

Profesor: Paulino Ruiz

Lab. de Sistemas Operativos

5º1º - 5º2º

Trabajo practico nº 1

Sistema operativo Linux

Leer y responder los siguientes puntos.

- 1 - ¿Qué es Linux?
- 2 - ¿Por qué utilizar Linux?
- 3 - Características generales de Linux.
- 4 - Historia de Linux.
- 5 - Distribuciones.
- 6 - Kernel y software.

¿Qué es Linux?

Linux es un sistema operativo cuyo creador fue Linus Torvalds a principios de la década de los 90.

El objetivo ha sido crear un sistema operativo similar al UNIX pero de libre distribución y cuyo código fuente fuese abierto pudiendo ser utilizado por cualquier persona.

El sistema Linux surgió como un entretenimiento de Linus Torvalds mientras era estudiante de la Universidad de Helsinki, en Finlandia. Linus comenzó la programación del sistema basándose en un pequeño Unix utilizado en ambientes académicos para enseñanza llamado minix.

El sistema lo diseñan y construyen cientos de programadores dispersos por todo el mundo.

El nombre Linux se refiere estrictamente al núcleo Linux (Kernel) pero se utiliza para referirse a las distribuciones GNU/Linux

¿Por qué utilizar Linux?

Porque es un sistema operativo para redes con la característica de ser multitarea, multiusuario, confiable y escalable. Además trabaja en un amplio rango de hardware (Intel, AMD, Motorola, Sparc, Digital, etc).

Es distribuido bajo licencia pública general de GNU (GNU GPL) y esta disponible en Internet en miles de servidores; también puede adquirirse mediante distribuidores como Red Hat, SuSE, Debian, entre otros.

Esta siendo utilizado por millones de personas alrededor del mundo. Es utilizado para servicios de red (Intranet e Internet), para desarrollos de programas.

Soporta todos los protocolos de red e Internet estándar y se puede personalizar gracias a su naturaleza abierta y la cantidad de herramientas de programación que posee.

Características generales de Linux

Las características principales del Linux son su potencia y flexibilidad. Estas son el resultado de varias características integradas al sistema que están disponibles tan pronto como se inicia.

Multitareas

Esta característica permite que múltiples tareas, también llamadas procesos, sean ejecutados al mismo tiempo compartiendo un recurso como el CPU. Multitarea se refiere a la habilidad de asignar tiempo de procesamiento a varios procesos dando la impresión que se están ejecutando simultáneamente.

El aumento de la velocidad de los CPU no solo se ve reflejado en la velocidad a la que corren las aplicaciones, sino que también el sistema operativo cambia entre aplicaciones más rápidamente produciendo una mejora del rendimiento global.

Tipos de multitarea

Cooperativa

Los procesos ceden la CPU al sistema operativo a intervalos regulares. Es el sistema multitarea más problemático dado que si el proceso de usuario se interrumpe y no cede la CPU al sistema operativo, todo el sistema queda detenido. También produce latencias irregulares y es impracticable si se requiere procesamiento en tiempo real. Por ejemplo se puede citar Windows 95.

Preferente

El sistema operativo se encarga de la administración de CPU, distribuye el tiempo de uso entre los procesos que estén en espera para utilizarlo. Cada proceso utiliza el procesador durante cortos períodos de tiempo siendo el resultado final prácticamente igual que si se estuviesen ejecutándose al mismo tiempo. Por ejemplo se pueden citar Unix, Linux, Windows NT, etc.

Real

Solo para sistemas multiprocesador. Es el caso en el que varios procesos se ejecutan al mismo tiempo en distintos procesadores. Suele también conjugarse y ser preferente. Por ejemplo se pueden citar Unix, Linux, Windows NT, MacOS X, etc.

Single Core

En el caso de un ordenador con un solo core de CPU, sólo se ejecuta un proceso a un determinado momento, lo que significa que la CPU está ejecutando instrucciones solo ese proceso.

La multitarea resuelve este problema organizando que proceso deberá ejecutarse en

un momento dado y cuando una tarea en espera tendrá su turno de ejecución.

