

Arquitectura del Computador y Sistemas Operativos

Décima Clase



03

Fundamentos de Sistemas Operativos



¿Por qué necesitamos un Sistema Operativo?

Las computadoras modernas son sistemas complejos, compuestos por múltiples componentes que a menudo fueron hechos por diferentes fabricantes. Las posibles combinaciones son prácticamente infinitas.

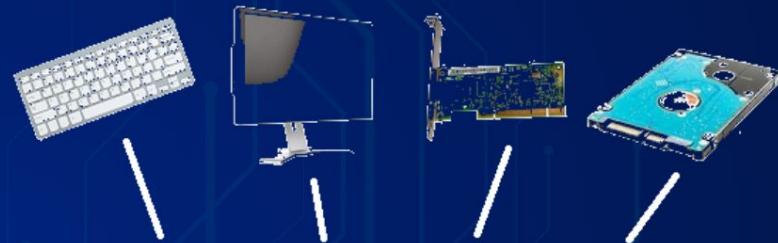
Todo fabricante de software desea que sus programas corran en la mayor cantidad de ordenadores, para tener más potenciales clientes. Sin embargo, el trabajo de hacer que su software corra con cada posible teclado, mouse, monitor, impresora, scanner, etc. sería enorme. Definitivamente la tarea de soportar todos estos dispositivos demandaría mucho más tiempo de desarrollo que la aplicación que desea comercializar en sí.

so

La programación sin SO (1/2)

La cantidad de trabajo para implementar aún la aplicación más sencilla sería enorme.

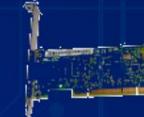
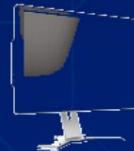
Tan grande que desarrollar aplicaciones sería inviable.



Teclado	Monitor	Placa de red	Disco
Rutinas de manejo de teclado	Rutinas de manejo de pantalla	Rutinas de manejo de red	Rutinas de manejo de disco
Distribuciones de teclados	FONTs, scrolling window mgr	Protocolos ARP, UDP, TCP, etc	Formatos FAT NTFS, HPFS EXTx, XFS, etc
Aplicación			



La programación sin SO (2/2)



Teclado	Monitor	Placa de red	Disco
Rutinas de manejo de teclado	Rutinas de manejo de pantalla	Rutinas de manejo de red	Rutinas de manejo de disco
Distribuciones de teclados	Fonts, scrolling window mgr	Protocolos ARP, UDP, TCP, etc	Formatos FAT NTFS, HPFS EXTx, XFS, etc
Aplicación			

Teclado	Monitor	Placa de red	Disco
Rutinas de manejo de teclado	Rutinas de manejo de pantalla	Rutinas de manejo de red	Rutinas de manejo de disco
Distribuciones de teclados	Fonts, scrolling window mgr	Protocolos ARP, UDP, TCP, etc	Formatos FAT NTFS, HPFS EXTx, XFS, etc
Aplicación			



La programación sin SO (2/2)



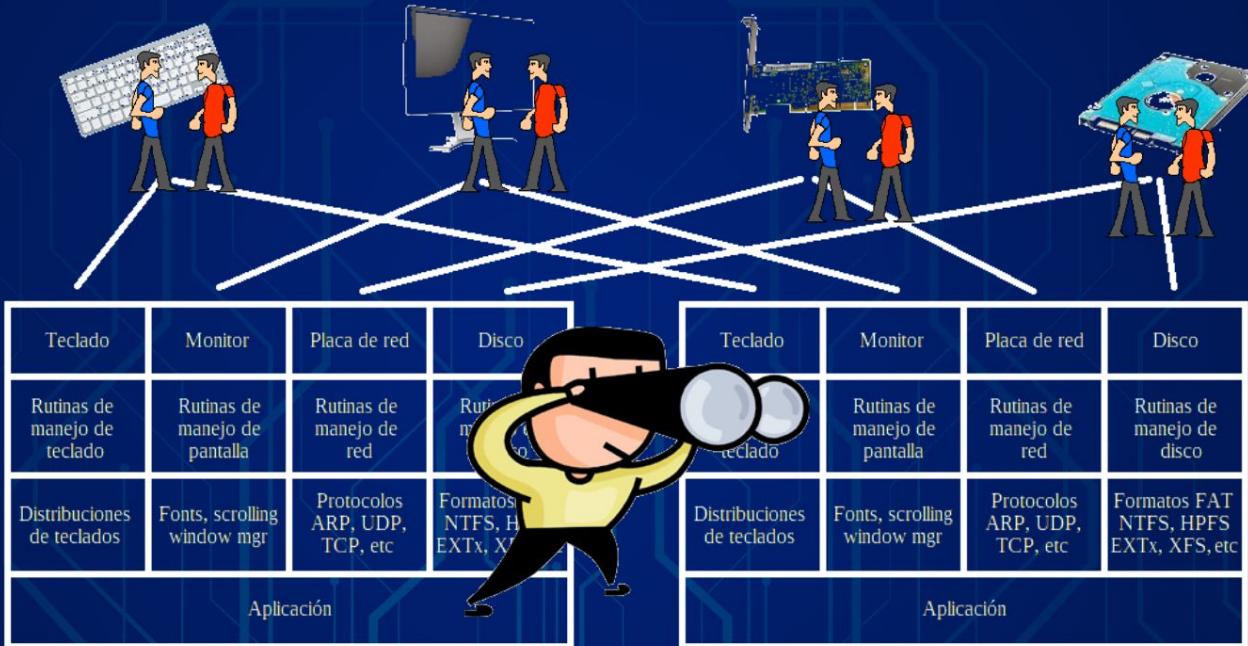
Teclado	Monitor	Placa de red	Disco
Rutinas de manejo de teclado	Rutinas de manejo de pantalla	Rutinas de manejo de red	Rutinas de manejo de disco
Distribuciones de teclados	Fonts, scrolling window mgr	Protocolos ARP, UDP, TCP, etc	Formatos FAT NTFS, HPFS EXTx, XFS, etc
Aplicación			

