



**Universidad Nacional Autónoma
de México**

Facultad de Ingeniería



Sistemas Operativos

*UNIX Y SU PAPEL EN LA HISTORIA DE LOS SISTEMAS
OPERATIVOS.*

Acosta Jacinto Alan, "320101179"

Rubio Carmona Jose Angel, "320118937"

Sexto semestre.

Fecha de entrega : 21 de abril de 2025

Profesora: Gunnar Eyal Wolf Iszaevich

Semestre 2025-2

Ciudad de México, 21 de abril de 2025.

1.-UNIX

Hablar de Unix hoy en día es hablar del origen de muchas de las tecnologías que usamos a diario. Aunque nació a finales de los años 60, sus principios siguen presentes en sistemas modernos como Linux, macOS, Android, e incluso en servidores y supercomputadoras que sostienen internet. Su diseño simple, modular y portátil revolucionó la forma en que se construyen los sistemas operativos. Comprender Unix es clave para entender la base técnica de la informática actual y el desarrollo de software moderno.

Una frase muy representativa sobre el legado de Unix es de **Dennis Ritchie**, uno de sus creadores:

— *Dennis Ritchie*

"Unix es simple. Solo se necesita un genio para entender su simplicidad."

2. Historia de UNIX

- Contexto histórico (1960s-1970s):

Como parte de este acuerdo, AT&T dirigió una pequeña fracción de sus ingresos a Bell Labs, con el propósito expreso de mejorar las comunicaciones. Monopolio , Transistor.

CTSS ("Compatible Time-Sharing System")

Esta falta de uniformidad entre los sistemas y el uso de lenguajes de bajo nivel incompatibles dificultaron enormemente el progreso. Un programa escrito para un sistema operativo tenía que ser reescrito desde cero para pasar a un sistema operativo o arquitectura diferente.

MULTICS y sus complicaciones.

Multics era intrínsecamente una perspectiva desafiante, y pronto se encontró con problemas. En retrospectiva, fue en parte víctima del segundo efecto del sistema: después de un éxito (como CTSS). La frase "sobre-ingeniería" aparece en varias descripciones, y Sam Morgan la describió como "un intento de trepar a demasiados árboles a la vez".

Desde la perspectiva de los Laboratorios Bell, en 1968 estaba claro que, aunque Multics era un buen entorno informático para el puñado de personas a las que apoyaba, no iba a lograr su objetivo. En consecuencia, Bell Labs abandonó el proyecto en abril de 1969, dejando que MIT y GE siguieran adelante.

Ken Thompson y Dennis Ritchie en Bell Labs.

1. Ken Thompson

Ken Thompson comenzó a trabajar en Bell Labs en 1966, inicialmente colaborando en el proyecto Multics, Thompson decidió continuar explorando el diseño de sistemas operativos por su cuenta.

Su oportunidad surgió al encontrar un PDP-7 sin uso, una computadora modesta pero con una buena interfaz gráfica. En ella escribió un juego de exploración espacial, pero más importante aún, usó esta experiencia para comenzar a diseñar un sistema operativo más simple, eficiente y útil: Unix.

Sus principales aportaciones:

- Diseñó el núcleo original de Unix en lenguaje ensamblador para el PDP-7.
- Propuso el modelo de herramientas pequeñas combinables, que se convirtió en la filosofía Unix.
- Fue pionero en implementar el sistema de archivos jerárquico y la noción de “todo es un archivo”.
- Más adelante, trabajó junto a Joe Condon en el desarrollo de Belle, la primera computadora en alcanzar el nivel de maestro en ajedrez.

2. Dennis Ritchie

Dennis Ritchie se unió formalmente a Bell Labs en 1967 como parte del Computing Science Research Center. Inicialmente trabajó también en Multics, pero su mayor contribución llegó poco después: el diseño del lenguaje de programación C.

Sus principales aportaciones:

- Co-creador de Unix junto a Thompson, especialmente a partir de la migración del sistema al PDP-11.
- En 1973, reescribió Unix casi por completo en lenguaje C, lo que convirtió al sistema en portable: podía ser modificado para funcionar en distintas arquitecturas sin reescribir desde cero.
- Escribió el manual de referencia del lenguaje C y fue coautor del influyente libro *The C Programming Language* junto a Brian Kernighan.
- Ayudó a estandarizar C, que se convirtió en uno de los lenguajes más utilizados en programación de sistemas.
- Fue jefe del departamento de Sistemas de Software en Bell Labs, liderando el desarrollo del sistema operativo Plan 9.

Primer UNIX en un PDP-7.

Thompson encontró un equipo PDP-7 (Programmed Data Processor 7), una computadora de 18 bits fabricada por Digital Equipment Corporation (DEC) que estaba prácticamente en desuso dentro de Bell Labs. Aunque limitada en potencia para los estándares de la época, el PDP-7 tenía una interfaz gráfica decente y suficientes capacidades para experimentar con software de sistema.

¿Por qué el PDP-7?

