El sistema operativo UNIX – Linux I. Conceptos básicos

Ken Thompson y Dennis Ritchie se basaron en el proyecto **MULTICS**.
Después de una primera versión en ensamblador PDP-7 denominada **UNICS** que posteriormente pasó a llamarse UNIX, se reescribe en un lenguaje de alto nivel, llamado **B** y posteriormente, Thompson y Ritchie desarrollan el **C** y lo vuelven a reescribir.

Estandarización→**IEEE** con un proyecto colectivo denominado **POSIX (1003.1 procedimientos de biblioteca)**

- **1003.4** Extensiones para tiempo real.
- **1003.6** Extensiones para la seguridad
- **1003.7** Administración del Sistema.
- **1003.8** Acceso transparente a archivos.
- **1003.10** Supercómputo

LINUX= UNIX (Ken Thompson y Dennis Ritchie, adopta caract, especif, y funcionamiento) + MINICS (sist. Educativo de Tanenbaun, adopta estructura y código del núcleo)

Principios de diseño de UNIX

- Interactivo: Acepta órdenes las ejecutar y espera más órdenes
- Multiusuario: Puede ser usado por más de una persona simultáneamente
- Multitarea: Puede realizar a la vez varios trabajos (procesos)
- Razonablemente seguro. Existe protección de acceso a los recursos
- Multiplataforma: Se potencia la portabilidad de las aplicaciones a nivel de código fuente. Soporta múltiples arquitecturas de procesadores de 16, 32 y 64 bits
- Soporta varias CPU's (SMP, Symmetric MultiProcessing).
- Soporta procesos en tiempo real (Real Time según POSIX 1003.1b).
- multiprogramación y tiempo compartido.
- Es un sistema preemptive, con planificación dependiente de la categoría del proceso: tiempo real o proceso ordinario. Sobre procesos ordinarios emplea la técnica de planificación por prioridad dinámica (la asignada inicialmente más una variable asignada por el sistema). En tiempo real se aplica un planificador FIFO non-preemptive para ciertas tareas de igual prioridad o Round Robin preemptive en otros casos, también de tiempo real.

