



SISTEMAS OPERATIVOS

ACTIVIDAD DE APRENDIZAJE 1

Profesor:
VIOLETA DEL ROCIO BECERRA VELAZQUEZ

VERDUZCO ROSALES LUIS ENRIQUE
223992388
Ingenieria en Computación

24/10/24

CUCEI DIVTIC D04

ÍNDICE

Archivo por lotes.....	2
Explicación de Comandos.....	2
Evolución de los Sistemas Operativos.....	10
Concepto de Sistema Operativo.....	11
Objetivos del Sistema Operativo.....	11
Sistema Operativo de Red (Network Operating Systems - NOS).....	12
Sistema Operativo Distribuido (Distributed Operating Systems - DOS).....	12
Diferencias y Semejanzas: NOS vs. DOS.....	13
Cluster de Computación.....	14
Grid de Computación.....	14
Diferencias y Semejanzas: Cluster vs. Grid.....	15
Modo Usuario y Modo Supervisor.....	16
Servicios del Sistema Operativo.....	16
Sección de preguntas.....	18
Conclusión.....	22
Bibliografía.....	23

TABLA IMAGENES

1. Sistemas Operativos Modernos (Andrew S. Tanenbaum).....	5
2. Operating System Concepts (Silberschatz, Galvin, Gagne).....	6
3. Principles of Operating Systems (Naresh Chauhan).....	7
4. Sistemas Operativos: Una Visión Aplicada (Jesús Carretero).....	8
5. Sistemas operativos, aspectos internos y principios de diseño (WILLIAM STALLINGS)....	9
Gráfico del ciclo Fetch.....	21

Archivo por lotes

```
@echo off
title Ejemplo Comandos
color 0A

echo Hola Mundo.
pause

md MiCarpeta
cd MiCarpeta
echo Hola > saludar.txt
echo Eres Yo? >> saludar.txt
dir
type saludar.txt
pause

ping 8.8.8.8 -n 2
timeout /t 3

del saludar.txt
cd ..
rd MiCarpeta

echo comandos terminados.
```

Explicación de Comandos

- @echo off → Oculta los comandos en ejecución.
- title → Cambia el título de la ventana.
- color 0A → Fondo negro, texto verde.
- echo → Muestra mensajes.
- pause → Pausa hasta que el usuario presione una tecla.
- md → Crea una carpeta.
- cd → Entra a la carpeta.
- > → Crea/sobrescribe un archivo.
- >> → Añade texto al archivo.
- dir → Lista archivos en el directorio.
- type → Muestra contenido de un archivo.
- ping → Prueba conectividad a una IP.
- timeout → Espera X segundos.
- del → Borra un archivo.
- rd → Elimina una carpeta.

Resultados

```
Ejemplo Comandos x + v

Microsoft Windows [Versión 10.0.26100.4946]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\FateL>@echo off
title Ejemplo Comandos
color 0A

echo Hola Mundo
Hola Mundo

pause
Presione una tecla para continuar . . .
```

```
Ejemplo Comandos x + v

color 0A

echo Hola Mundo
Hola Mundo

pause
Presione una tecla para continuar . . .

md MiCarpeta
cd MiCarpeta

echo Hola > saludar.txt
echo Eres yo? >> saludar.txt

dir
El volumen de la unidad C es Acer
El número de serie del volumen es: 06B4-DF47

Directorio de C:\Users\FateL\MiCarpeta

08/17/2025 04:33 PM    <DIR>      .
08/17/2025 04:32 PM    <DIR>      ..
08/17/2025 04:34 PM           18 saludar.txt
                  1 archivos          18 bytes
                  2 dirs   98,653,061,120 bytes libres

type saludar.txt
Hola
Eres yo?
```

Actividad de Aprendizaje 1

```
Ejemplo Comandos x + v

08/17/2025 04:33 PM <DIR> .
08/17/2025 04:32 PM <DIR> ..
08/17/2025 04:34 PM 18 saludar.txt
      1 archivos          18 bytes
      2 dirs  98,653,061,120 bytes libres

type saludar.txt
Hola
Eres yo?

pause
Presione una tecla para continuar . . .

ping 8.8.8.8 -n 2

Haciendo ping a 8.8.8.8 con 32 bytes de datos:
Respuesta desde 8.8.8.8: bytes=32 tiempo=16ms TTL=109
Respuesta desde 8.8.8.8: bytes=32 tiempo=21ms TTL=109

Estadísticas de ping para 8.8.8.8:
  Paquetes: enviados = 2, recibidos = 2, perdidos = 0
    (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
  Mínimo = 16ms, Máximo = 21ms, Media = 18ms

timeout /t 5

Esperando 0 segundos, presione una tecla para continuar . . .
```

```
Ejemplo Comandos x + v

pause
Presione una tecla para continuar . . .

ping 8.8.8.8 -n 2

Haciendo ping a 8.8.8.8 con 32 bytes de datos:
Respuesta desde 8.8.8.8: bytes=32 tiempo=16ms TTL=109
Respuesta desde 8.8.8.8: bytes=32 tiempo=21ms TTL=109

Estadísticas de ping para 8.8.8.8:
  Paquetes: enviados = 2, recibidos = 2, perdidos = 0
    (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
  Mínimo = 16ms, Máximo = 21ms, Media = 18ms

timeout /t 5

Esperando 0 segundos, presione una tecla para continuar . . .

del saludar.txt
cd ..
rd MiCarpeta

echo comandos terminados
comandos terminados

pause
Presione una tecla para continuar . . .
```