Teclado	Monitor	Placa de red	Disco
Rutinas de manejo de teclado	Rutinas de manejo de pantalla	Rutinas de manejo de red	Rutinas de manejo de disco
Distribuciones de teclados	Fonts, scrolling window mgr	Protocolos ARP, UDP, TCP, etc	Formatos FAT NTFS, HPFS EXTx, XFS, etc
Aplicación			

Y el trabajo de manejar cada dispositivo no es todo, todo se complica si deseamos que dos tareas usen el mismo recurso a la vez!



La programación sin SO (2/2)



Y el trabajo de manejar cada dispositivo no es todo, todo se complica si deseamos que dos tareas usen el mismo recurso a la vez!
Y no olvidemos la seguridad. Nada impide que una tarea espíe a otra!



SO

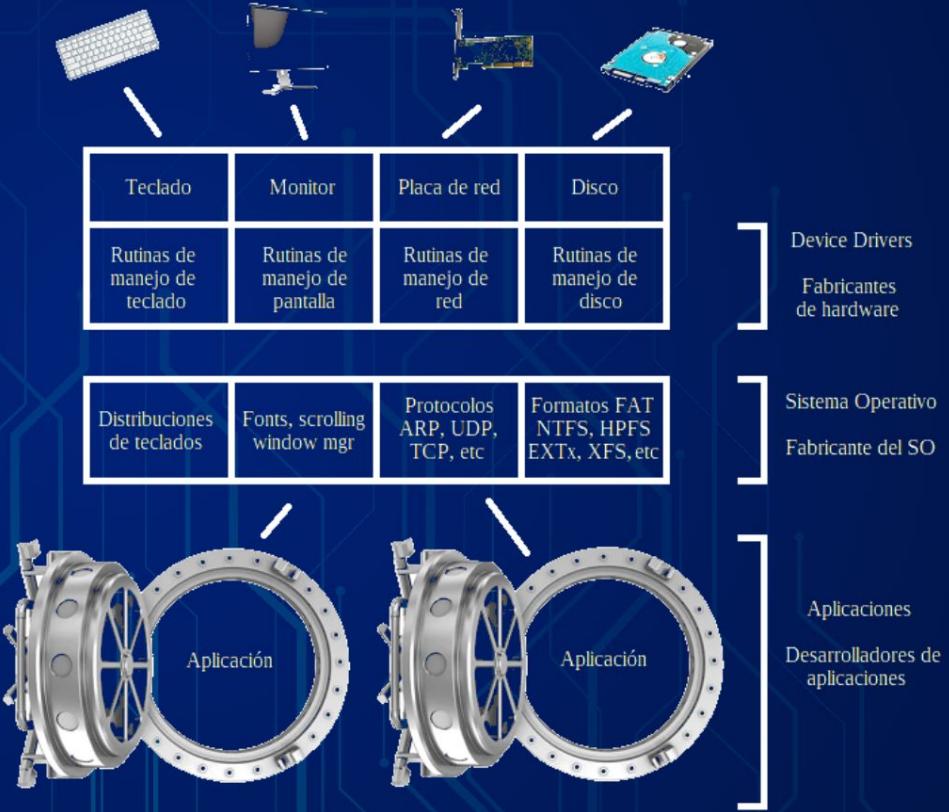
El Sistema Operativo y la división de las tareas

La creación del concepto de SO permite resolver los problemas vistos mediante la división de tareas.

