero.
- Puede provocar **injusticia (starvation)** si los procesos cortos siguen llegando, dejando a los procesos largos esperando indefinidamente.

### Round Robin

- Utiliza un **quantum de tiempo** fijo (en este caso, 2 unidades de tiempo).
- Si un proceso no finaliza en su quantum, es enviado al final de la cola y otro proceso toma la CPU.
- Es eficiente para **procesos interactivos**, ya que garantiza que ningún proceso monopolice el CPU.
- Puede provocar **overhead** si el quantum es demasiado pequeño, debido al cambio frecuente de contexto.

### Enunciado del Problema

Se tienen **cuatro procesos** en un sistema operativo, cada uno con un tiempo de llegada y un tiempo de ejecución definido:

#### ID Tiempo de Llegada Tiempo de Ejecución

1	0	5
2	1	3
3	2	8
4	3	6

El objetivo es determinar los tiempos de finalización y espera de cada proceso utilizando los siguientes algoritmos de planificación:

1. **FCFS**
2. **SJF sin desalojo**
3. **Round Robin con quantum de 2**

Para cada algoritmo, se debe calcular y analizar:

- **Tiempo de finalización:** Momento en que un proceso termina su ejecución.
- **Tiempo de espera:** Tiempo que un proceso pasa en la cola de listos antes de ejecutarse.

### Consideraciones Adicionales

- Se asume que no hay interrupciones por E/S, y todos los procesos requieren solo CPU.
- En caso de empate en los algoritmos, se elige el proceso con menor ID.
- Los procesos deben ejecutarse tan pronto como la CPU esté libre y sigan el orden definido por cada algoritmo.

### Resultados:

El programa muestra una tabla con los resultados de cada algoritmo, permitiendo a los estudiantes comparar cómo cada método afecta el tiempo de espera y finalización de los procesos.

=== Práctica de Planificación de Procesos ===

Se tienen los siguientes procesos:

ID	Tiempo de Llegada	Tiempo de Ejecución
1	0	5
2	1	3
3	2	8
4	3	6

Resuelve el problema utilizando los siguientes algoritmos:

1. FCFS (First-Come, First-Served)
2. SJF (Shortest Job First)
3. Round Robin (Quantum = 2)

Resultados del algoritmo FCFS:

ID	Tiempo de Llegada	Tiempo de Ejecución	Tiempo de Finalización	Tiempo de Espera
1	0	5	5	0
2	1	3	8	4
3	2	8	16	6
4	3	6	22	13