Libros sobre Sistemas Operativos

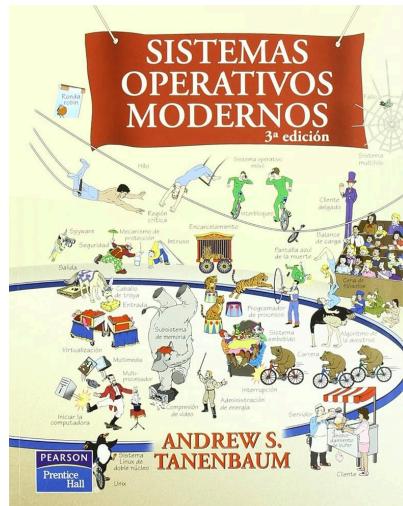
El conocimiento de los sistemas operativos es fundamental en la ingeniería de software. Es lo que distingue a los desarrolladores capaces de escribir código de quienes realmente comprenden cómo se ejecuta. Ya sea que se trate de optimizar el rendimiento o depurar problemas complejos, este conocimiento resulta invaluable.

Para comprender su diseño y funcionamiento, se han escrito varios libros que abordan desde conceptos básicos hasta implementaciones avanzadas.

5 libros sobre el tema.

1. Sistemas Operativos Modernos (Andrew S. Tanenbaum)

- Portada:



- Autor: Andrew S. Tanenbaum, experto en SO y redes.

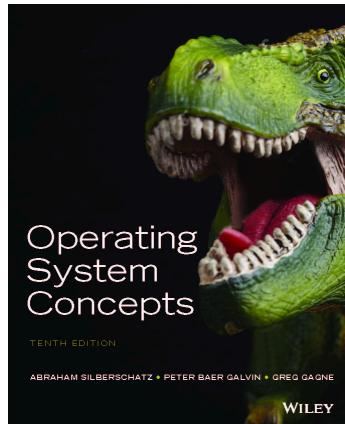
- Contenido:

- Introducción
- Procesos e Hilos
- Administración de memoria
- Sistemas de archivos
- Entrada/Salida
- Interbloqueos
- Sistemas operativos multimedia
- Sistemas de múltiples procesadores
- Seguridad
- Linux
- Windows Vista
- Symbian Os
- Diseño de sistemas operativos

Actividad de Aprendizaje 1

2. Operating System Concepts (Silberschatz, Galvin, Gagne)

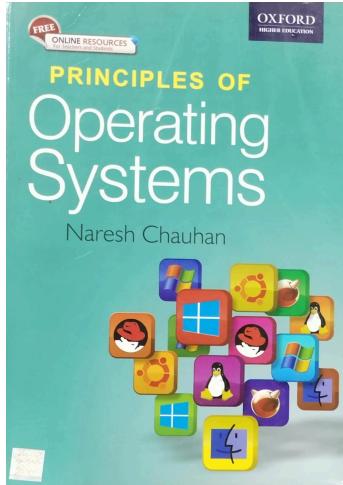
- Portada:



- Autores: ABRAHAM SILBERSCHATZ, PETER BAER GALVIN, GREG GAGNE
- Contenido:
 - OVERVIEW
 - PROCESS MANAGEMENT
 - PROCESS SYNCHRONIZATION
 - MEMORY MANAGEMENT
 - INPUT OUTPUT MANAGEMENT
 - FILE SYSTEM
 - SECURITY AND PROTECTION
 - ADVANCED TOPICS
 - LINUX
 - WINDOWS
 - APPENDICES

3. Principles of Operating Systems (Naresh Chauhan)

- Portada:



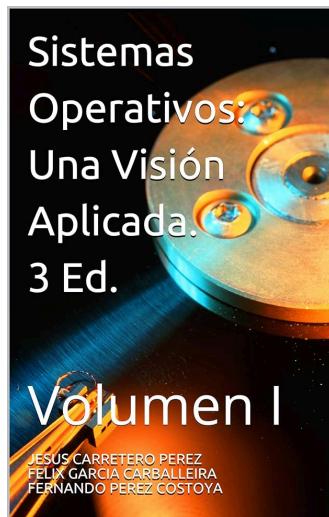
- Autor: Naresh Chauhan, especialista en sistemas embebidos.

