

# Introducción a la Administración de Sistemas Unix/Linux

Departamento de Sistemas Telemáticos y Computación (GSyC)

gsvc-profes (arroba) gsvc.es

Septiembre de 2012



©2012 GSyC  
Algunos derechos reservados.  
Este trabajo se distribuye bajo la licencia  
Creative Commons Attribution Share-Alike 3.0

- 1 Un poco de historia de Unix y Linux
  - Partes de un sistema operativo
  - ¿GNU/Linux ó Linux?
- 2 Linux es Software Libre
  - ¿Qué no es software libre?
  - Tipos de licencia
  - Motivos para desarrollar software libre
  - Inconvenientes del software libre para el usuario
  - Ventajas del software libre
- 3 El Núcleo del Sistema Operativo
- 4 Procesos
  - Servicios principales de un sistema UNIX
- 5 Interfaces de usuario

- UNIX surgió en 1969 en los Laboratorios Bell (Ken Thomson, Dennis Ritchie)
- Dos grandes vertientes
  - BSD: SunOS, NetBSD, OpenBSD, Mac OS
  - System V: Solaris, Iris, Aix, Linux (año 1991)  
Distribuciones Linux
    - Slackware
    - Gentoo
    - Suse
    - RedHat y derivados: Fedora, Mandriva (Mandrake)
    - Debian y derivados: Ubuntu, knoppix, GnuLiNex, guadalinux

- Kernel (Núcleo): elemento más importante. Permite que las aplicaciones accedan al hardware. Es responsable de la gestión de recursos, seguridad, etc
- Procesos de usuario: distintos programas ejecutándose concurrentemente en un sistema
- La interacción entre el núcleo y los procesos se hace mediante llamadas al sistema (*system calls*)

La shell es un interfaz de usuario en modo texto. Es una aplicación como otra cualquiera

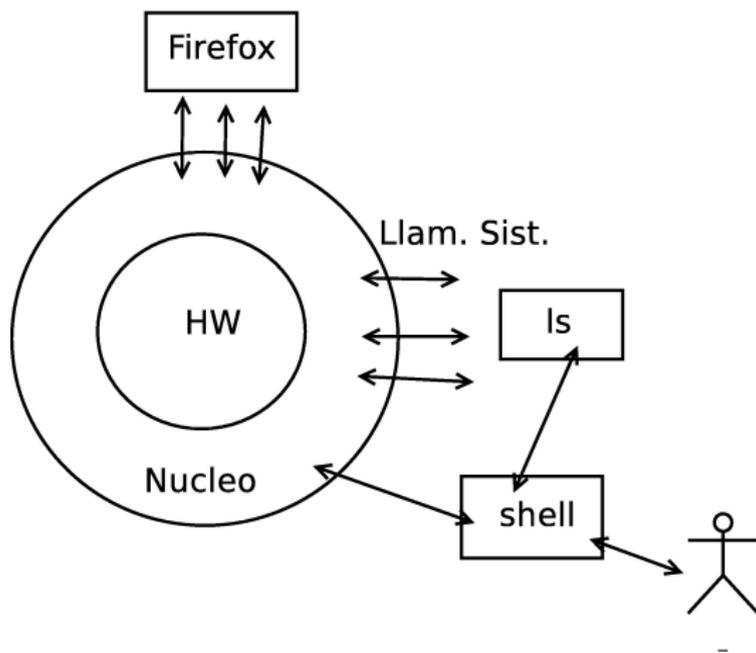


Figura: El Sistema Operativo

- 1 La Free Software Foundation (Richard Stallman) considera que:
  - Linux es estrictamente el kernel
  - Los procesos de usuario (*programas y otras utilidades básicas para el sistema*) provienen del proyecto GNU (y algunos otros).
  - Al conjunto se le debe llamar GNU/Linux.
- 2 Un número importante de personas y organismos se oponen a esta definición. La mayoría de la gente lo llama simplemente Linux

Linux es el *producto estrella* del Soft. Libre

- Hay software libre para cualquier S.O.
- Hay software propietario para Linux

Cuatro libertades. Quien lo recibe tiene:

- libertad de uso. Usarlo como quiera, donde quiera
- libertad de redistribución. Redistribuirlo a quien quiera, como quiera
- libertad de modificación. Modificar, adaptar, corregir, mejorar
- libertad de distribuir las modificaciones

Imprescindible: disponibilidad de código fuente.