ARQUITECTURA UNIX / LINUX

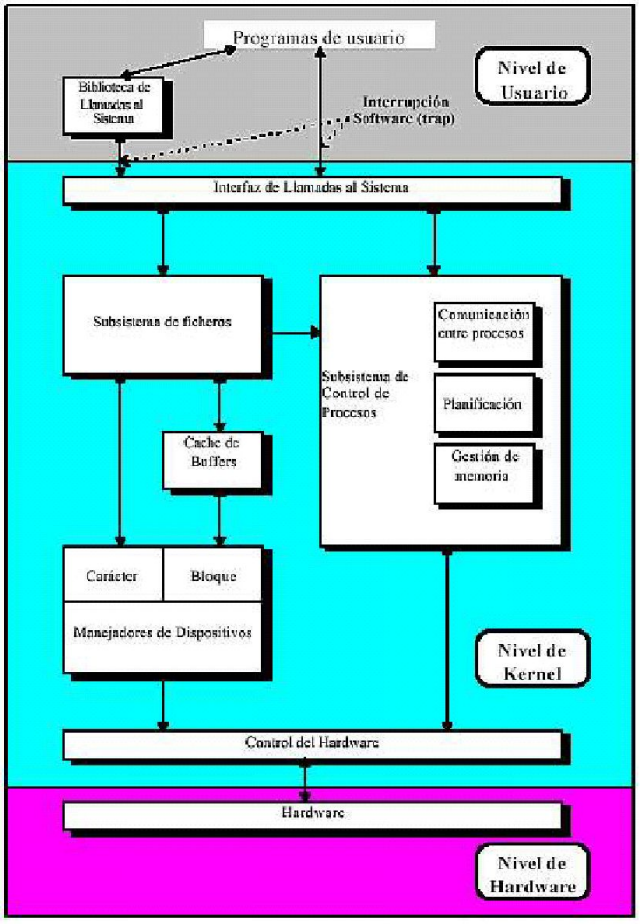
ELEMENTOS

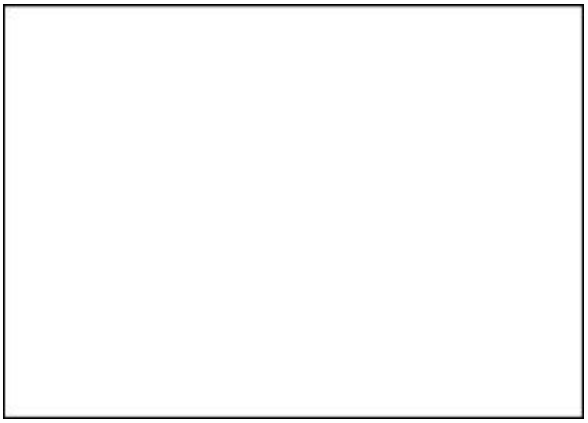
Los componentes del sistema Unix - Linux son:

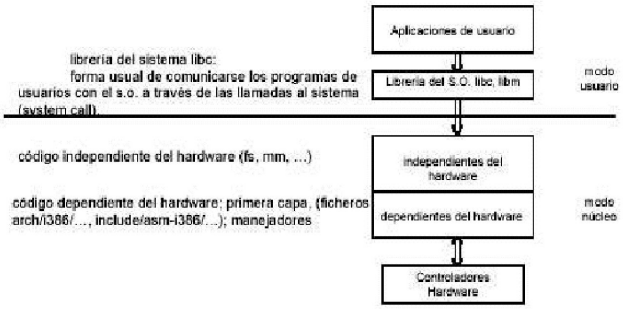
Kernel → Se clasifica como de tipo monolítico pero permite la incorporación dinámica de módulos

- **Núcleo dependiente**: Se encarga de las interrupciones, los manejadores de dispositivos de bajo nivel (lower half) y parte del manejo de la memoria.
- **Núcleo independiente: Es igual en todas las plataformas e incluye:**
    - Manejo de llamadas del sistema
    - Controlar la ejecución de procesos, permitiendo su creación, terminación o suspensión y comunicación
    - Planificar los procesos para su realización por la CPU. Los procesos la comparten mediante un sistema de tiempo compartido
    - Asignar memoria principal a un proceso en ejecución. Permite que los procesos compartan porciones de su espacio de direcciones bajo ciertas condiciones pero impide que un proceso tenga un espacio específico de direcciones.
    - Asigna memoria secundaria para lograr un almacenamiento y recuperación eficiente de los datos de usuario. Asigna almacenamiento secundario para los archivos de usuario, recupera el almacenamiento no empleado, estructura el sistema de archivos de un modo comprensible y los protege del acceso de usuarios no autorizados
    - Permite el acceso controlado de los procesos a los periféricos tales como: terminales, unidades de disco, equipos de red**.**

- Los usuarios o sus procesos a través **de traps** y **la interfaz de llamadas al sistema** acceden al modo kernel del sistema operativo. Los principales módulos a los que accederán serán el subsistema de control de procesos (gestión de memoria, planificación, comunicación entre procesos) y al subsistema de ficheros y manejadores de dispositivos.







Procesos y threads

Un proceso es una instancia de un programa en ejecución.

El subsistema de control de procesos (Process Control Subsystem) es el responsable de:

- Comunicación entre procesos
- Gestión de memoria
- Actuación del planificador

El módulo de comunicación entre procesos controla todas las posibilidades, desde señalización asíncrona de sucesos hasta transmisión síncrona de mensaje entre procesos

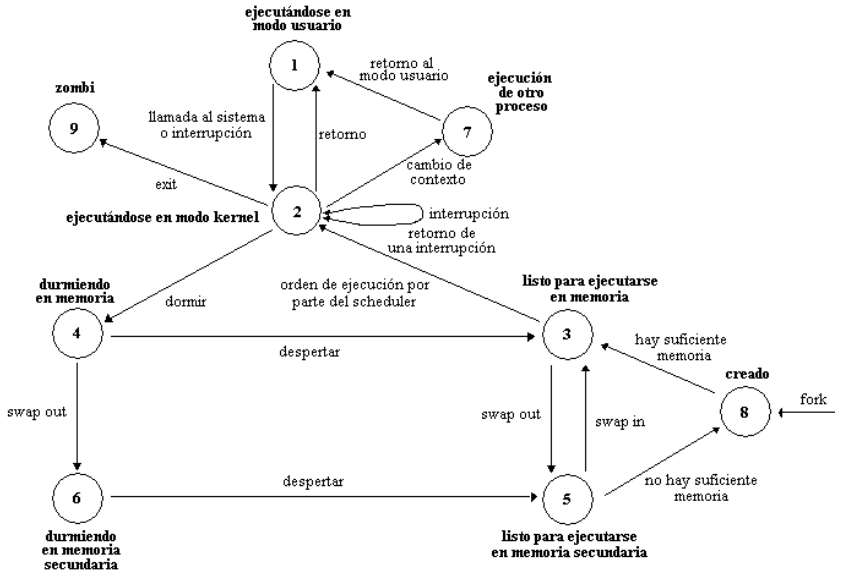
El scheduler (planificador) asigna CPU a los procesos. El manejo de procesos en UNIX es por **prioridad** y **round robin**. En algunas versiones se maneja también un **ajuste dinámico de la prioridad** de acuerdo al tiempo que los procesos han esperado y al tiempo que ya han usado el CPU