- Contenido:

- Introduction
- Process Management
- Memory Management
- File Management
- Input–Output Management
- Security and Protection
- Advanced Operating Systems
- Shell Programming

4. Sistemas Operativos: Una Visión Aplicada (Jesús Carretero)

- Portada:



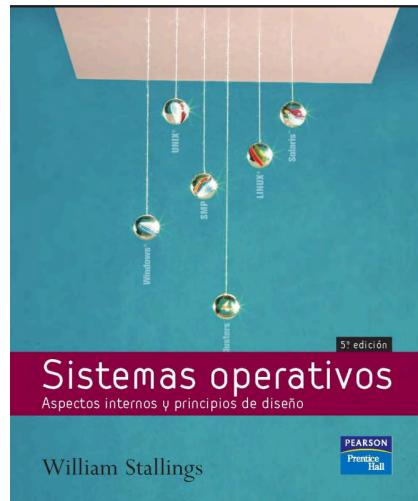
- Autor: Jesús Carretero, catedrático español.

- Contenido:

- Conceptos arquitectónicos del computador
- Introducción a los sistemas operativos
- Procesos
- Gestión de memoria
- Entrada y salida
- E/S y Sistema de ficheros
- Comunicación y sincronización de procesos
- Apéndices

5. Sistemas operativos, aspectos internos y principios de diseño (WILLIAM STALLINGS)

- Portada:



- Autor: WILLIAM STALLINGS

- Contenido:

- Antecedentes
- Procesos
- Memoria
- Entrada/Salida
- Sistemas distribuidos y seguridad
- Apéndices

Estos libros son los que me parecieron más esenciales para entender los sistemas operativos desde diferentes perspectivas:

- Teórica: Tanenbaum y Silberschatz.
- Práctica: Carretero y Chauhan.
- Actualizada: Gagne.

Evolución de los Sistemas Operativos

Procesamiento en Serie

Década de 1940-1950 (Primera Generación):

La informática tal y como se le conoce hoy día, surgió a raíz de la II Guerra Mundial, en la década de los 40. En esos años no existía siquiera el concepto de "Sistema Operativo" y los programadores interactuaban directamente con el hardware de las computadoras trabajando en lenguaje máquina (esto es, en binario, programando únicamente con 0s y 1s).

Las computadoras ejecutaban una tarea a la vez.

El operador cargaba manualmente programas (cintas o tarjetas perforadas).

Ineficiente por el tiempo de configuración entre trabajos.

Procesamiento por Lotes (Batch, Segunda generación)

Década de 1950-1960:

El concepto de Sistema Operativo surge en la década de los 50. El primer Sistema Operativo de la historia fue creado en 1956 para un ordenador IBM 704, y básicamente lo único que hacía era comenzar la ejecución de un programa cuando el anterior terminaba.

En los años 60 se produce una revolución en el campo de los Sistemas Operativos. Aparecen conceptos como sistema multitarea, sistema multiusuario, sistema multiprocesadores y sistema en tiempo real.

Es en esta década cuando aparece UNIX, la base de la gran mayoría de los Sistemas Operativos que existen hoy en día.

Se agrupaban trabajos similares en "lotes" para ejecución secuencial.

Primeros SO como GM-NAA I/O (IBM) automatizaban la transición entre trabajos.

Redujo tiempos muertos pero carecía de interacción en tiempo real.

Decadas de 1970 a 2000 (Tercera Generación)

La electrónica avanza hacia la integración a gran escala, se crean circuitos con miles de transistores en un centímetro cuadrado de silicón y ya es posible hablar de las computadoras personales y las estaciones de trabajo. Surgen los conceptos de interfaces amigables intentando así atraer al público en general al uso de las computadoras como herramientas cotidianas.

En los años 70 se produce un boom en cuestión de ordenadores personales, acercando estos al público general de manera impensable hasta entonces. Esto hace que se multiplique el desarrollo, creándose el lenguaje de programación C (diseñado específicamente para reescribir por completo el código UNIX).

Como consecuencia de este crecimiento exponencial de usuarios, la gran mayoría de ellos sin ningún conocimiento sobre lenguajes de bajo o alto nivel, hizo que en los años 80, la prioridad a la hora de diseñar un sistema operativo fuese la facilidad de uso, surgiendo así las primeras interfaces de usuario.

En los 80 nacieron sistemas como MacOS, MS-DOS, Windows.

Multiprogramación (1960s): Múltiples programas en memoria.

Time-Sharing (1970s): Tiempo compartido para usuarios interactivos.

SO Modernos (1980s-act.): GUI, multitarea, redes, móviles.

