

Sistemas operativos y procesos

INNOVA PHYSICS

Índice:

1. ¿Qué es un sistema operativo?
 2. Arquitectura general de un SO
 3. Gestión avanzada de procesos
 4. Hilos y concurrencia
 5. Interfaz Gráfica (GUI) vs Línea de Comandos (CLI)
 6. Permisos, usuarios y control de acceso
 7. Gestión de memoria
 8. CRUD en archivos (Operaciones fundamentales)
 9. Sistemas de archivos
 10. Comandos comunes por sistema operativo
-

1. ¿Qué es un sistema operativo?

- Es un **software** fundamental que actúa como **intermediario** entre el **hardware** de un dispositivo (como una computadora o teléfono móvil) y las **aplicaciones** que utilizas.
- Funciones principales:
 - Abstracción de hardware (dispositivos, memoria, CPU)
 - Gestión de recursos compartidos
 - Coordinación de tareas y procesos
 - Control de acceso y seguridad
 - Aislamiento y concurrencia



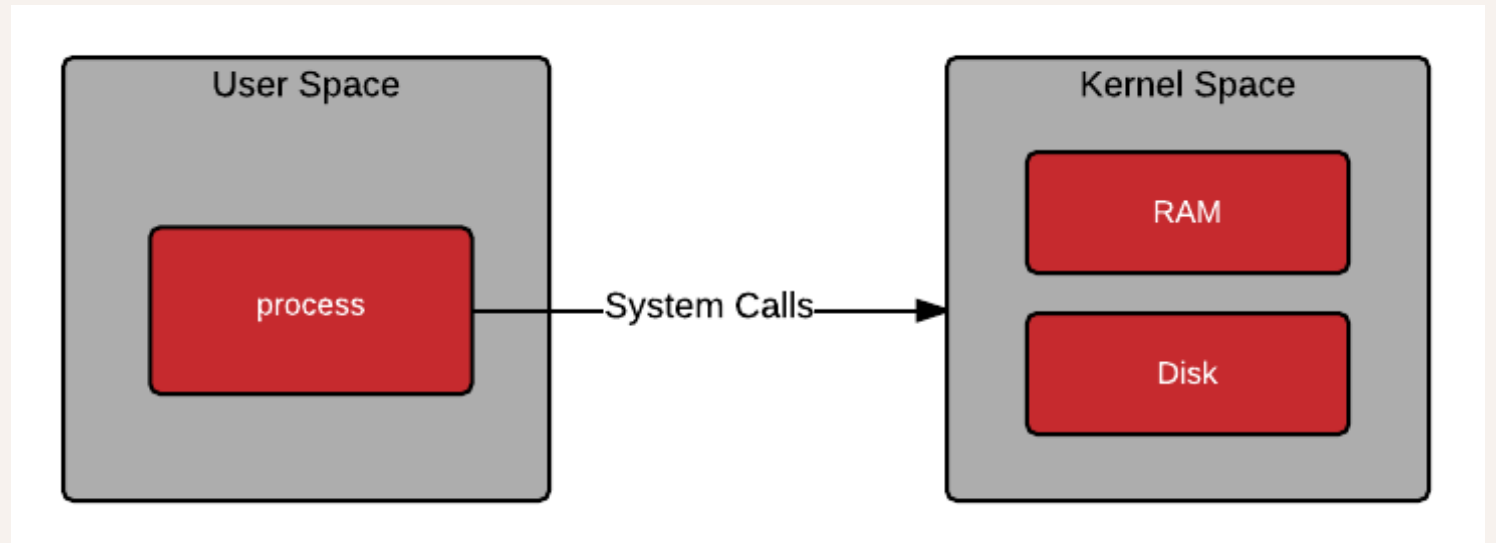
2. Arquitectura general de un SO

Kernel space: núcleo monolítico o microkernel, maneja interrupciones, planificación, sincronización, drivers.

User space: procesos de usuario, bibliotecas dinámicas, demonios/ sistemas de servicios.

System calls: punto de entrada para que las aplicaciones soliciten servicios al kernel.

Modelos de diseño: monolítico (Linux), microkernel (Minix, QNX), híbrido (Windows NT, XNU de macOS) y exokernel (experimental, mínima abstracción)



3. Gestión avanzada de procesos

Un **proceso** es una unidad de ejecución que incluye código, estado, recursos asignados y contexto de ejecución.

Estados clásicos del ciclo de vida:

- New, Ready, Running, Waiting, Terminated
- Transiciones administradas por el planificador de procesos

Tablas de procesos:

- PCB (Process Control Block): almacena PID, registros, puntero de pila, estado de planificación, etc.