- Tenía solo 8 KB de memoria principal (unas 8 mil palabras de 18 bits).
- Era una máquina barata, ya disponible, que no requería aprobación presupuestaria para usar.
- Su hardware era lo suficientemente flexible como para probar ideas experimentales, como sistemas de archivos, multitarea básica y llamadas al sistema.

Ken Thompson, con el apoyo ocasional de Dennis Ritchie, comenzó a escribir desde cero un sistema operativo simple, con las ideas clave de lo que luego se convertiría en Unix:

- Sistema de archivos jerárquico.
- Separación de procesos.
- Herramientas pequeñas y reutilizables.
- Manejo de archivos como flujos de bytes (todo es un archivo).

Desarrollo técnico

El primer Unix en el PDP-7 fue escrito completamente en lenguaje ensamblador, ya que no existía aún el lenguaje C. El sistema incluía:

- Un núcleo básico (kernel) capaz de manejar archivos y procesos.
- Un intérprete de comandos rudimentario (shell).

Este entorno era muy básico, pero funcional, y permitía por primera vez en Bell Labs que un sistema operativo respondiera a varios usuarios de forma interactiva, aunque uno a la vez (todavía no era multiusuario plenamente).

Transición hacia el PDP-11

El éxito del experimento en el PDP-7 motivó al grupo a solicitar una computadora más potente. Aunque la dirección no estaba interesada en financiar otro gran sistema como Multics, aceptaron la propuesta de adquirir un PDP-11, más moderno y económico. Esta decisión fue clave para la expansión de Unix: en el PDP-11, Unix fue reescrito en lenguaje C (por Dennis Ritchie), lo que lo hizo portable y escalable a futuro.

Frase destacada

“Unix fue creado con cero presupuesto, un equipo olvidado, y la necesidad de jugar un videojuego.”

— Parafraseando a Brian Kernighan

Impacto del PDP-7 en Unix

- Permitió la primera implementación funcional de Unix.
- Demostró que un sistema operativo podía ser eficiente, modular y escrito por pocas personas.
- Sirvió como laboratorio de pruebas para ideas que se mantienen vigentes hasta hoy.

Reescritura en C: pionera en portabilidad.

- Importancia del lenguaje C:

C se convirtió también en un estándar en la enseñanza de la programación a nivel universitario y sigue siendo imprescindible en áreas como sistemas embebidos, controladores de hardware, firmware y ciberseguridad.

4. Sistema de Archivos

Es la estructura que organiza los datos en una partición de disco. Se basa en un modelo arborescente y recursivo, en donde los nodos pueden ser tanto archivos como directorios, y estos últimos pueden contener a su vez directorios o subdirectorios.

En Unix todo se considera como un archivo. Esto permite que se pueda leer o escribir a un dispositivo.

Tipos de archivos:

Una de las ideas fundamentales de Unix es que “todo es un archivo”. Esto significa que, además de documentos de texto o programas, Unix trata directorios, dispositivos, y hasta conexiones de red como archivos, lo que unifica la forma de interactuar con el sistema.

ORDINARIOS – Contienen los datos de los usuarios (ASCII o binarios)

Son los archivos más comunes, que contienen datos creados o usados por los usuarios y programas.

DIRECTORIOS – Contienen archivos.

Son archivos especiales que almacenan referencias a otros archivos y directorios.
Son la base del sistema de archivos jerárquico de Unix.

ESPECIALES Representan rutinas del kernel que proporcionan acceso a diversos dispositivos del sistema;

- Bloque.

Dispositivos que transfieren datos en bloques de tamaño fijo.

- Carácter.

Dispositivos que transfieren datos en flujos de caracteres.

- FIFO (pipes)

Canales especiales para comunicación entre procesos (IPC), donde un proceso escribe y otro leer

- Sockets

Archivos especiales que permiten la comunicación entre procesos a través de redes o dentro del sistema local (IPC).

- Ligas simbólicas

Archivos especiales que actúan como accesos directos apuntando a otro archivo o directorio.

Unix utiliza diferentes tipos de archivos no solo para almacenar información, sino para interactuar con el sistema operativo, dispositivos físicos, otros procesos y servicios. Esta unificación bajo el modelo de “todo es un archivo” permite una interfaz simple y potente, y ha sido una de las razones del éxito y longevidad del sistema Unix y sus derivados como Linux y macOS.