Año 2000 en adelante (Cuarta Generación)

En la década de los 90 hace su aparición Linux, publicándose la primera versión del núcleo en septiembre de 1991, que posteriormente se uniría al proyecto GNU, un sistema operativo completamente libre, similar a UNIX, al que le faltaba para funcionar un núcleo funcional. Hoy en día la mayoría de la gente conoce por Linux al Sistema Operativo que realmente se llama GNU/Linux y así los niveles de interacción humano máquina se van haciendo cada vez más profundos.

Concepto de Sistema Operativo

Es un software que gestiona hardware, provee interfaz usuario-a-máquina y coordina recursos (CPU, memoria, E/S), su propósito es proporcionar un entorno en el cual el usuario pueda ejecutar programas.

Un Sistema Operativo es una parte importante de cualquier sistema de computación. Un sistema de computación puede dividirse en cuatro componentes:

El hardware, el Sistema Operativo, los programas de aplicación y los usuarios.

El hardware (Unidad Central de Procesamiento(UCP), memoria y dispositivos de entrada/salida (E/S)) proporciona los recursos de computación básicos.

Objetivos del Sistema Operativo

El objetivo principal de un sistema operativo es el lograr que el sistema de computación se use de manera cómoda, y el objetivo secundario es que el hardware del computador se emplee de manera eficiente.

Sistema Operativo de Red (Network Operating Systems - NOS)

Un Sistema Operativo de Red es un software que se ejecuta en un servidor y permite gestionar, controlar y proveer recursos (como archivos, impresoras, aplicaciones y dispositivos) a otras computadoras, llamadas clientes, que están conectadas a través de una red (LAN - Red de Área Local).

Cada máquina en la red mantiene su propio sistema operativo independiente y su autonomía. El servidor solo actúa como un proveedor de servicios centralizado.

La función del NOS es entonces proporcionar servicios y características básicas de red que soporten múltiples solicitudes de entrada simultáneamente en un entorno multiusuario.

Debido a que las versiones anteriores de los sistemas operativos básicos no estaban diseñadas para su uso en red, los sistemas operativos en red surgieron como una solución para las computadoras de un solo usuario.

Cómo funciona:

Un usuario en una estación de trabajo cliente debe iniciar sesión explícitamente en el servidor para acceder a sus recursos. Es como tener una llave (credenciales) para entrar a una bodega (el servidor) y tomar una caja (un archivo). La computadora cliente es perfectamente funcional por sí sola, sin el servidor.

Ejemplos:

- Windows Server (y sus clientes Windows 10/11)
- Novell NetWare (era el rey en los 90s)
- Linux con servicios como Samba (para compartir archivos con Windows), NFS, o Apache.

Sistema Operativo Distribuido (Distributed Operating Systems - DOS)

Un Sistema Operativo Distribuido es un software que gestiona un conjunto de computadoras independientes y conectadas por red, haciéndolas aparecer ante el usuario como una sola computadora poderosa y cohesiva. Es una abstracción mucho más compleja. Aquí no hay "clientes" y "servidores" en el sentido tradicional; hay una colección de nodos que colaboran transparentemente.

Se construye a partir de una cierta arquitectura un sistema potente que permita una computación eficiente, escalable y con tolerancia a fallos en más de una máquina. Su objetivo es brindar a los usuarios todas las ventajas de compartir recursos, mayor

tolerancia a fallos y escalabilidad, a la vez que les ofrece la ventaja de una interacción transparente con el sistema.

Cómo funciona:

El sistema operativo está instalado en todos los nodos y se encarga de distribuir automáticamente las tareas (procesos), los archivos y los recursos entre todas las máquinas. El usuario no sabe ni necesita saber en qué máquina física se está ejecutando su programa o almacenando sus datos.

Diferencias y Semejanzas: NOS vs. DOS

Característica	Sistema Operativo de Red (NOS)	Sistema Operativo Distribuido (DOS)
Acoplamiento	Débil. Las máquinas son independientes.	Fuerte. Las máquinas dependen del conjunto para funcionar.
Autonomía	Alta. Cada nodo tiene su propio SO y puede trabajar offline.	Baja o nula. Los nodos están diseñados para trabajar en conjunto.
Transparencia	Baja. El usuario debe conocer la ubicación de los recursos.	Alta (El objetivo principal). El sistema oculta la distribución al usuario.
Escalabilidad	Generalmente buena. Es fácil añadir nuevos clientes o servidores.	Compleja. Añadir un nuevo nodo requiere una integración profunda.
Comunicación	Basada en protocolos de red (TCP/IP, IPX/SPX) y servicios (FTP, HTTP).	Basada en paso de mensajes y memoria compartida distribuida.
Tolerancia a Fallos	Limitada. Si el servidor se cae, los servicios se caen. Los clientes siguen funcionando.	Alta (Objetivo clave). El sistema puede redirigir tareas a nodos sanos.
Semejanza Principal	Ambos permiten compartir recursos hardware y software a través de una red.	Ambos permiten compartir recursos hardware y software a través de una red.