UNIX permite que un proceso haga una **copia de sí mismo** por medio de la llamada **'fork'**, lo cual es muy útil cuando se realizan trabajos paralelos o concurrentes; también se proveen facilidades para el envío de **mensajes entre procesos (pipes, signals)**

Los procesos no interactivos se denominan **daemons** o procesos background. Cuando se inicia un proceso, se le asigna un identificador **PID**, se guarda el proceso que lo lanzó **PPID**, el propietario que lo lanzó **UID** y el grupo de pertenencia **GID**, lo que definirá el perfil de permisos de acceso a los que tendrá derecho.

Un proceso puede estar en 9 estados

1. Ejecución en modo usuario
2. Ejecución en modo kernel
3. Ready to run, pero se halla preparado para funcionar a la espera de que el kernel lo decida,
4. Listo a la espera de que el swapper lo transfiera a memoria principal para que el kernel decida su ejecución
5. El proceso está durmiendo (sleeping) en la memoria principal
6. El proceso está sleeping y el swapper lo ha transferido al almacenamiento secundario para hacer sitio en la memoria principal
7. El proceso está volviendo desde el kernel al modo usuario, pero el kernel le da preeminencia a éste último y hace un cambio de contexto para realizar otro proceso
8. El proceso acaba de ser creado y se encuentra en estado de transición, existe pero no se halla preparado para funcionar ni está durmiendo
9. El proceso ha realizado la llamada exit y se encuentra en estado fantasma, ya no existe pero deja un registro con un código de salida y alguna información para el proceso padre



Trabajos no interactivos→ programar trabajos: nohup, cron, at

Gestión de memoria

El esquema más usado la **paginación por demanda** y **combinación de segmentos paginados**, en ambos casos con páginas de tamaño fijo.
En todos los sistemas UNIX **se usa una partición de disco duro para el área de intercambio.** Todos **los procesos que forman parte del kernel no pueden ser intercambiados** a disco.
Cada proceso dispone de su propio espacio de direcciones.

- Para la paginación se emplea un algoritmo conocido como **algoritmo de reloj** de una manecilla, examinando de forma circular los marcos de página.

El sistema de archivo

Existen varias versiones del sistema de archivos UNIX:
• Virtual File System (VFS) de SVR4
• Fast Files System (FFS) de Berkeley
• **Network File System (NFS) de Sun (actual)**
• Remote File System (RFS) de ATT

Cualquier sistema de archivos antes citado se caracteriza por:

- Una estructura jerárquica
- Tratamiento y protección coherente de los archivos de datos
- Capacidad de crear y destruir archivos
- Crecimiento dinámico de estos archivos
- Tratamiento de cualquier dispositivo periférico como archivo

Cada partición de un disco tiene una estructura:

- Bloque Boot : bloque de arranque
- Superbloque (nº de bloques, nº de i-nodes, comienzo de la lista de bloques libres)
- Bloques de datos
- **i-nodes**: Es la representación interna de un fichero. Su contenido son punteros a los bloques que forman los datos del fichero y propiedades del fichero (UID propietario, GUID, permisos, fecha, etc.). Hay uno por fichero (el número de ellos está limitado en la creación del sistema de archivos). **→ NO contiene el nombre del fichero**. Un directorio mantiene una lista con los nombres de los ficheros y asocia a cada nombre de fichero su i-nodo correspondiente. Distintos nombres de fichero pueden estar asociados al mismo i-nodo.