El SO establece interfaces claras entre:

- SO y Driver
- SO y Aplicación

De esta forma cada uno puede hacer su parte sin preocuparse por las otras partes.





Un Sistema Operativo Moderno

- Define una interfaz clara para implementar “*Device Drivers*”.
- Define una interfaz clara a través de la que expone su funcionalidad a las tareas que ejecutará.
- El SO “virtualiza” los dispositivos. Cada tarea piensa que tiene el teclado, monitor, interfaz de red, disco, etc. para su exclusivo uso. De hecho trabaja asumiendo que éso es verdad, mientras que en realidad los comparte con todas las otras tareas.
- Implementa mecanismos para que una tarea no espíe ni altere otra.
- Si bien el programa que ve el usuario al arrancar el SO (sea en modo texto TUI o gráfico GUI) no es parte del mismo sino que es una tarea más, suele venir con el SO. Su función es abstraer el usuario del hardware tanto como sea posible y darle un manejo simple e intuitivo del sistema.
- Reparte la memoria entre las tareas de forma que cada una “vea” sólo la suya



Un poco de historia... (1/4)

Los primeros ordenadores eran construidos y operados por equipos de cerrados de personas, generalmente en universidades. Los programas accedían directamente al hardware que conocían perfectamente (a veces incluso había que modificar el HW para correr ciertos programas). No tenían Sistema Operativo.

Luego comenzaron a comercializarse, y los ingenieros de desarrollo se separaron de los programadores. Estas máquinas tenían dos dispositivos de E/S, un lector de tarjetas y una impresora.

Más tarde aparece la cinta magnética como dispositivo de E/S. Permitía cargarle un conjunto de tareas que se ejecutaban en forma secuencial (sin multitarea) que se llamaba lote o “batch” que tomaba el programa de la cinta y guardaba los resultados ahí. Este “programa” podría considerarse el primer SO.



Un poco de historia... (2/4)

Con la expansión de las computadoras del mundo de los problemas matemáticos complejos (mucho cálculo, poca E/S) a otras tareas, se perdía mucho tiempo en E/S y aparece la necesidad de aprovechar ese tiempo muerto.

Nace la multitarea, pero no con programas que coexisten, sino como una virtualización del caro CPU, que saltaba de un banco de memoria a otro, cada uno con un programa distinto. Estos bancos también servían como “cola de jobs”, dado que cuando un programa terminaba se cargaba el siguiente en el banco libre mientras la CPU ejecutaba los otros.

La idea se extendió a multiplicar la cantidad de bancos y multiplexar la atención del procesador y aparecieron sistemas con muchas terminales. Ésto volvió a conformar a los programadores, que volvieron a tener control interactivo al igual que con las primeras computadoras.

Con la aparición de computadoras pequeñas y baratas desde mediados de los '70, aparece el primer SO de microcomputadoras (el CP/M) que dominó el mercado durante unos años.



Un poco de historia... (3/4)

A principios de los '80 IBM desarrolla la IBM PC, y buscaba un SO. Bill Gates sugiere que contacten a Digital Research (propietaria del CP/M), pero DR no demuestra interés en el negocio. Bill Gates compra un SO y le propone usarlo a IBM. IBM acepta y aparece el DOS, primer SO de lo que ahora es Microsoft.

El DOS era monotarea (a excepción de los TSRs), tenía un concepto muy básico de drivers y solamente exponía un teclado, un monitor y acceso a archivos en disco.

A mediados de los '80 Windows y MAC OS cambian a una interfaz gráfica. Nace la multitarea en las microcomputadoras personales y el concepto de "User Friendly", conceptos que siguen hasta el día de hoy.

En 1996, con se hace popular el Windows NT 4.0, que no solamente implementaba multitarea sino protección entre ellas.

Linux se hace popular en las microcomputadoras recién durante los '90 y su uso ha crecido hasta hoy, principalmente por su robustez y licenciamiento gratuito.



Presiona **Esc** para salir pantalla completa

Un poco de historia... (4/4)