Cluster de Computación

Un cluster es un conjunto de computadoras similares o idénticas (nodos) estrechamente acopladas, conectadas a través de una red de alta velocidad (como InfiniBand), y que trabajan juntas para ser vistas como un solo sistema.

El objetivo principal es combinar el poder de cómputo para lograr un alto rendimiento (HPC), alta disponibilidad, o balanceo de carga.

Con esto, las computadoras comienzan a actuar dentro de un solo sistema, trabajando juntas en el procesamiento, análisis e interpretación de datos e información, y/o realizando tareas simultáneas.

Cómo funciona:

Un software especial (middleware de clustering, como Kubernetes o Slurm) se encarga de presentar el cluster como una única unidad y de distribuir las tareas entre los nodos. Los nodos suelen estar en la misma ubicación física y están diseñados específicamente para trabajar en conjunto.

Tipos comunes:

- Cluster de Alto Rendimiento (HPC):
Para problemas científicos que requieren mucho cálculo (ej.: simulación del clima).
- Cluster de Alta Disponibilidad (HA):
Para servicios críticos que nunca deben fallar (ej.: un servidor de bases de datos). Si un nodo falla, otro toma el control inmediatamente.

Grid de Computación

Un grid es una infraestructura que coordina recursos heterogéneos y geográficamente distribuidos que pertenecen a diferentes dominios administrativos (diferentes organizaciones o instituciones).

El foco no es verse como un solo sistema, sino en estandarizar el acceso a recursos distribuidos para resolver problemas masivos.

Cómo funciona:

Las computadoras o servidores en una red de computación grid se denominan nodos. Cada nodo ofrece recursos de computación no utilizados, como CPU,

memoria y almacenamiento a la red de grid. Al mismo tiempo, también puede usar los nodos para realizar otras tareas no relacionadas. No hay límite para el número de nodos en la computación grid. Hay tres tipos principales de nodos: de control, proveedor y nodos de usuario.

Se basa en protocolos y middleware (como Globus Toolkit) que establecen estándares para compartir recursos de forma segura. Un usuario puede enviar un trabajo que use CPU de una universidad en España, almacenamiento de una en México, y visualización de una en EE.UU., todo de forma coordinada pero sin un control centralizado absoluto.

Diferencias y Semejanzas: Cluster vs. Grid

Característica	Cluster	Grid
Acoplamiento	Fuerte y estrecho. Nodos homogéneos y cercanos.	Débil y amplio. Recursos heterogéneos y distribuidos.
Ubicación	Centralizada. Todos los nodos suelen estar en el mismo lugar.	Distribuida Geográficamente. Los recursos están en todo el mundo.
Administración	Centralizada. Un único ente controla todo el cluster.	Descentralizada. Cada organización administra sus propios recursos.
Homogeneidad	Alta. Hardware, SO y red suelen ser similares.	Baja. Hardware, SO y redes son diversos.
Objetivo Principal	Alto Rendimiento o Alta Disponibilidad.	Compartir recursos a gran escala para resolver problemas masivos.
Interconexión	Red de Alta Velocidad y Baja Latencia.	Internet y redes de área amplia.
Semejanza Principal	Ambos son tipos de Sistemas Distribuidos que buscan unir recursos de cómputo para un fin común.	Ambos son tipos de Sistemas Distribuidos que buscan unir recursos de cómputo para un fin común.

Modo Usuario y Modo Supervisor

El modo sin restricciones se denomina modo supervisor o modo kernel y el modo restringido se denomina modo usuario.

- El sistema operativo se ejecuta en modo kernel o modo supervisor y está protegido de la manipulación del usuario por el hardware, mientras que el compilador y los editores se ejecutan en modo usuario.
- Si un usuario desea escribir un nuevo compilador y reemplazar el proporcionado, puede hacerlo, pero no es libre de escribir su propio manejador de interrupciones de reloj, que es parte del sistema operativo y normalmente está protegido por hardware contra los intentos de los usuarios de modificarlo.
- Los modos de usuario no permiten operaciones como escrituras en memoria aleatoria, para proteger programas entre sí, mientras que el modo supervisor permite operaciones como las que el sistema operativo necesita.

<u>Modo Usuario</u>	<u>Modo Supervisor (Kernel)</u>
Apps sin acceso directo a hardware.	Acceso total a hardware.
Llamadas al sistema para operaciones privilegiadas.	Ejecuta instrucciones críticas.
Protege al SO de errores en apps.	Gestiona interrupciones y memoria.

