

PRACTICA # 1

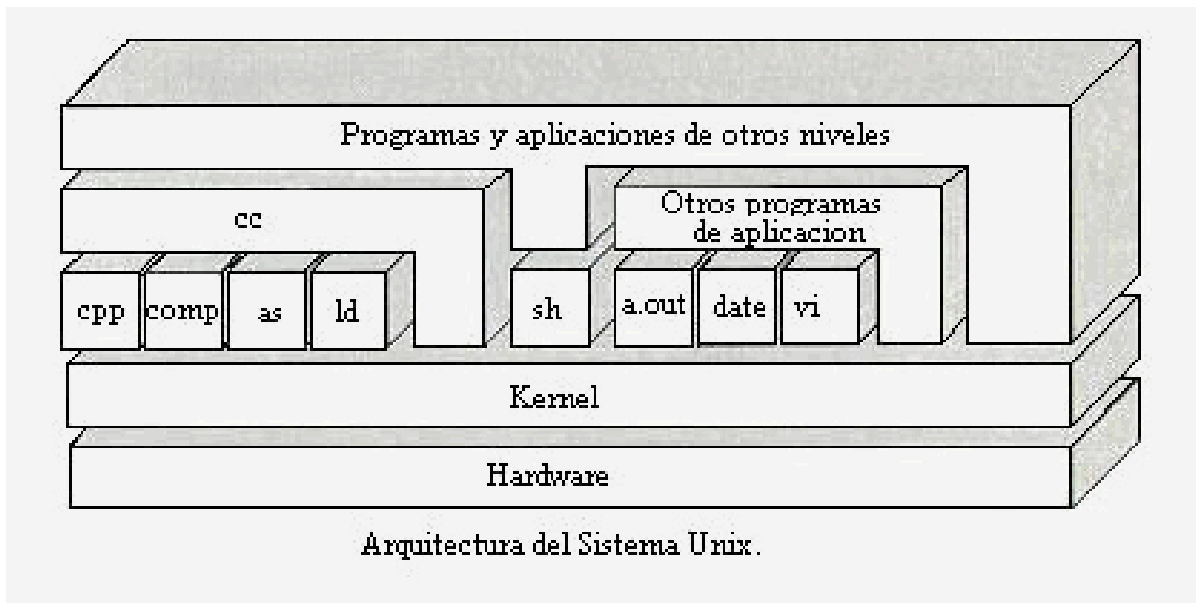
“RECONOCIMIENTO DEL S.O. UNIX”

Un **sistema operativo** (S.O.) es un programa o conjunto de programas que en un sistema informático gestiona los recursos de hardware y provee servicios a los programas de aplicación, ejecutándose en modo privilegiado respecto de los restantes.



En la actualidad existen varios S.O. siendo los mas populares: Microsoft Windows, Mac OS, OS/2 de IBM y todas las vertientes de UNIX. Todos los S.O. cuentan con un sistema de archivos (File System). Unix es un S.O. multitarea y multiusuario. Su sistema de archivos proporciona un método sencillo de organizar archivos y permite la protección de archivos. Sin embargo, las instrucciones del UNIX no son intuitivas.

ARQUITECTURA DEL SISTEMA OPERATIVO UNIX



El Kernel o Núcleo del sistema operativo Unix es un programa escrito casi en su totalidad en lenguaje C, con excepción de una parte del manejo de interrupciones, expresada en el lenguaje ensamblador del procesador en el que opera. Como se observa en la imagen, el kernel proporciona servicios a los programas de las capas que están arriba de él. Además gracias al kernel se cuenta con un ambiente multitarea y multiusuario.

El Kernel reside siempre en la memoria central y tiene el control sobre la computadora, por lo que ningún otro proceso puede interrumpirlo; sólo pueden llamarlo para que proporcione algún servicio de los ya mencionados. Un proceso llama al Kernel mediante módulos especiales conocidos como **llamadas al sistema** (o System Calls).

El Kernel inicializa un proceso especial, llamado proceso 0. En general, los procesos se crean mediante una llamada a una rutina del sistema (fork), que funciona por un mecanismo de duplicación de procesos. Sin embargo, esto no es suficiente para crear el primero de ellos, por lo que el Kernel asigna una estructura de datos y establece apuntadores a una sección especial de la memoria, llamada tabla de procesos, que contendrá los descriptores de cada uno de los procesos existentes en el sistema.

Después de haber creado el proceso 0, se hace una copia del mismo, con lo que se crea el proceso 1; éste muy pronto se encargará de "dar vida" al sistema completo, mediante la activación de otros procesos que también forman parte del núcleo. Es decir, se inicia una cadena de activaciones de procesos, entre los cuales destaca el conocido como despachador, o scheduler, que es el responsable de decidir cuál proceso se ejecutará y cuáles van a entrar o salir de la memoria central. A partir de ese momento se conoce el número 1 como proceso de inicialización del sistema, init.

El proceso init es el responsable de establecer la estructura de procesos en Unix. Normalmente, es capaz de crear al menos dos estructuras distintas de procesos: el modo monousuario y el multiusuario. Comienza activando el intérprete del lenguaje de control (Shell) en la terminal principal, o consola, del sistema y proporcionándole privilegios de "súper usuario". En la modalidad de un solo usuario la consola permite iniciar una primera sesión, con privilegios especiales, e impide que las otras líneas de comunicación acepten iniciar sesiones nuevas. Esta modalidad se usa con frecuencia para revisar y reparar sistemas de archivos, realizar pruebas de funciones básicas del sistema y para otras actividades que requieren uso exclusivo de la computadora.

Init crea otro proceso, que espera pacientemente a que alguien entre en sesión en alguna línea de comunicación. Cuando esto sucede, realiza ajustes en el protocolo de la línea y ejecuta el programa login, que se encarga de atender inicialmente a los nuevos usuarios. Si la clave del usuario, y la contraseña proporcionadas son las correctas, entonces entra en operación el programa Shell, que en lo sucesivo se encargará de la atención normal del usuario que se dio de alta en esa terminal.

A partir de ese momento el responsable de atender al usuario en esa terminal es el intérprete Shell.