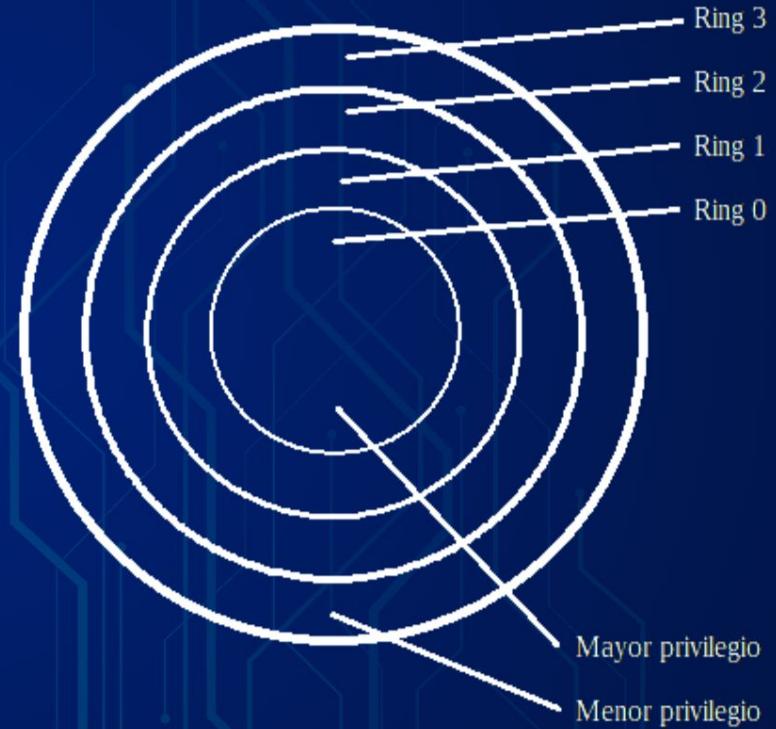
Le la misma forma que en los '80 las computadoras llegan a los hogares y pequeñas Empresas, durante los '00 llegan a dispositivos cada vez más pequeños. Primero son las tabletas y celulares, pero sigue con relojes y hasta electrodomésticos. Aparecen Android y IOS como SOs para estos dispositivos.



Protección entre tareas (1/2)

Es imposible pensar un sistema de protección entre tareas eficiente (no solo eficaz sino rápido) sin la ayuda del hardware.

Los procesadores ejecutan el código en distintos niveles de privilegio o “rings”. Cada uno puede hacer ciertas cosas, si alguien quiere ejecutar lo que no puede, se suspende la tarea y transfiere el control al nivel inferior (de mayor privilegio).





SO

Protección entre tareas (2/2)

El uso teórico de los anillos es:

Ring	Uso
0	Sistema operativo
1	Drivers del sistema operativo
2	Tareas privilegiadas (login, manipulación de credenciales, etc)
3	Tareas comunes

Sin embargo los SOs usan sólo algunos de ellos:

Ring	Windows	Linux	Mac OS	OS/2	Hipervisores
0	✓	✓	✓	✓	
1					✓
2					✓
3	✓	✓	✓	✓	



Tipos de Sistemas Operativos (1/2)

Clasificados por función, podemos encontrar distintos tipos de SO:

- Para mainframes: Atienden grandes cantidades de terminales. Están orientados a trabajos batch, transaccionales o multitasking.
- Para servers: Orientados a atender a muchos usuarios por la red, que comparten los recursos del servidor. Muy comunes hoy en día en internet.
- Para arquitecturas multiprocesador: Tiene soporte para múltiples procesadores. Según la estructura puede clasificarse como “computación paralela” (división del problema), “multi-computadoras” (varias computadoras unidas por backbone) o “multiprocesadores”.
- Para computadores personales: Menos volumen de datos, simples y orientados a un solo usuario.
- Para dispositivos personales: Un solo usuario, intuitivos. Soporte para GPS y touch screen.



Tipos de Sistemas Operativos (2/2)

- Para sistemas embebidos: Se usan en heladeras, televisores, etc. Son equipos con dispositivos de I/O limitados, generalmente orientados a aplicaciones puntuales, no de propósitos generales.
- SO de tiempo real: Para aplicaciones que no pueden demorar ciertas tareas, como control de trenes, algunos procesos industriales, control de vuelo de un misil, etc. Usar un “tipo virtual” puede terminar en grandes desastres. Algunos garantizan el cumplimiento de ciertas metas (hard) otros simplemente lo intentan (soft).



SO

Alternativas de implementación

Se exploraron varias alternativas de implementación de un SO, aunque este listado no debe ser considerado como un tema cerrado:

- Monolíticos: Es SO es un único “gran” bloque que implementa todo sin una estructura especial. Dentro del mismo cada función puede llamar a cualquier otra.
- En capas: Se definen capas de abstracción, por ejemplo la de abajo virtualiza la CPU, la segunda maneja la memoria, etc. Cada función del SO puede llamar a las de su capa o capas inferiores.
- Microkernels: Dejar la mínima expresión en el kernel, todo el resto va al espacio de usuario. Minimizan los errores críticos.
- Cliente-Servidor: Caso particular de lo anterior. Hay procesos de usuario para manejo de los dispositivos a los que se accede a través del kernel.
- Máquinas virtuales: Sólo virtualizan el hardware.
- Exokernels: Recursos repartidos entre máquinas, cada uno controla su parte. El exokernel reparte y monitorea usos indebidos.



Fin
¿Preguntas?