Servicios del Sistema Operativo

- **Servicios que garantizan el funcionamiento eficiente del sistema**
 - Asignación de recursos:
Cuando hay múltiples usuarios o múltiples trabajos ejecutándose los recursos deben ser asignados a cada uno de ellos.
Esta asignación debe hacerse con el fin de garantizar la máxima eficacia del sistema.

- Monitorización:
Es normal querer hacer seguimiento de los recursos que los usuarios usan y en qué cantidad. Esto puede ser útil para facturar a los usuarios por el uso de los recursos, para configurar el sistema mejorando el rendimiento o para limitar cuánto de cada recurso puede usar cada usuario como máximo.
- Protección y seguridad:
Protección implica asegurar que el acceso a los recursos del sistema está controlado, que la información almacenada en un sistema multiusuario solo pueda ser accedida por su propietario o que un proceso no pueda interferir con otro o con el sistema operativo.
- Agentes exteriores:
La seguridad del sistema respecto a los agentes exteriores también es importante. Empieza obligando a los usuarios a autenticarse en él para obtener acceso a los recursos del mismo, pero incluye defender de intentos de acceso inválidos a través de la red.

- **Servicios útiles para el usuario**

- Interfaz de usuario:
Los sistemas operativos diseñados para que los usuarios interactúen con ellos deben proporcionar una interfaz de usuario adecuada, que puede tener diferentes formas según el propósito del sistema.
- Operaciones de E/S:
Un programa puede necesitar realizar operaciones de E/S que pueden involucrar a archivos o a dispositivos de E/S. Por eficiencia y protección un usuario, normalmente los procesos no puede tener acceso directo a los dispositivos; por lo que el sistema operativo debe proporcionar medios para solicitar estas operaciones a los componentes correspondientes del sistema operativo.
- Manipulación de sistemas de archivos:
Los programas necesitan leer y escribir archivos y directorios, crearlos y borrarlos por nombre, buscar un archivo dado y listar información acerca del mismo.

Actividad de Aprendizaje 1

- Comunicaciones:
Los procesos necesitan poder intercambiar información entre ellos, tanto si se ejecutan en el mismo ordenador, como en diferentes equipos unidos por una red.
- Detección de errores:
El sistema operativo necesita tener conocimiento de los posibles errores y para cada tipo de error debe tomar la acción apropiada para asegurar una computación consistente y segura, ya que puede haber errores del hardware como fallos de energía o errores en la memoria, también errores de paridad o en los programas del usuario.

Resumiendo...

- Gestión de Procesos: Creación, planificación y terminación.
- Gestión de Memoria: Asignación, paginación, swaps.
- Sistema de Archivos: Organización y acceso a datos.
- Control de Dispositivos: Drivers para E/S.
- Seguridad: Autenticación y permisos.

Sección de preguntas

1. De los libros que busco liste los temas que tienen en común.

- Antecedentes
- Procesos
- Memoria
- Entrada/Salida
- Seguridad y protección
-

2. ¿Qué es un archivo por lotes?

Es un archivo que contiene una serie de instrucciones (código) secuenciales que la computadora entiende y ejecuta de manera estructurada.

3. Explique con sus palabras el Procesamiento en Serie.

Es como una serie de pasos que se deben realizar de manera secuencial empezando por el primero y terminando en el último de tal manera que no se pueden saltar pasos y todos deben ser ejecutados uno por uno en orden.

4. Definición de Sistema Operativo.

Se encarga de lograr una mayor interacción humano-máquina al unir hardware y software, ya que el sistema debe controlar todas las respuestas de la computadora para darnos una salida visual del trabajo que estamos realizando.

5. Liste cada uno de los Gestores del Sistema Operativo, así como su función principal.

1. Gestor de Procesos:

Es el que se ocupa de todo el ciclo de vida de los programas en ejecución (procesos). Su trabajo es repartir el tiempo de la CPU entre todos ellos, decidir cuál se ejecuta y cuándo, y asignarles los recursos que necesitan para que podamos hacer varias cosas a la vez sin que el sistema se trabe.

2. Gestor de Memoria:

Este se encarga de administrar la RAM. Se asegura de que cada proceso tenga su propio espacio en memoria para trabajar, de que no se metan con la memoria de otros (para evitar conflictos) y de optimizar el uso de esta para que no se desperdicie.

3. Gestor de Entrada/Salida (E/S):

Actúa como un intermediario o traductor entre el software (el SO y las apps) y el hardware (como el disco duro, el teclado, la impresora). Controla los drivers de los dispositivos para que nosotros podamos interactuar con ellos de manera sencilla sin tener que saber cómo funcionan por dentro.

4. Gestor de Archivos:

Es el responsable de cómo se guardan, ordenan y acceden los archivos en los discos (en carpetas/directorios). Gracias a él podemos crear, buscar, leer, borrar y organizar nuestros documentos y programas.