Multicore

Si hablamos de un sistema multicore, con multitarea se pueden realmente ejecutar múltiples tareas simultáneamente. Cada core puede trabajar en forma independiente con diferentes procesos.

Por ejemplo en un sistema dualcore podemos ejecutar cuatro aplicaciones como procesador de textos, correo electrónico, navegación web, y software antivirus; cada una de estas aplicaciones podrá acceder a uno de los cores al mismo tiempo. Esto permite realizar varias tareas como chequear e-mail y escribir una carta al mismo tiempo, sin que se vea afectado el rendimiento general.

Multithreading

Multithreading extiende la idea de multitarea dentro de las aplicaciones, por lo que permite subdividir las operaciones de una aplicación en varios hilos o threads. Cada uno de los hilos puede ejecutarse en paralelo. El sistema operativo divide el tiempo de procesamiento, no sólo entre diferentes aplicaciones, sino también entre cada hilo dentro de una aplicación.

Multiusuario

La capacidad para asignar tiempo del CPU a numerosas aplicaciones simultáneas se prestó como consecuencia a servir a múltiples usuarios al mismo tiempo, cada uno ejecutando una o más aplicaciones. Los sistemas multiusuario permiten que más de un usuario utilice la misma aplicación simultáneamente, desde las mismas terminales o desde diferentes terminales. Ejemplos de sistemas operativos con característica de multiusuario son Linux, Unix, Windows NT, Mac OS X.

Multiacceso

Un sistema multiacceso es el que permite a múltiples usuarios desde sus respectivas terminales utilizar el mismo ordenador.

A partir del concepto, el hecho de poder compartir los recursos de procesamiento, almacenamiento y periféricos facilita la reducción de tiempo ocioso en los procesadores, e indirectamente implica la reducción de los costos de energía y equipamiento para resolver las necesidades de cómputo de los usuarios.

Otras características

Multiplataforma: puede ser utilizado en 386, 486. Pentium, Pentium Pro, Pentium II/III, Amiga, Atari, Alpha, ARM, MIPS, PowerPC y SPARC.

Multiprocesador: soporte para sistemas con más de un procesador Intel y SPARC.

Protección de la memoria entre procesos, de manera que uno de ellos no pueda colgar el sistema.

Carga de ejecutables por demanda: Linux sólo lee del disco aquellas partes de un programa que están siendo usadas actualmente.

Política de copia en escritura para poder compartir páginas entre ejecutables: esto significa que varios procesos pueden usar la misma zona de memoria para ejecutarse. Cuando alguno intenta escribir en esa memoria, la página (4Kb de memoria) se copia a otro lugar. Esta política de copia en escritura tiene dos beneficios: aumenta la velocidad y reduce el uso de memoria.

Memoria virtual usando paginación (sin intercambio de procesos completos) a disco: A una partición o un archivo en el sistema de archivos, o ambos, con la posibilidad de

añadir más áreas de intercambio sobre la marcha

La memoria se gestiona como un recurso unificado para los programas de usuario y para el caché de disco, de tal forma que toda la memoria libre puede ser usada para caché y ésta puede a su vez ser reducida cuando se ejecuten grandes programas.

Librerías compartidas de carga dinámica (DLL's) y librerías estáticas.

Se realizan volcados de estado (core dumps) para posibilitar los análisis post-mortem, permitiendo el uso de depuradores sobre los programas no sólo en ejecución sino también tras abortar éstos por cualquier motivo.

Historia de Linux

En 1983 Richard Stallman inició el Proyecto GNU, con el propósito de crear un sistema operativo similar y compatible con UNIX y los estándares POSIX. Dos años más tarde, 1985, creó la Fundación del Software Libre (FSF) y desarrolló la Licencia pública general de GNU (GNU GPL), para difundir el software libremente. De este modo el software de GNU fue desarrollado muy rápidamente, y por muchas personas. A corto plazo, una multiplicidad de programas fueron desarrollados, de modo que a principios de los años 1990 había bastante software disponible para crear un sistema operativo completo. Sin embargo, aun faltaba un kernel.