UNIX EN MAC

COMANDOS

UNICS -Monousuario y monotarea

brian Kernighan

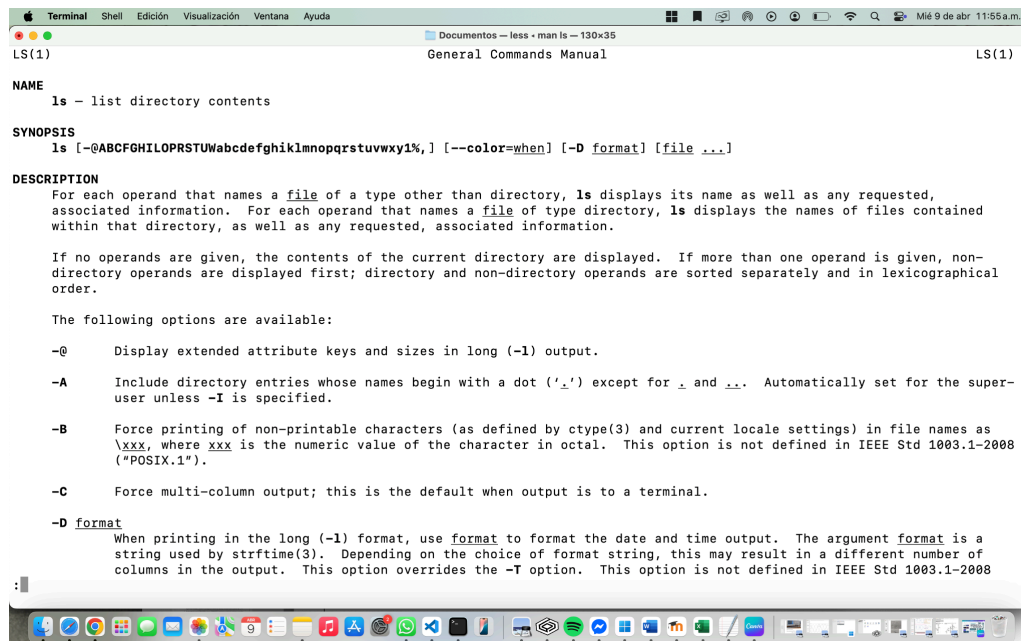
Luego pasó a UNIX

Los comandos básicos de unix, si se dan cuenta los comandos son muy similares a los que se utilizan en Linux. Esto se debe a que Linux tiene sus raíces en Unix, y muchos de sus comandos provienen directamente de la misma base. De hecho, una de las principales características de ambos sistemas operativos es su filosofía de herramientas pequeñas y especializadas que se pueden combinar para realizar tareas complejas. Esta compatibilidad entre comandos permite que los usuarios que dominan Unix puedan moverse fácilmente entre ambos sistemas.

Comando “man”

Este es un comando el cual sustituye la guía UNIX “programmer's manual”

Este se ocupa como man “comando deseado”.



Unix al inicio fue creado como un entorno sencillo y de manera modular(Es decir varios módulos que trabajan en conjunto conectados a través de PIPES)

PIPELINES

Un pipeline es una forma de conectar comandos entre sí, de modo que la salida de uno se convierte en la entrada del siguiente.

Se conectan con “| ” (pipe).

El comando de la izquierda produce salida.

El comando de la derecha la consume como entrada.

Se pueden encadenar tantos como sean posibles.

Comando GREP

Es una herramienta para buscar texto que coincida con un patrón o palabra, ya sea en uno o más archivos.

Buscar palabras clave en archivos de texto

- Filtrar resultados en logs
- Encontrar errores, nombres, valores, etc.
- Usarlo con otras herramientas (pipes)

Comando Sed

Stream Editor (editor de flujo)

Sirve para modificar texto automáticamente, línea por línea, sin abrir el archivo manualmente.

- Reemplazar palabras o frases
- Eliminar líneas específicas
- Insertar o modificar contenido
- Automatizar limpieza de datos

UNIX y su legado

Linux

Creado por Linus Torvalds, Linux es un sistema operativo inspirado en UNIX y basado originalmente en MINIX. Aunque no está basado directamente en el código fuente de UNIX, adopta su estructura, comandos, y filosofía. Hoy en día es el corazón de servidores, supercomputadoras, sistemas Android, dispositivos IoT y más.

Windows NT

Desarrollado por Microsoft como una respuesta empresarial a UNIX, Windows NT fue diseñado desde cero, pero con muchas ideas inspiradas en los sistemas UNIX, como la multitarea, seguridad para usuarios y arquitectura modular.

Mac OS X

Después del regreso de Steve Jobs a Apple, el sistema operativo clásico de Mac fue reemplazado por macOS, basado en UNIX BSD (Berkeley Software Distribution) a través de NeXTSTEP.

FUENTES

Kernighan, B. W. (2020). *UNIX: A history and a memoir*. Kindle Direct Publishing.

Hostinger Academy. (2021, 24 de febrero). *What is UNIX? UNIX Operating System Explained* [Video]. YouTube. <https://www.youtube.com/watch?v=KuGCq7L6WaM>

Hostinger. (n.d.). *50 Most Commonly Used Linux Commands (With Examples)*. Hostinger Tutorials. <https://www.hostinger.com/tutorials/linux-commands>

UNIX / Linux tutorial for beginners. (n.d.). Retrieved April 17, 2025, from Surrey.ac.uk website: <https://info-ee.surrey.ac.uk/Teaching/Unix/>

Lourdes Yolanda Flores Salgado, M. (n.d.). Introducción a UNIX. Retrieved April 17, 2025, from Unam.mx website: https://triton.astroscu.unam.mx/fruiz/introunix_plan_2013.pdf

AprendoLibre. (2022, 1 de noviembre). *¿POR QUÉ NO APRENDO?* [Video]. YouTube. <https://www.youtube.com/watch?v=YvcPJlsfL8U>.