- Como cualquier modelo, puede ser criticado
- Pero algunos argumentos en contra habituales no tienen ningún sentido:

*Los médicos, los abogados y los fontaneros no trabajan gratis.  
¿Por qué habrían de hacerlo los programadores?*

- Software gratuito
- Shareware
- Adware
- Versiones de evaluación
- Dominio Público
  
- Minimalistas. Permiten *cerrar* el código. Pj BSD
- *protectoras de la libertad*. GPL.  
Redistribuciones con mismos derechos que la primera distribución

- Ética, satisfacción personal, pertenencia a una comunidad
- Aprendizaje
- Tesis doctorales, PFCs
- Empresas que se dedican a otra cosa
- Organismos públicos
- Empresas que obtienen dinero por servicios
- Empresas de Hardware
- etc etc

- Ninguno

En ciertas ocasiones (cada vez menos) puede ser indicado software propietario:

- Software inexistente o insuficiente
- Hardware no soportado
- Otros. (discutible) *Quien me rodea usa determinado software*

- 4 libertades
- Facilita la reutilización
- Mucho menor coste
- Nadie impone la renovación de Hw, Sw ni formación de usuarios
- Mejor interoperabilidad y escalabilidad
- Garantía de privacidad
- Permite conocer mejor el software y comprobar su calidad
- Igualdad de oportunidades: Mismas herramientas para todos.  
Promoción de economía local

Más información: Estudio FLOSSImpact

- Gestión de procesos
- Gestión de memoria
- Gestión de dispositivos
- Gestión de sistemas de ficheros
- Gestión de red

- Procesos = ejecutables + librerías dinámicas
- Identificadores asociados a cada proceso:
  - PID: Identificación única de cada proceso
  - UID: Identificación de usuario
  - GID: Identificación de grupo (posibilidad de varios grupos por proceso)
- $uid=0 \Rightarrow$  *super-usuario*, "root":
  - Control sobre el resto de procesos
  - Permiso para acceder a todos los ficheros
  - Posibilidad de realizar ciertas tareas privilegiadas

- `init`. Primer proceso, padre de todos los demás. Se encarga de arrancar y parar el sistema.
- Terminales remotas: *login* y *logout*
- `syslog`
- Ejecución periódica de órdenes: `cron` y `at`
- Entorno gráfico (X Window)
- Entorno de red (demonios)
- Correo electrónico, sistema de impresión, ...

# Interfaz gráfico

- Supone un gran avance. Excelente para usuarios, o para tareas que hagamos de vez en cuando
- Mucho menos eficiente: obligan a hacer las cosas *a mano* y de una en una
- Solo se puede hacer lo que el interfaz haya previsto que se haga
- No es la filosofía Unix, no son estándar
- Exigen sesión gráfica (mucho más caro que `ps` `ssh`)
- No siempre disponibles (sistemas empotrados, routers, etc)
- Hay gestores gráficos, pero no serán válidos en esta asignatura

- Unix dispone de interfaz gráfico desde los 80. *X Window*. (No confundir con Microsoft Windows).
  - X Window System es un sistema gráfico utilizado fundamentalmente en sistemas Unix, aunque es multiplataforma
  - Proporciona un mecanismo para mostrar ventanas gráficas basado en dos partes: cliente y servidor
    - Servidor X: Se ejecuta típicamente en la máquina en la que está sentado el usuario.
    - Clientes X: Aplicaciones que producen una salida gráfica que envían al Servidor X para que la presente en pantalla. Pueden ejecutarse en ordenadores remotos.
- Sobre las X Window van el *gestor de ventanas* ( Kwin, Enlightenment, Metacity, Xfwm, MWM...)
- Sobre el gestor de ventanas, va el *escritorio* (KDE, Gnome, Xfce...)

# Interfaz de texto: consola

*Write programs that do one thing and do it well. Write programs to work together. Write programs that handle text streams, because that is a universal.*

- interfaz texto: teclado
  - terminales x
  - consola: terminales virtuales (Ctrl+Alt+F1) (Ctrl+Alt+F6)
  - Vuelta a sesión X (Ctrl+Alt+F7)
- `exit` (EOF, Ctrl + D)

En MS Windows el interfaz de consola para la administración es una opción viable desde la aparición en 2006 de PowerShell

Interfaz de texto en Unix:



Interfaz gráfico en Unix:



¿Qué debería manejar un fotógrafo?