Gestión Entrada / Salida

Toda entrada / salida está basada en el principio de que todos los dispositivos se pueden tratar como ficheros simples. El subsistema de entrada / salida utiliza como elementos principales: Buffercache; el código general de manejo de dispositivos; y drivers de dispositivos de hardware.
Existen dos tipos de dispositivos

- Dispositivos de bloques:
    - Usan secuencias de bytes (bloques)
    - Utilizan buffer-cache
    - Están estructurados en bloques de tamaño fijo (512 bytes)
    - Permiten optimizar el rendimiento
- Dispositivos de carácter:
    - Son dispositivos sin estructura, son raw
    - No usan buffer
    - La operaciones se realizan carácter a carácter

Interrupciones y excepciones

UNIX permite a dispositivos como periféricos de I/O o al reloj del sistema interrumpir la CPU asíncronamente.
Los 6 niveles típicos de interrupción son:
• Interrupciones por SW • Terminales • Dispositivos de red • Disco • Reloj • Errores de máquina

Seguridad

Protección de ficheros: Los ficheros del sistema de archivos tienen tres niveles de permisos:

- Los del usuario individual
- Los del grupo al que el usuario pertenece
- Los de los demás usuarios de la máquina

Un fichero tiene tres grupos de permisos para cada uno de los tres niveles de seguridad: Lectura, escritura y ejecución para el propietario, para el grupo y para todos los usuarios (world). Se puede ceder la propiedad del fichero con la orden "chown" y la del grupo con la orden "chgrp".

Propietario	Grupo	Resto (world)
rwx	rwx	rwx

r: Lectura
w: Escritura
x: Ejecución

El corazón del esquema de seguridad de UNIX-LINUX es el **Id de presentación y la contraseña del usuario individual**. Puesto que la contraseña se almacena de forma cifrada, ni tan siquiera el administrador es capaz de determinar cuál es. La herramienta que permite modificar la contraseña **es passwd.** La información crítica que controla las identificaciones de los usuarios está mantenida en el fichero de base de datos **/etc/passwd→** legible por todos los usuarios pero no es modificable
Existe también un segundo fichero en el sistema para contener la contraseña cifrada y algunos otros datos. Este fichero es el **/etc/shadow** que es legible para root.. Este fichero contiene los Ids del usuario, sus contraseñas cifradas, un código numérico que describe cuándo fue modificada por última vez la contraseña y el número mínimo y máximo de días requerido entre días entre cambios de contraseña.

Cifrado de archivos:

- Mediante editores (vi, edit, emacs): permite cifrar/descifrar ficheros
- Mediante comando crypt: lee entrada estándar y escribe en salida estándar. Si la entrada es cifrada la salida es texto plano y viceversa. Usa DES de 56 bits.

Herramientas de seguridad:

- TripWire: esta herramienta mantiene una base de datos de sumas de comprobación de los ficheros importantes del sistema. Puede servir como sistema IDS preventivo. Nos sirve para "tomar" una foto del sistema, poder comparar después cualquier modificación introducida y comprobar que éste no haya sido corrompido por un atacante
- Nmap: es una herramienta de escaneo de puertos para redes grandes. Puede escanear desde máquinas individuales a segmentos de red.
- Ethereal: es un analizador de protocolos y captura el tráfico de la red (es un sniffer).
- Snort: es un sistema IDS que permite realizar análisis de tráfico en tiempo real y guardar logs de los mensajes. Permite realizar análisis de los protocolos, búsquedas por patrones (protocolo, origen, destino, etc.).
- Nessus: es un detector de vulnerabilidades conocidas y asesora sobre cómo mejorar la seguridad para las detectadas.

