

Sistemas Operativos

Práctico 1

Curso 2025

Objetivos

- Conocer los conceptos básicos de un sistema operativo.
- Conocer la evolución de los sistemas operativos.
- Conocer la estructura de los sistemas de computación.
- Comprender las estructuras y los componentes de un sistema operativo.
- Ejercitar el desarrollo de programas que usen llamadas al sistema.

Duración

- 1 semana.

Ejercicio 1 (básico) Respecto a un sistema operativo:

- ¿Qué es? Incluya sus metas y tareas principales.
- Liste sus componentes principales.
- Liste y describa los cinco servicios principales que debe brindar.
- Mencione tres servicios que debe proveer en lo que concierne a la gestión de procesos.
- Mencione las tres tareas principales en lo que concierne a la gestión de la memoria.
- Mencione las tres actividades principales en lo que concierne a la gestión del sistema de archivos.

Ejercicio 2 (básico) Mencione la principal ventaja de la multiprogramación.

Ejercicio 3 (básico) Defina las propiedades esenciales de los siguientes tipos de sistemas operativos:

- Por lotes.
- De tiempo compartido.
- De tiempo real.
- Paralelos.

Ejercicio 4 (medio)

- Desde el punto de vista de un sistema operativo, describa las diferencias entre multiprocesamiento simétrico y asimétrico.
- En un sistema multiprocesador, ¿Qué ventajas presenta un sistema de multiprocesamiento simétrico ante uno asimétrico?

Ejercicio 5 (básico) ¿Cuándo son útiles los caches? ¿Qué problemas resuelven? ¿Qué problemas causan? Si un cache puede hacerse tan grande como el dispositivo para el cual esta agilizando la transferencia de información (por ejemplo, un cache tan grande como un disco duro) ¿Por qué no hacerlo de ese tamaño y eliminar el disco duro?

Ejercicio 6 (básico) ¿Cuál es el propósito de los manejadores de dispositivos device drivers? ¿Son una componente de hardware o software?

Ejercicio 7 (básico) Describa los mecanismos de protección que brinda el hardware para que el sistema operativo pueda controlar el sistema.

Ejercicio 8 (básico) Explique cómo la distinción entre modo monitor y modo usuario sirve de protección.

Ejercicio 9 (básico) ¿Cuáles de las instrucciones siguientes deberían ser privilegiadas?

- (a) Cambio al modo usuario.
- (b) Cambio al modo monitor.
- (c) Establecer al valor del temporizador del sistema.
- (d) Leer el reloj del sistema.
- (e) Lectura de la memoria protegida por el modo monitor.
- (f) Escritura de la memoria protegida por el modo monitor.
- (g) Desactivar las interrupciones.

Ejercicio 10 (medio) Considere las llamadas al sistema (system calls)

- (a) ¿Quiénes las invocan y para qué sirven?
- (b) (*examen febrero 2024*) Describa:
 - i. las tres formas utilizadas para pasar parámetros cuando se realizan.
 - ii. los pasos que realiza un llamado al sistema al ejecutar, indicando qué componente ejecuta cada paso (proceso de usuario, hardware, o núcleo del sistema operativo).
- (c) Liste y describa las cinco clasificaciones básicas.

Ejercicio 11 (medio) Comente ventajas y desventajas de los distintos enfoques en el diseño del sistema operativo.

Ejercicio 12 (medio) Utilizando llamados al sistema, implementar un programa que lea un archivo (cuyo nombre recibe como parámetro) e imprima su contenido en el dispositivo impresora. Dicho dispositivo de nombre *impresora* cuenta con un buffer de 512 bytes. Cuando el dispositivo tiene el buffer vacío, se lo indica al sistema operativo, el cual genera el evento *impresora_libre*. El sistema operativo ofrece los siguientes llamados al sistema:

- **Fin()**
Indica final exitoso de la ejecución.
- **Abort()**
Indica final anormal de la ejecución.
- **Abrir_archivo(nombre_archivo): Entero** y **Cerrar_archivo(nombre_archivo): Entero**
Abren y cierran un archivo dado su nombre. Devuelven **-1** en caso de error.
- **Leer(nombre_archivo, buffer, n): Entero**
Lee *n* caracteres de un archivo identificado por su nombre, almacenándolos en el array *buffer*, y devuelve la cantidad de bytes leídos, o **-1** en caso de error. En caso de que se haya alcanzado el fin del archivo, devuelve **0**.
- **Obtener_recurso(nombre_recurso)** y **Liberar_recurso(nombre_recurso)**
Obtiene y libera un recurso (que puede ser un dispositivo, memoria, etc.) dado su nombre. Se bloquean hasta que el recurso sea obtenido o liberado.
- **Escribir_dispositivo(nombre_dispositivo, buffer, n): Entero**
Escribe *n* bytes del *buffer* en el dispositivo indicado por el nombre, y devuelve la cantidad de bytes que pudo escribir, o **-1** en caso de error.
- **Esperar_evento(nombre_evento)**
Bloquea al proceso hasta que suceda el evento esperado. Retornan el control inmediatamente si el evento ya ocurre en el momento de invocarlo.

Nota: El sistema es multiprogramado.