Mecanismos:

- Fork/Exec (Unix) o CreateProcess (Windows)
 - Planificadores: FIFO, Round-Robin, Multilevel Feedback Queue
-

4. Hilos y concurrencia

Los **threads** (hilos) son unidades ligeras de ejecución dentro de un mismo proceso.

Comparten código, memoria, y recursos del proceso padre.

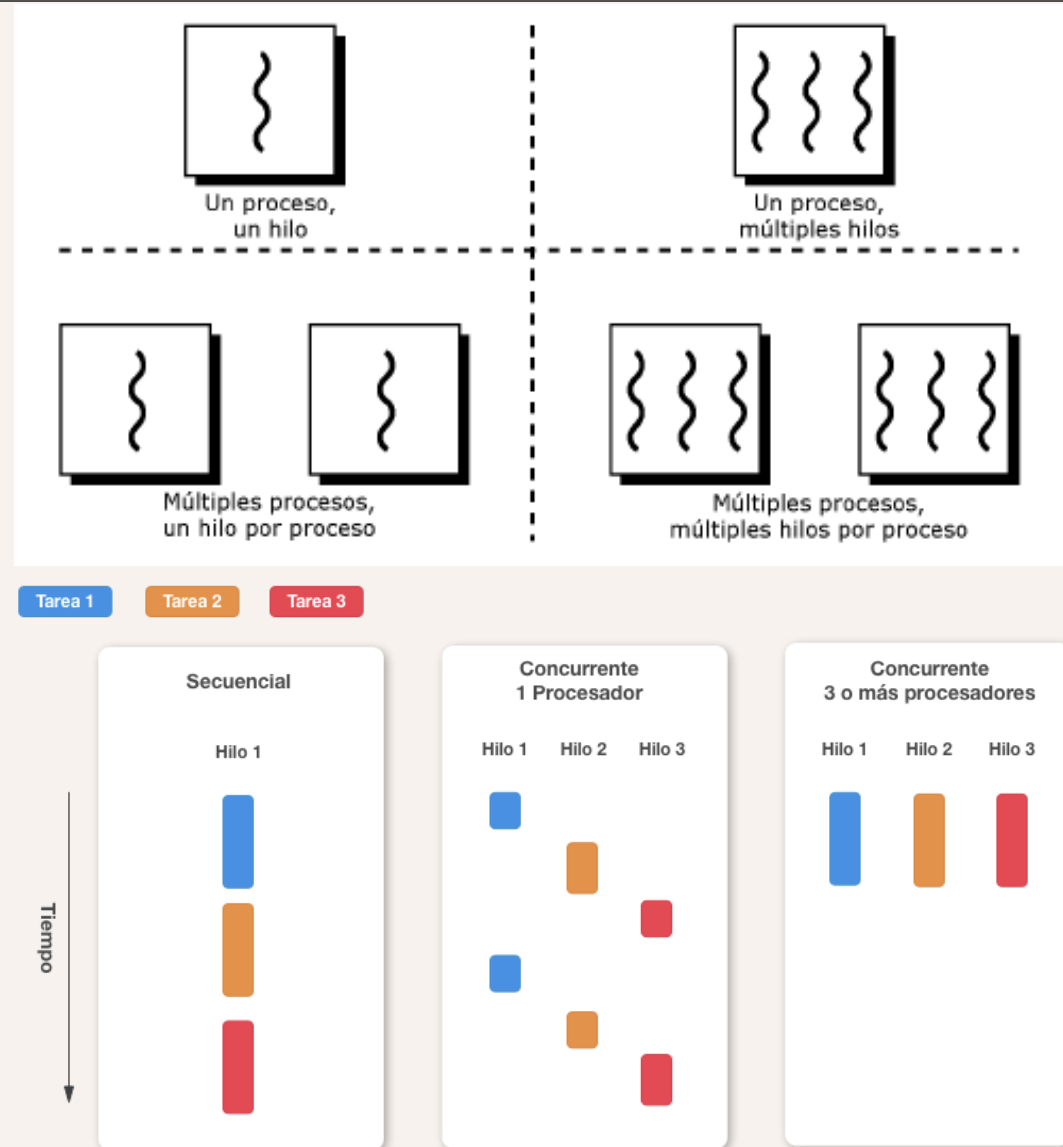
Permiten mejorar el rendimiento en tareas paralelizables (multithreading).

Modelos de concurrencia:

User-level vs Kernel-level threads

SMP (Symmetric Multiprocessing)

Mecanismos de sincronización: semáforos, mutexes, monitores, spinlocks



5. Interfaz Gráfica (GUI) vs Línea de Comandos (CLI)

GUI (Graphical User Interface)

- Basada en elementos visuales: ventanas, iconos, botones y menús.
- Intuitiva para usuarios finales.
- Ideal para tareas visuales o simples (como edición de imágenes, navegación por carpetas).
- Ejemplos: Explorador de Windows, Finder (macOS), Nautilus (Linux).