Productos de seguridad:

- **STD (Security Tools Distribution), LocalAreaSecurity →** ejecutadas desde CD, multitud de herramientas de seguridad, ambas basadas en Knoppix
- **R.I.P. (Recovery Is Posible) Linux→** distribución de Linux pensada por recuperar datos de sistemas de ficheros defectuosos.
- **WARLINUX 0.5:** Esta distribución de Linux, en modo texto, está especialmente pensada para la verificación de la seguridad de las redes inalámbricas
- **FIRE:** Esta versión de Linux incluye las herramientas necesarias para la realización de valoraciones de seguridad, respuesta a incidentes de seguridad.
- **Los TCP wrappers** son un software que actúa de intermediario entre las peticiones del usuario de servicio y los daemons de los servidores que ofrecen el servicio

ADMINISTRACIÓN DE RED

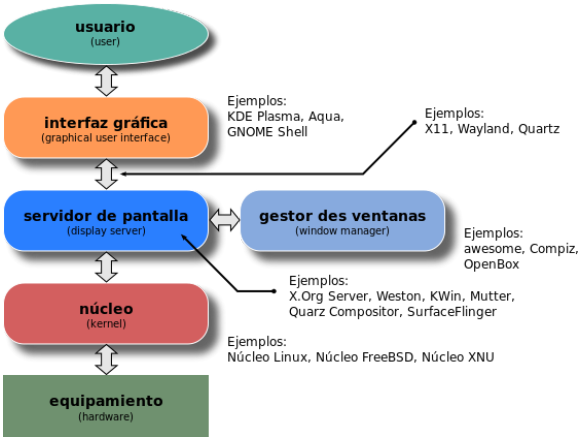
- **NFS (Network File System)⟶** está orientado principalmente hacia el entorno Ethernet. Los equipos servidores exportan una lista de subdirectorios a compartir, lo que incluye toda la jerarquía que tienen por debajo y los clientes conectan esos recursos a sus propias jerarquías haciendo el acceso completamente homogéneo y transparente al usuario.
- **RFS (Remote File Sharing)⟶** está orientado hacia StarLan. Hay que configurar un dominio y la aparición de la máquina en el dominio
- **Soporte de red⟶** El soporte de red está construido sobre tres características:
    - El Sistema de ficheros Virtual (Virtual File System – VFS) permite correspondencia entre cualquier tipo de sistema de ficheros
    - Los flujos (Streams) ⟶ permite la configuración en tiempo de ejecución de los canales de comunicación.
    - Los sockets (conectores) ⟶ son interfases de comunicación abstractas construidas a partir de las capacidades de los streams. Permiten que las aplicaciones se comuniquen

HERRAMIENTAS INTEGRADAS DE ADMINISTRACIÓN: OA&M: Operación, administración y mantenimiento. Está basado en las herramientas FMLI: Intérprete de Lenguajes de formularios y Menús (Form and Menu Language Interpreter).

IBTERFAZ MODO COMANDO: SHELL C, SHELL KORN, SHELL CHS, SHELL BASH

INTERFACES GRÁFICOS

- **Arquitectura X-Windows : versión X11R7.8 :** sistema de gráficos distribuidos basado en el modelo cliente / servidor: Xserver, Xclient. Las aplicaciones gráficas se ejecutan el servidor y los resultados se visualizan en el cliente. Por encima del servidor X está el gestor de ventanas que define la apariencia final.
    - Utiliza **XFree86 (libre)** provee una interfaz gráfica cliente/servidor entre el hardware (sistemas gráficos y dispositivos de entrada, como el mouse o el teclado) y un entorno de escritorio que provee un sistema de ventanas. Es un servidor de pantalla



    - Las capas quedarían: Servidor X – Gestor de ventanas – Gestor de escritorio – Temas
    - **Gestor de ventanas**: se encarga de cómo los programas windows son visualizados en la pantalla. **KDE** utiliza como gestor de ventanas uno propio el KDE Windows Manager. **GNOME** no posee gestor de ventanas propio, pero puede ser usado con otros, siempre y cuando éstos se ajusten a la norma GNOME. El más popular gestor de ventanas que trabaja con GNOME es Enlightenment o Metacity

- **KDE 5 (Julio 2014)** A los diferentes entornos de trabajo a nivel gráfico o wokspaces se les denomina **PLASMA**, como por ejemplo el Plasma Desktop o Plama NoteBook. Eso ha provocado que se popularice el nombre de PLASMA al hablar de KDE. De hecho a los pequeños widgets que es posible programar en KDE se los denomina PLASMAOIDES
    - Utiliza las librerías gráficas **Qt** para mostrar los elementos de la interfaz. Licencia GPL / LGPL (QPL)
    - Permite definir hasta 8 escritorios virtuales (4 por defecto)
    - Subsistemas KDE:
        - **DCOP** Desktop Communication Protocol: Se encarga de la gestión del protocolo de comunicaciones del escritorio.
        - **KIO** Network Transparent I/O: Gestiona las comunicaciones dentro de un entorno de red de forma transparente para el usuario
        - **SYCOCA** SYstem COnfiguration Cache : Se encarga de la configuración de la memoria caché
        - **KParts :** Componentes integrados
        - **KHTML**: librería HTML 4.0
        - **XMLGU:** Arquitectura dinámica GUI basada en XML**.**
        - **Art.**: Sistema Multimedia
    - Elementos KDE