El kernel debía ser desarrollado en el proyecto GNU Hurd. Pero Hurd demostró desarrollarse muy lentamente, porque el debugging era muy difícil, debido a las características técnicas del diseño del microkernel.

Otro proyecto de sistema operativo libre fue BSD. Este fue desarrollado en la Universidad de Berkeley a partir de la 6ª edición de Unix de AT&T. Puesto que el código de AT&T Unix estaba contenido en BSD, AT&T presentó una demanda a principios de los años 1990 contra la Universidad de Berkeley, la cual limitó el desarrollo de BSD y redujo el desarrollo. Así a principios de los años 1990 no produjo ningún sistema completo. El futuro de BSD era incierto debido al pleito y detuvo el desarrollo. El proyecto GNU se desarrollaba gradualmente pero carecía de un Kernel maduro. Esto dejó un nicho crítico abierto, que Linux llenaría.

En 1985 el profesor Andrew Tanenbaum escribió un sistema operativo tipo Unix desde cero. Llamado MINIX y basado en System V, POSIX y las normas IEEE, fue desarrollado para las PC Intel 386 de los investigadores de ciencias. MINIX también se distribuyó con un popular libro de estudio sobre sistemas operativos del autor.

Un estudiante finlandés llamado Linus Torvald tomó contacto por primera vez con los sistemas Unix a través del uso de MINIX en las clases de Ciencias de la computación en la universidad de Helsinki, Finlandia. Linus Torvald quería mejorar MINIX y agregarle funciones y mejoras, pero Andrew Tanenbaum quería Minix en la forma que había sido creado. Esto motivó a Linus a escribir su propio kernel.

Al principio trabajó en un emulador de terminal, dado que solía tener acceso a los grandes servidores UNIX de la universidad.

Escribió el programa expresamente para el hardware que él usaba, e independiente del sistema operativo, porque quería usar las funciones de su nueva computadora personal con un procesador 80386. El sistema operativo que él usó durante el desarrollo fue Minix, y el compilador inicial fue el GNU C.

Como Torvalds escribió en su libro "Solamente por diversión", tarde o temprano comprendió que había escrito un kernel de sistema operativo. El 25 de agosto de 1991, 20:57:08 GMT, anunció este sistema en un envío a la red Usenet, en el newsgroup (grupo de noticias): comp.os.minix.

El nombre proviene de Linus' Minix. A su juicio, el nombre era demasiado egoísta y lo cambió por Freax (free + freak + x). Un amigo lo alentó a subirlo a Internet para que pudiera ser fácilmente descargado. El mencionado amigo creó un directorio en un servidor FTP llamado "linux", dado que el nombre Freax no le gustó. Como a Linus le agradó el nombre del directorio decidió mantener el nombre del nuevo sistema

operativo en Linux.

Linux bajo la GNU GPL

La primera licencia de Linux fue propietaria, era casi una licencia de código fuente compartida con restricción contra la actividad comercial. En 1992, Linus sugirió cambiar a la GNU GPL. Primero anunció este cambio en las notas de lanzamiento de la versión 0.12. A mediados de diciembre de 1992 publicó la versión 0.99 usando la GNU GPL. Más tarde, Torvalds dijo en una entrevista, "registrar a Linux bajo la GPL fue definitivamente la mejor cosa que alguna vez hice".

GNU/Linux

La designación "Linux" al principio fue usada por Torvalds sólo para el kernel. Este fue sin embargo con frecuencia usado junto con otro software, especialmente con el del proyecto de GNU. Esta variante de GNU se hizo la más popular, ya que no había ningún otro kernel libre que funcionara. Cuando la gente comenzó a referirse hacia esta recopilación como "Linux", Richard Stallman, el fundador del proyecto de GNU, solicitó que el nombre GNU/Linux fuera usado, para reconocer el rol del software de GNU.

En junio de 1994, en el boletín de GNU, Linux fue mencionado como un "clon libre de UNIX", y el Proyecto Debian comenzó a llamar a su producto GNU/Linux. En mayo de 1991 Aug25.205708.9541@klaava.helsinki.