CLI (Command Line Interface)

- Interfaz basada en texto: el usuario interactúa mediante comandos escritos.
 - Ofrece mayor control, velocidad y automatización.
 - Permite scripting, ejecución remota, gestión de recursos avanzados.
 - Esencial en administración de sistemas, desarrollo, ciberseguridad y DevOps.
-

6. Permisos, usuarios y control de acceso

```
total 72
drwxr-xr-x 20 root root 4096 Apr 21 04:32 .
drwxr-xr-x 20 root root 4096 Apr 21 04:32 ..
lrwxrwxrwx 1 root root 7 Jul 28 2021 bin -> usr/bin
drwxr-xr-x 4 root root 4096 Jul 28 2021 boot
drwxr-xr-x 18 root root 3980 Apr 21 04:32 dev
drwxr-xr-x 95 root root 4096 Apr 21 04:46 etc
drwxr-xr-x 4 root root 4096 Apr 21 04:37 home
lrwxrwxrwx 1 root root 7 Jul 28 2021 lib -> usr/lib
lrwxrwxrwx 1 root root 9 Jul 28 2021 lib32 -> usr/lib32
lrwxrwxrwx 1 root root 9 Jul 28 2021 lib64 -> usr/lib64
lrwxrwxrwx 1 root root 10 Jul 28 2021 libx32 -> usr/libx32
drwx----- 2 root root 16384 Jul 28 2021 lost+found
drwxr-xr-x 3 root root 4096 Jul 28 2021 media
drwxr-xr-x 2 root root 4096 Jul 31 2020 mnt
drwxr-xr-x 3 root root 4096 Jul 28 2021 opt
dr-xr-xr-x 158 root root 0 Apr 21 04:32 proc
drwx----- 3 root root 4096 Apr 21 04:34 root
drwxr-xr-x 27 root root 900 Apr 21 04:32 run
lrwxrwxrwx 1 root root 8 Jul 28 2021 sbin -> usr/sbin
drwxr-xr-x 2 root root 4096 Jul 28 2021 snap
drwxr-xr-x 2 root root 4096 Jul 31 2020 srv
dr-xr-xr-x 13 root root 0 Apr 21 04:32 sys
drwxrwxrwt 10 root root 4096 Apr 21 04:50 tmp
drwxr-xr-x 15 root root 4096 Jul 28 2021 usr
drwxrwxrwx 1 vagrant vagrant 0 Mar 5 10:20 vagrant
drwxr-xr-x 13 root root 4096 Jul 28 2021 var
```

Permisos, usuarios y control de acceso: los sistemas operativos modernos implementan modelos de permisos para proteger archivos, procesos y recursos.

Tipos de usuarios:

- Root / Administrador: Acceso total al sistema (riesgo y poder).
- Usuario estándar: Acceso limitado para tareas comunes.
- Invitado: Acceso mínimo, generalmente solo lectura.

Permisos de archivos (modelo POSIX en Unix/Linux):

r = read (lectura) w = write (escritura) x = execute (ejecución)

- Ejemplo en Linux:

-rwxr-xr-

Propietario: lectura, escritura, ejecución

Grupo: lectura y ejecución

Otros: solo lectura

Control de acceso: gestión de seguridad a través de **grupos**, **ACLs** (Access Control Lists), y **políticas de usuario**. Es fundamental para mantener el aislamiento y la integridad del sistema.

7. Gestión de memoria

Objetivo: maximizar el uso eficiente de la RAM mientras se proporciona aislamiento entre procesos.

Conceptos clave:

- Segmentación y paginación: partición lógica y física
- MMU (Unidad de gestión de memoria): traducción de direcciones lógicas a físicas
- Memoria virtual: permite sobrepasar los límites físicos mediante el uso de swap/page files

Técnicas avanzadas:

- Copy-On-Write (COW)
 - Page replacement (LRU, FIFO)
 - Protección por privilegios y segmentación
-

8. CRUD en archivos (Operaciones fundamentales)

El ciclo de vida de cualquier archivo en un sistema se resume en operaciones CRUD:

- **Create:** Crear archivo o carpeta
- **Read:** Leer datos de un archivo
- **Update:** Modificar contenido existente
- **Delete:** Eliminar archivo/carpeta

Ejemplo práctico en CLI:

```
touch tareas.txt           # Create
cat tareas.txt             # Read
echo "Nueva tarea" >> tareas.txt # Update
rm tareas.txt              # Delete
```

Estas operaciones son base de cualquier sistema de almacenamiento y son usadas también en bases de datos, APIs y sistemas web.

