UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL CÓRDOBA - EXTENSIÓN ÁULICA
BARILOCHE
INGENIERÍA EN SISTEMAS DE INFORMACIÓN
AÑO LECTIVO 2025

Sistemas Operativos

Resumen para el primer parcial

Unidades 1, 2 y 3 de la cátedra

Profesor: Eduardo Tapia
Ayudante:
Fecha: 24/05/2025

Alumno: Ricardo Nicolás Freccero
Número de legajo: 415753

Índice

1. Unidad 1 - Concepto de Sistemas Operativos

1.1. ¿Qué es el sistema operativo?

Un sistema operativo es un programa que controla la ejecución de aplicaciones y programas. Es quien se encarga de "conectar" o "comunicar" las apliaciones con el hardeware de la computadora. Se puede considerar que un sistema operativo tiene tres objetivos:

- **Facilidad de uso.** Un sistema operativo facilita el uso de un computador.
- **Eficiencia.** Un sistema operativo permite que los recursos de un sistema de computación se puedan utilizar de una manera eficiente.
- **Capacidad para evolucionar.** Un sistema operativo se debe construir de tal forma que se puedan desarrollar, probar e introducir nuevas funciones en el sistema sin interferir con su servicio.

1.1.1. El sistema operativo como una interaz de usuario/computador

El hardware y software que le permiten al usuario acceder y utilizar las aplicaciones y programas de una computadora se pueden ver de forma jerárquica como muestra la Figura 1. Al usuario no le suelen interesar los detalles del hardware de la computadora y vé al sistema de computación como un conjunto de aplicaciones. Por otro lado, cada aplicación se puede expresar en un lenguaje de programación y normalmente es desarrollada por un programador. Al programador sí le importan los detalles del hardware, pero no se comunica casi nunca directamente con él ya que suele ser una tarea extremadamente compleja. Para eso existe un *conjunto de programas de sistema* que le permiten al programador comunicarse de una manera mas eficiente con el hardware. El programa de sistema mas importante es el **sistema operativo** que proporciona una interfáz entre el software y el hardware, actuando como mediador y facilitando el acceso del programador a las utilidades y servicios del sistema.

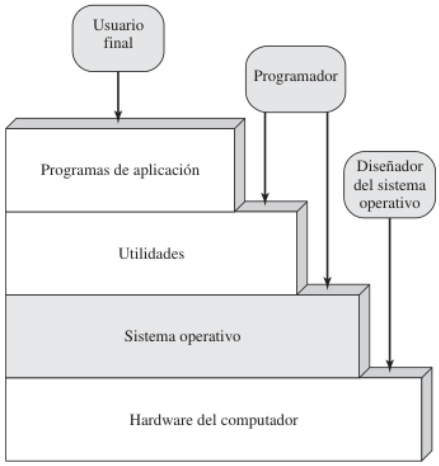

Figura 2.1. Capas y vistas de un sistema de computación.

Figura 1: Imagen sacada de (Stallings, 2005). Jerarquía de un sistema de computación.

El sistema operativo proporciona normalmente servicios en las siguientes áreas:

- **Desarrollo de programas.** Ofrece editores y depuradores para asistir al programador en la creación de programas y aplicaciones.
- **Ejecución de programas.** Se encarga de realizar todos los pasos necesarios para la ejecución de programas en nnombre del usuario.
- **Acceso a dispositivos de E/S.** Facilita el acceso de los programadores a estos dispositivos.
- **Acceso al sistema.** Protege los recursos y los datos,e vitando el uso no autorizado de los usu-sarios y resolviendo conflictos en el caso de conflico de recursos.
- **Detección y respuesta a errores.** Debe proporcionar una respuesta a cualquier error que pueda surgir de manera que se elimine la condición de error, suponiendo el menor impacto en las aplicaciones que están en ejecución.
- **Contabilidad.** Recoge estadísticas de uso de los diferentes recursos y monitoriza parámetros de rendimiento. Esta información es útil para anticipar las necesidades de mejoras futuras y para optimizar el sistema a fin de mejorar su rendimiento.

1.1.2. El sistema operativo como gestor de recursos

El sistema operativo dirige al procesador en el uso de los recursos de la computadora y en el tiempo que debe tomarse para la ejecución de otros programas. Para eso, el sistema operativo cede el control para que el procesador realice sus tareas y luego retoma el control para decirle qué sigue.

El sistema operativo decide cuándo un programa en ejecución puede utilizar un dispositivo de E/S y controla el acceso y uso de los ficheros, además decide cuánto tiempo debe tomarse cada procesador para la ejecución de un programa.

1.1.3. Facilidad de evolución de un sistema operativo

Un sistema operativo debe evolucionar en el tiempo por las siguientes razones:

- Actualizacioes de hardware y nuevos tipos de hardware.
- Nuevos servicios. (Por ejemplo, si es dificil mantener un buen rendimiento con las herramientas existentes, se pueden añadir al sistema operativo nuevas herramientas de medida y control.)
- Resolución de fallos.