1996, Richard Stallman publicó al editor Emacs 19.31, en el cual el tipo de sistema fue renombrado de Linux a Linux. Esta "escritura" fue pretendida para referirse expresamente a la combinación de GNU y Linux, pero esto pronto fue abandonado en favor de "GNU/Linux".

Este nombre recoge variadas reacciones. Mientras los proyectos GNU y Debian aceptaron el nombre, la mayor parte de desarrolladores y otras distribuciones Linux lo rechazaron. Esto fue justificado por un lado por comodidad, porque el nombre Linux fue considerado como más simple, y por otra parte, el hecho que una cantidad considerable de software que no es de GNU es liberada junto con Linux-GNU/Linux.

Distribuciones

Linux es un sistema de libre distribución por lo que se puede encontrar todos lo necesario para su funcionamiento en Internet.

El trabajo de reunir todo, el sistema operativo mas las aplicaciones necesarias para tener un sistema completo, así como instalarlos y configurarlos puede ser una tarea ardua y no apta para muchos. Para solucionar esto nacen las distribuciones de Linux, empresas y organizaciones que se dedican a recopilar y preconfigurar el sistema aliviando así nuestra parte del trabajo.

Una distribución es una recopilación de Linux con el agregado de programas y archivos, organizados y listos para su instalación.

Las distribuciones más populares de Linux

RedHat Enterprise

Se fundó en 1994 con los emprendedores: Bob Yong y Marc Ewing. Su modelo de negocios se basa en el código abierto. Las soluciones de Red Hat combinan Red Hat Linux, tecnologías de desarrollo, entrenamiento, servicios de gerencia, soporte técnico. Enfocada a empresas. Tiene muy buena calidad, contenidos y soporte a los usuarios por parte de la empresa que la distribuye. Es necesario el pago de una licencia de soporte.

Fedora

Esta es una distribución patrocinada por RedHat y soportada por la comunidad Linux. Fácil de instalar y buena calidad. El equipo de ingenieros de Red Hat participa activamente en el desarrollo de Fedora.

Debian

Otra distribución con muy buena calidad. Privilegian la estabilidad a los últimos avances. El proyecto Debian es una asociación de individuos que tienen como causa común crear un sistema operativo libre. Debian viene con más de 20000 paquetes libres.

OpenSuSE

Otra de las grandes distribuciones. Es fácil de instalar. Es la versión libre de la distribución comercial SuSE. Es uno de los proveedores más antiguos de soluciones Linux.

Slackware

Esta distribución es de las primeras. Slackware Linux por Patrick Volkerding es un sistema operativo Linux avanzado, diseñado con los objetivos de facilidad de uso y estabilidad como prioridades.

Proporciona a los usuarios con experiencia y novatos un sistema completamente equipado para funcionar como una estación de trabajo o como un servidor.

Gentoo

Esta distribución es una de las únicas que han incorporado un concepto totalmente nuevo en Linux. Es un sistema inspirado en BSDports.

Permite compilar/optimizar el sistema completamente desde cero. No es recomendable para usuarios sin conocimientos previos.

Ubuntu

Distribución basada en Debian, centrada en el usuario final y la facilidad de uso. Muy popular y con mucho soporte en la comunidad. El entorno de escritorio por defecto es GNOME.

Kubuntu

Distribución basada en Ubuntu, centrada en el usuario final y la facilidad de uso. La gran diferencia con Ubuntu es que el entorno de escritorio por defecto es KDE.

Mandriva

Esta distribución fue creada en 1998 con el objetivo de acercar el uso de Linux a todos los usuarios, en un principio se llamo Mandrake Linux. Facilidad de uso.

Distribuciones Argentinas

DNALinux

DNALinux es una distribución tipo live Linux basada en SLAX con el agregado de aplicaciones para bioinformática como EMBOSS, Primer3, y otros.

Kwort

Kwort es una distribución moderna y facil de usar basada en slackware, usa GTK+ toolkit y el escritorio Xfce. La característica más importante es el package manager, llamado kpkg, para la descarga de paquetes desde los servidores.