5. Gestor de Seguridad:

Su función es proteger la información y los recursos del acceso de personas o programas que no deberían entrar. Usa contraseñas, permisos y cifrado para autenticar usuarios y controlar lo que cada uno puede hacer o ver.

6. Escriba los objetivos de un sistema operativo.

- Eficiencia: Optimizar el uso de recursos.
- Comodidad: Facilitar la interacción humano-máquina.
- Evolución: Adaptarse al nuevo hardware sin cambios graves en las apps.
- Seguridad: Proteger los datos y procesos.

7. Ilustre y explique en qué consiste el ciclo Fetch.

El CPU es el componente de la computadora que se encarga de ejecutar los programas. Para hacerlo, sigue un proceso fundamental conocido como el ciclo Fetch-Decodify-Execute (Traer, Decodificar, Ejecutar), que es la base de todas las operaciones que puede realizar una máquina.

Aunque este ciclo ha evolucionado y tiene sus variantes con los avances tecnológicos, las etapas esenciales son:

- Fetch (Traer la instrucción):

El CPU busca y obtiene la siguiente instrucción a ejecutar desde la memoria principal (RAM) y la guarda en un registro interno para poder trabajarla.

- Decode (Decodificar la instrucción):

El CPU "interpreta" la instrucción que acaba de traer. Identifica qué operación debe hacer (como sumar o guardar un dato) y localiza en dónde está la información necesaria para realizarla (si en la memoria, en un registro interno o en la misma instrucción).

- Cargar operandos:

Antes de ejecutar, el CPU recoge y prepara todos los datos (operandos) que necesita la instrucción, cargándolos en los lugares adecuados para que estén listos para ser usados.

- Execute (Ejecutar):

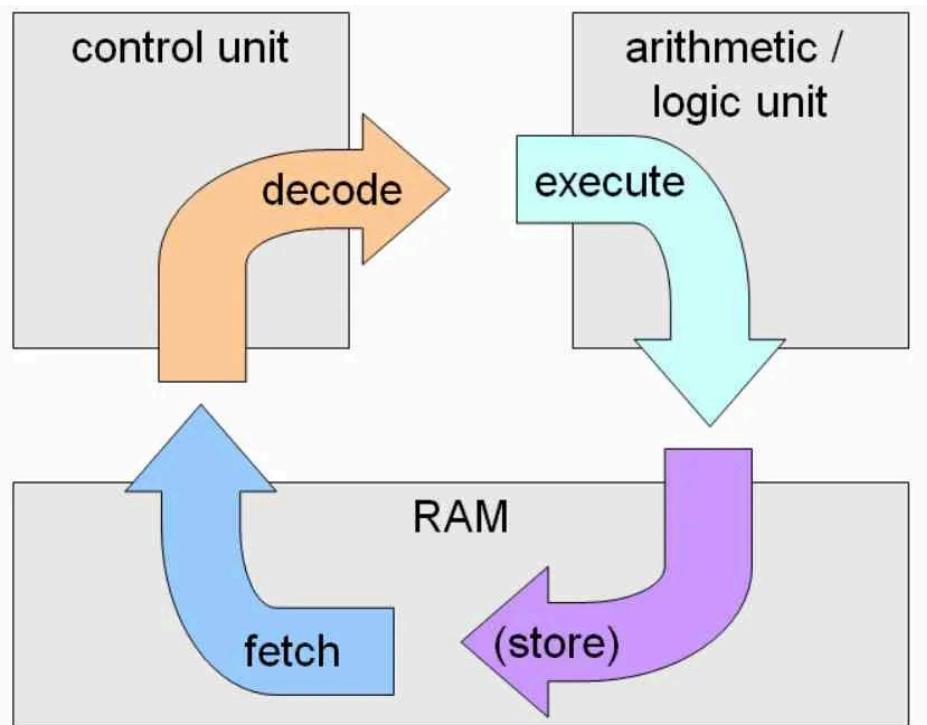
Es el momento de hacer la operación. El CPU lleva a cabo la acción que se le indicó, como realizar un cálculo matemático, mover datos o comparar valores.

- Almacenar el resultado:

Una vez hecha la operación, el CPU guarda el resultado obtenido en el lugar que corresponda, que puede ser un registro interno o la memoria principal.

- Actualizar el Contador de Programa (PC):

Por último, el CPU actualiza el registro PC (Program Counter), que es como un bookmark que siempre apunta a la siguiente instrucción que toca ejecutar, para que el ciclo pueda continuar sin problemas.

Gráfico del ciclo Fetch**8. ¿Cómo podrían clasificarse los diferentes sistemas operativos?**

Los sistemas operativos se pueden clasificar en varias categorías según sus capacidades:

1. Por el número de usuarios:

- **Monousuario:**

Solo permite que un solo usuario use la computadora a la vez. No hay distinción entre usuarios, por lo que toda la información del sistema está disponible para quien tenga acceso físico a la máquina. Ejemplo: MS-DOS.