1.2. Evolución de los sistemas operativos

1.2.1. Procesamiento en serie

El programador interactuaba directamente con el hardware de la computadora, que contaba con una consola, luces, interruptores, algún dispositivo de entrada y una impresora; *no existía ningún sistema operativo*. Los programas en código máquina se cargaban a través del dispositivo de entrada y si un error provocaba la parada del programa, las luces indicaban la condición de error. El programador podía examinar los registros del procesador y memoria principal para determinarl la causa del error. Si el programa terminaba de forma normal, la salida se imprimía.

1.2.2. Sistemas en lotes sencillos

Se empieza a usar un software denominado **monitor**, que permite que el usuario no tenga que acceder directamente a la máquina. En su lugar, el la computadora recibe una secuencia de trabajos que va a usar el monitor. El monitor de encarga de leer uno a uno cada trabajo y decirle al procesador que los vaya realizando en ese orden.

1.2.3. Sistemas en lotes multiprogramados

Aún usando un sistema de lotes sencillos, el procesador está haciendo nada la mayoría del tiempo ya que este es mucho mas rápido que los dispositivos de E/S. La computadora pasa mas tiempo buscando la información que tiene que procesar el procesador que procesando esa información. Lo que se puede hacer entonces es que mientras se está esperando por la E/S, se le puede asignar al procesador otro trabajo que no esté esperando por una operación de E/S.

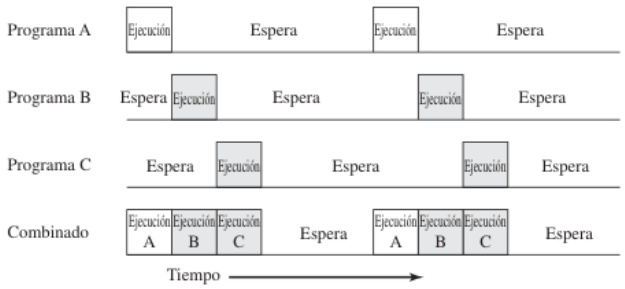

Figura 2: Imagen sacada de (Stallings, 2005). Ejemplo de multiprogramación con tres programas.

1.2.4. Sistemas de tiempo compartido

Son sistemas que comparten el tiempo de programación entre múltiples usuarios.

1.3. Desarrollos que llevaron a los sistemas operativos modernos

1.3.1. Arquitectura micronúcleo o microkernel

Antes, la mayoría de los sistemas operativos estaban formados por **núcleos monolíticos**. Estos núcleos proporcinaban la mayoría de las funcionalidades consideradas propias del sistema operativo, incluyendo planificación, los sistemas de ficheros, las redes, los controladores de dispositivos, la gestióno de memoria, etc. Una **arquitectura microkernel** asigna sólo unas pocas funciones esenciales al kernel, incluyendo los espacios de almacenamiento, comunicación entre procesos, y planificación básica.

1.3.2. Multithreading

Es una técnica en la cual un proceso, ejecutando una aplicación, se divide en una serie de *hilos* o *threads*.

- **Thread o hilo.** Es una unidad de trabajo. Incluye el contexto del procesador (que contiene el contador del programa y el puntero de pila) y su propia área de datos para una pila (para posibilitar el salto a subrutinas). Un hilo se ejecuta secuencialmente y se puede interrumpir para dar paso a otro hilo.
- **Proceso.** Es una colección de uno o más hilos y sus recursos de sistema asociados. Es un programa en ejecución.

Esta técnica es útil para aplicaciones que llevan a cabo tareas que no necesitan correrse en serie, es decir, que se pueden ejecutar en simultáneo.

1.3.3. Multiprocesamiento simétrico (SMP)

Se refiere a la arquitectura del hardware de la computadora y al comportamiento del sistema operativo que explota dicha arquitectura. Tiene las siguientes características:

- Tiene múltiples procesadores.
- Los procosadores comparten las mismas utilidades de momeira principal y de E/S.
- Todos los procesadores pueden realizar las mismas funciones.

El sistema operativo de un SMP planifica procesos o hilos a través de todos los procesadores.