9. Sistemas de archivos

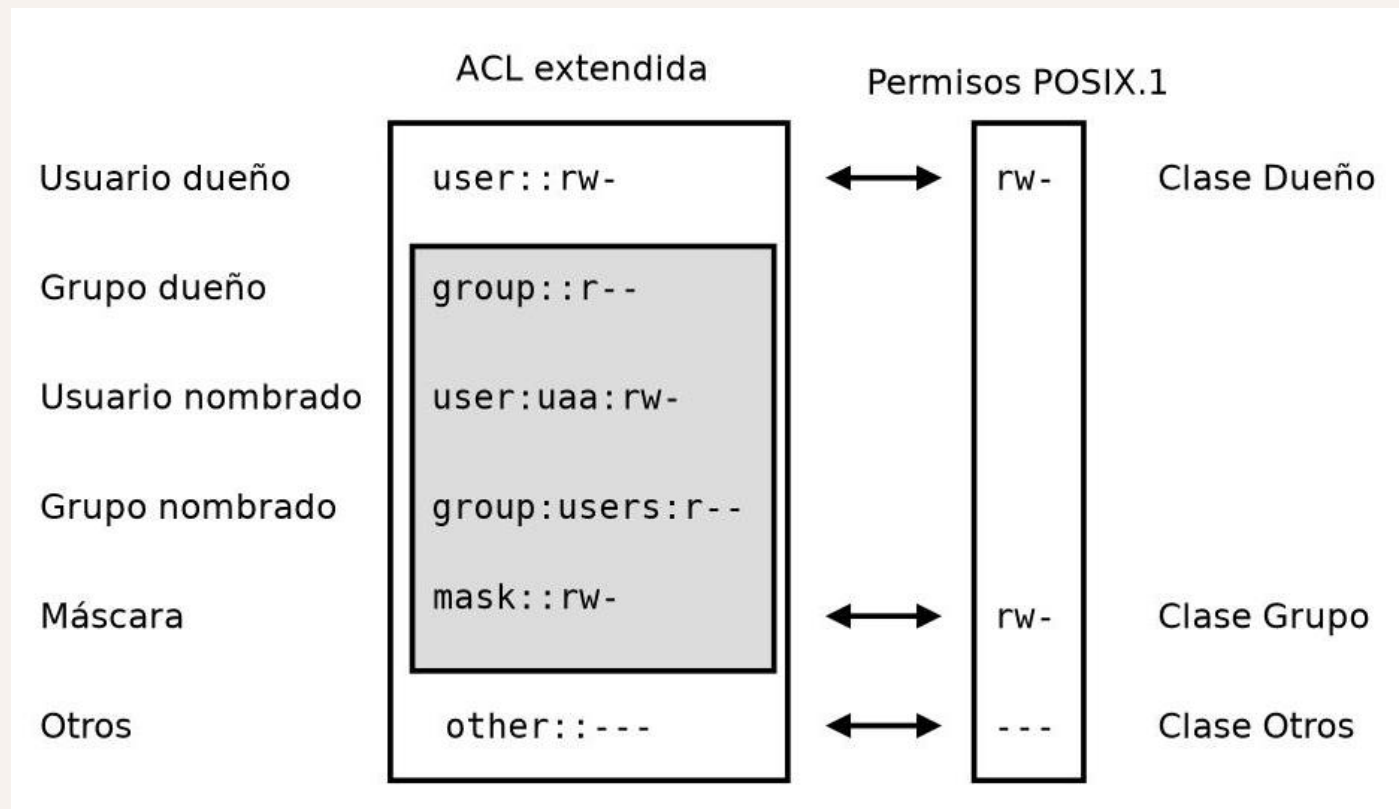
El SO abstrae los dispositivos de almacenamiento como un sistema de archivos estructurado.

Componentes:

- i-nodes (Unix), FAT tables (Windows)
- Árbol de directorios, enlaces duros y simbólicos
- Journaling (NTFS, ext4) para recuperación tras fallos

Seguridad:

- Permisos POSIX (rwx para user/group/others)
- ACLs (Access Control Lists)



10. Comandos comunes por sistema operativo

- Linux / macOS (bash/zsh)

Comando	Acción
ls	Lista archivos y carpetas
cd	Cambia de directorio
mkdir	Crea una carpeta
rm	Elimina archivos
chmod	Cambia permisos
ps	Muestra procesos activos

- Windows (cmd / PowerShell)

Comando	Acción
dir	Lista archivos
cd	Cambia de carpeta
mkdir	Crea una carpeta
del	Elimina archivos
tasklis	Muestra procesos
attrib	Cambia atributos de archivo