        - Panel (KDE panel for short o también llamado kicker)
        - / Centro de control/ Panel de configuración /Ejecución de programas en un icono / Herramientas del panel
    - Aplicaciones y frameworks
        - Phonon: es el framework multimedia estándar de KDE 4
        - Decibel : es framework de comunicaciones en tiempo.
        - Solid es el framework de tratamiento de dispositivos de KDE 4.
        - Calligra Suite: Integrated office suite.
        - KDEWebdev: Web development tools.
    - Programas: Kmail: Correo electrónico / Kfm: Gestión de ficheros parecido al explorador de Microsoft Windows/ Klix: Procesador de texto

- **GNOME (v 3.12) usa** librerías **GTK**. No tiene gestor de ventanas
    - Los principales componentes de GNOME son: • Gestor de escritorio (visualización y gestión de los iconos visibles del escritorio.) • Panel GNOME (Es la unidad central de control. Proporciona el menú de inicio y los applets que muestran sus salidas en pequeñas áreas del panel)
    - Proyectos GNOME
        - Bonobo: tecnología (obsoleta en las actuales versiones) de arquitectura de programación
        - GConf: almacenamiento de configuración del sistema.
        - GVFS: sistema de archivos virtual.
        - GNOME Keyring: sistema de seguridad.
        - GNOME Print: sistema de impresión de documentos.
        - Gstreamer: el framework o «esqueleto» multimedia para aplicaciones.
        - GTK+: bibliotecas para desarrollar interfaces gráficas de usuario.
            - ATK: bibliotecas para ofrecer accesibilidad, por ejemplo, a personas con alguna discapacidad.
            - Pango: biblioteca para el diseño y renderizado de texto internacional.
            - Cairo: biblioteca de renderización avanzada de controles de aplicación
        - LibXML: biblioteca XML.
        - Orbit: un CORBA ORB para componentes software.
        - Metacity: administrador de ventanas.
        - HIG: investigación y documentación iniciadas por Sun Microsystems para aumentar la usabilidad.
        - Nautilus: administrador de archivos.
        - gucharmap: mapa de caracteres UNICODE.

    - **Xfce** es un entorno de escritorio libre para sistemas tipo Unix como GNU/Linux, BSD, Solaris y derivados. Su objetivo es ser rápido y ligero, sin dejar de ser visualmente atractivo y fácil de usar. También utiliza GTK

ANEXO

Fip: utilidad que dividir el disco duro sin perder información

Si se quiere arrancar Linux si existe Windows, la forma más sencilla es instalar **LILO o GRUB**, un programa que se encarga de arrancar ambos sistemas operativos según lo que indique el usuario al arrancar el PC.

Cuando un sistema GNU/Linux arranca, primero se carga el kernel del sistema, después se inicia el primer proceso, denominado init, que es el responsable de ejecutar y activar el resto del sistema, mediante la gestión de los niveles de ejecución (o **runlevels**)

Runlevel	Función	Descripción
0	Parada	Finaliza servicios y programas activos, así como desmonta *filesystems* activos y para la CPU.
1	Monousuario	Finaliza la mayoría de servicios, permitiendo sólo la entrada del administrador (*root*). Se usa para tareas de mantenimiento y corrección de errores críticos.
2	Multiusuario sin red	No se inician servicios de red, permitiendo sólo entradas locales en el sistema.
3	Multiusuario	Inicia todos los servicios excepto los gráficos asociados a X Window.
4	Multiusuario	No suele usarse, típicamente es igual que el 3.
5	Multiusuario	Igual que 3, pero con soporte X para la entrada de usuarios (*login* gráfico).
6	Reinicio	Para todos los programas y servicios. Reinicia el sistema.

Servidores de impresión: BSD LPD, LPRng, CUPS