LiveCD Router

Linux LiveCD Router es una distribución diseñada para compartir una conexión broadband mediante WiFi. Puede además ser utilizada como firewall o access point. Si bien no requiere instalación en necesario que la maquina sea dedicada.

Loco

Linux Loco es una distribución basada en Debian. Ofrece una instalación simple, seguridad y un escritorio actualizado. El objetivo principal de la distribución es la de crear conciencia del software libre en ambiente públicos y privados.

Musix

Musix esta basado en KNOPPIX y tiene una gran colección de software de audio gratuito. Es un sistema operativo multimedia libre destinado a músicos, sonidistas, cineastas y usuarios en general.

Tuquito

Tuquito es un liveCD basado en Debian. Tiene muy buen soporte de scanners, web cams y camaras digitales. Compatibilidad con documentos en formato MS Office. Fue diseñado para usuarios principiantes e intermedios de Linux.

Ututo

Ututo es una distribución desarrollada por Diego Saravia de la Universidad Nacional de Salta, esta basada en Gentoo. El Proyecto UTUTO es un proyecto de investigación y desarrollo de tecnología informática de aplicación social. El Proyecto UTUTO lleva adelante varios subproyectos, de los cuales el más conocido es la distribución del sistema operativo GNU denominada UTUTO XS.

Kernel y software

El kernel se puede definir como el sistema nervioso. Entre otras cosas se encarga de manejar las relaciones entre el hardware y el software.

Las funciones más relevantes del mismo son:

Administración del tiempo de procesador

Administración de la memoria

Control de periféricos

Sincronización entre procesos

Linux fue desarrollado siguiendo el diseño de un Kernel monolítico y no como un microkernel (diseño que era considerado el mejor en aquella época). La decisión conlleva a una serie de intercambios y discusiones entre Linus Torvalds y Andrew Tanenbaum.

Actualmente Linux es un núcleo monolítico con la característica de ser modular. Los controladores de dispositivos y las extensiones del kernel se ejecutan en el ring 0 (espacio privilegiado de memoria), con acceso irrestricto al hardware.

A diferencia de los núcleos monolíticos tradicionales, los controladores de dispositivos y las extensiones al sistema operativo se pueden cargar y descargar como módulos, mientras el sistema continúa funcionando sin interrupciones. Los controladores también pueden ser prevcados (o sea detenidos para dar prioridad a una tarea de mayor importancia).

Núcleo monolítico

Como ejemplo de sistema operativo de núcleo monolítico pueden citarse UNIX y Linux. En estos el Kernel es de gran tamaño y alta complejidad dado que abarca todos los servicios del sistema. Todas las funcionalidades posibles (planificación, sistema de archivos, redes, controladores de dispositivos, gestión de memoria, etc.) están contenidas dentro de este programa de gran tamaño. Tiene un rendimiento mayor que un micronúcleo. La desventaja es que cualquier cambio o agregado requiere la recompilación del kernel y el reinicio.

Normalmente este tipo de kernels surgen de sistemas operativos primitivos, sencillos y pequeños a los que se les ha ido agregando funcionalidades. Esta evolución implica el aumento de líneas de código y trae aparejada la complicación que implica agregar nuevas funcionalidades y servicios.

La estructura de kernel monolítico cuenta con ramificaciones, entre las cuales se encuentra la que permite modular el sistema. El sistema de módulos agrega ventajas de que poseen los microkernels. Los módulos pueden ser modificados, cargados y descargados en tiempo de ejecución, de manera similar a los servicios de un microkernel; la diferencia es que se ejecutan en el ring 0. La desventaja de esto es que si un modulo se bloquea todo el kernel podría quedar fuera de operación.

Micronúcleo

El microkernel provee un conjunto de primitivas o llamadas al sistema mínimas, implementando los servicios básicos. El resto de los servicios se ejecutan como procesos servidores en espacio de usuario.

Las principales ventajas de su utilización son la reducción de la complejidad del kernel, la descentralización de los fallos y el una mejora en el debugging para la creación y mejora de dispositivos.

Mandar las respuestas a pauliruib944@gmail.com