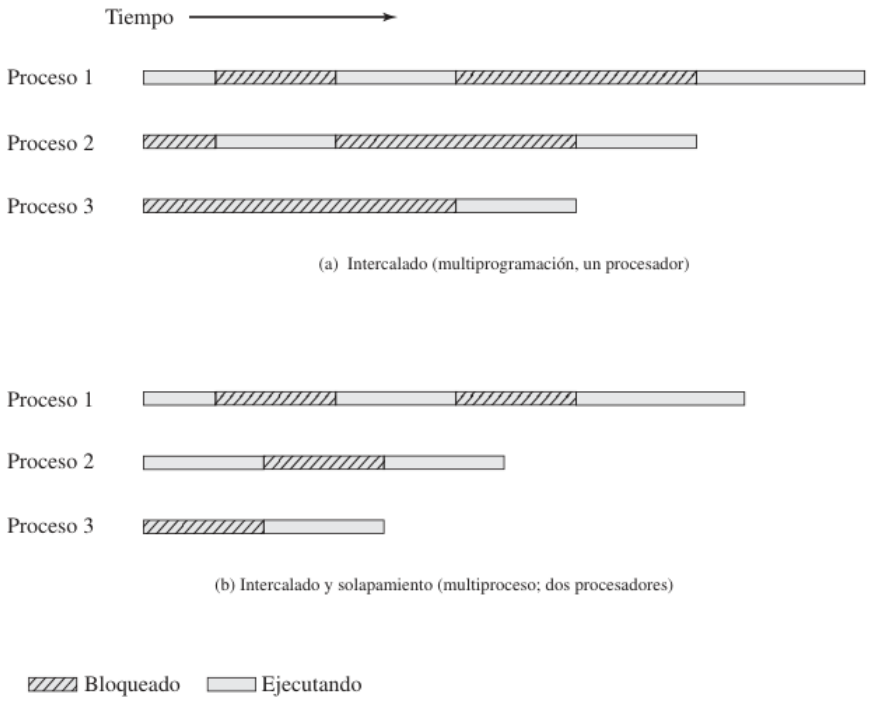

Figura 3: Imagen sacada de (Stallings, 2005). Multiprogramación y multiproceso.

1.3.4. Sistema operativo distribuído

Da la ilusión de tener un solo espacio de memoria principal y un solo espacio de memoria secundario, cuando tenemos un “cluster” de computadoras (varias computadoras que operan como si fuese una sola).

1.3.5. Diseño orientado a objetos

Introduce una disciplina al proceso de añadir extensiones modulares a un pequeño núcleo. Permite a los programadores personalizar un sistema operativo sin eliminar la integridad del sistema.

1.4. Sistemas UNIX tradicionales

UNIX es un sistema operativo que se desarrollo inicialmente en los laboratorios de Bell y se hizo operacional en 1970. En la figura 4 se puede ver la arquitectura general de UNIX.

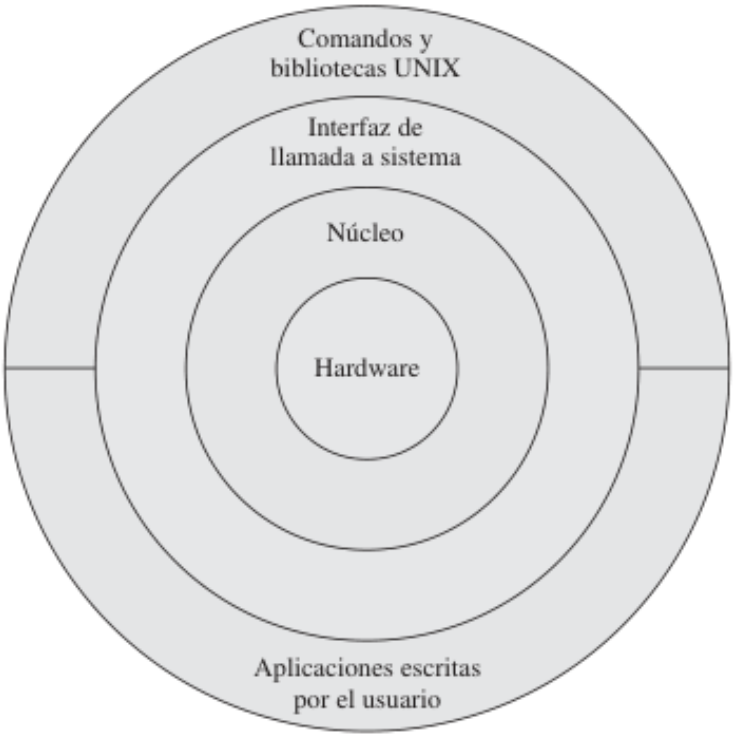

Figura 4: Imagen sacad de (Stallings, 2005). Arquitectura general de UNIX.

1.5. Linux

Linux comenzó como una variante UNIX. Linus Torvalds, un estudiante finlandés de informática, escribió la versión inicial. Linux es un sistema UNIX completo de software libre (cualquier persona puede ver y modificar el código fuente).

La mayoría de los núcleos de Linux son monolíticos. Aunque no usa la técnica de microkernel, logra muchas de las ventajas potenciales de esta por medio de su arquitectura modular particular. Linux está estructurado como una colección de módulos, algunos de los cuales pueden cargarse y descargarse automáticamente bajo demanda. Estos bloques se denominan **módulos cargables**.

Los módulos cargables de Linux tienen dos características importantes:

- **Enlace dinámico**. Un módulo de núcleo puede cargarse y enlazarase al núcleo mientras el núcleo está en memoria y ejecutándose. Un módulo también puede desenlazarase y eliminarse de la memoria en cualquier momento.
- **Módulos apilables**. Los módulos se gestionan como una jerarquía. Cuando un módulo superior en la jerarquía referencia a un módulo inferior, el primero actúa como módulo cliente y el segundo como biblioteca.

2. Unidad 2 - Administración y Gestión de Archivos

Referencias

Stallings, W. (2005). *Sistemas Operativos* (5.ª ed.). Pearson Educación, S.A.