- **Multiusuario:**

Permite que varios usuarios utilicen el sistema al mismo tiempo, ya sea desde terminales conectadas directamente o de forma remota.

Actividad de Aprendizaje 1

2. Por el número de tareas:

- Monotarea:

Solo pueden ejecutar un programa o proceso a la vez. Aunque el sistema pueda ser multiusuario (varias personas conectadas), cada usuario solo puede hacer una cosa a la vez.

- Multitarea:

Permiten que un usuario ejecute varios programas o procesos de manera simultánea (como imprimir un documento mientras se edita otro). Esto crea la ilusión de que todo funciona al mismo tiempo, lo que hace que el usuario sea mucho más productivo.

3. Por el manejo de procesadores:

- Uniproceso:

Solo son capaces de gestionar y utilizar una única CPU o núcleo de procesador. Si la computadora tiene más de uno, no podrían aprovecharlos.

- Multiproceso (o multiprocesador):

Están diseñados para funcionar en computadoras con más de un CPU. Esto permite distribuir la carga de trabajo entre todos los procesadores, mejorando enormemente el rendimiento. Estos sistemas se dividen en:

- Simétricos:

Todos los procesadores son iguales y las tareas se reparten por igual entre ellos.

- Asimétricos:

Un procesador actúa como el "jefe" (maestro), controlando y asignando el trabajo específico a los demás procesadores (esclavos).

Conclusión

En resumen, los sistemas operativos son extremadamente diversos y se pueden clasificar según tres criterios principales: por el número de usuarios (monousuario vs. multiusuario), por el número de tareas (monotarea vs. multitarea) y por su manejo del hardware (uniproceso vs. multiproceso). Entender estas categorías es clave para saber cómo un SO gestiona los recursos, interactúa con el usuario y se adapta a las necesidades específicas de cada entorno computacional, desde una computadora personal hasta un gran servidor. Básicamente, la evolución de los sistemas operativos ha ido dirigida a aumentar la eficiencia, la productividad y el aprovechamiento del hardware.

Bibliografía

Upadhyay, A. (2025, 18 de febrero). *My Top 15 OS Books: From Theory and Implementation to Systems Programming. Confessions of a Code Addict.* Recuperado de <https://blog.codingconfessions.com/p/my-top-15-os-books>

Arturo Feria Gerónimo (s. f.). *Historia de los sistemas operativos.* Monografías.com. Recuperado de <https://www.monografias.com/trabajos12/hisis/hisis>

Facultad de Informática de Barcelona, Universidad Politécnica de Cataluña (s. f.). *Historia de los sistemas operativos. Retro-informática.* Recuperado de <https://www.fib.upc.edu/retro-informatica/historia/so.html>

Universidad de Valladolid (2006, 4 de octubre). *SO_TE01_20061004* [PDF]. Universidad de Valladolid. Recuperado de https://www.infor.uva.es/~fdiaz/so/doc/SO_TE01_20061004.pdf

Jesús María Aransay Azofra (2011-2012). *Tema 3* [PDF]. Universidad de La Rioja. Recuperado de https://www.unirioja.es/cu/jearansa/1112/ficheros/Tema_3.pdf

Torres, J. (2023–2024). *Servicios del sistema.* En *Sistemas Operativos* (Curso 2023-2024). Universidad de La Laguna. Recuperado de https://ull-esit-sistemas-operativos.github.io/ssoo-apuntes/so2324/servicios_del_sistema.html

Alejandro Teófilo Pacheco Lliuya (2022). *Documento PDF* [PDF]. Repositorio UNE. Recuperado de <https://repositorio.une.edu.pe/server/api/core/bitstreams/dbd52176-5712-469d-a2c1-5abc45c5da8b/content>

ComputerWeekly.es. (s.f.). Sistema operativo de red (NOS). Recuperado de <https://www.computerweekly.com/es/definicion/Sistema-operativo-de-red-NOS>

DriveNets. (s.f.). What is a Network Operating System (NOS)? Recuperado de <https://drivenets.com/resources/education-center/what-is-a-network-operating-system-nos/>

Talent500. (2023, diciembre 19). Distributed Operating System: A Complete Guide. Recuperado de <https://talent500.com/blog/distributed-operating-system-guide/>

Actividad de Aprendizaje 1

Zendesk. (2023, junio 22). ¿Qué es un cluster? Tipos y funcionamiento.
Recuperado de <https://www.zendesk.com.mx/blog/cluster-que-es/>

Amazon Web Services, Inc. (s.f.). ¿Qué es la computación en grid? - Computación en grid explicada.

Recuperado de <https://aws.amazon.com/es/what-is/grid-computing/>