

Informática II

Introducción al sistema operativo GNU/Linux

Gonzalo F. Perez Paina



Universidad Tecnológica Nacional
Facultad Regional Córdoba
UTN-FRC

– 2019 –

Sistemas operativos: historia y contexto

- ▶ ¿Qué es un sistema operativo?

Sistemas operativos: historia y contexto

- ▶ ¿Qué es un sistema operativo?
 - ▶ Programa o conjunto de programas para **administrar** los recursos de hardware y dar **servicios** a los programas de aplicación (software)

Sistemas operativos: historia y contexto

- ▶ ¿Qué es un sistema operativo?
 - ▶ Programa o conjunto de programas para **administrar** los recursos de hardware y dar **servicios** a los programas de aplicación (software)
 - ▶ Administración: tareas (scheduler), memoria, red, seguridad, disco.

Sistemas operativos: historia y contexto

- ▶ ¿Qué es un sistema operativo?
 - ▶ Programa o conjunto de programas para **administrar** los recursos de hardware y dar **servicios** a los programas de aplicación (software)
 - ▶ Administración: tareas (scheduler), memoria, red, seguridad, disco.
- ▶ ¿Cuales conocen?

Sistemas operativos: historia y contexto

- ▶ ¿Qué es un sistema operativo?
 - ▶ Programa o conjunto de programas para **administrar** los recursos de hardware y dar **servicios** a los programas de aplicación (software)
 - ▶ Administración: tareas (scheduler), memoria, red, seguridad, disco.
- ▶ ¿Cuales conocen?
 - ▶ Windows

Sistemas operativos: historia y contexto

- ▶ ¿Qué es un sistema operativo?
 - ▶ Programa o conjunto de programas para **administrar** los recursos de hardware y dar **servicios** a los programas de aplicación (software)
 - ▶ Administración: tareas (scheduler), memoria, red, seguridad, disco.
- ▶ ¿Cuales conocen?
 - ▶ Windows
 - ▶ Android

Sistemas operativos: historia y contexto

- ▶ ¿Qué es un sistema operativo?
 - ▶ Programa o conjunto de programas para **administrar** los recursos de hardware y dar **servicios** a los programas de aplicación (software)
 - ▶ Administración: tareas (scheduler), memoria, red, seguridad, disco.
- ▶ ¿Cuales conocen?
 - ▶ Windows
 - ▶ Android
 - ▶ Unix

Sistemas operativos: historia y contexto

- ▶ ¿Qué es un sistema operativo?
 - ▶ Programa o conjunto de programas para **administrar** los recursos de hardware y dar **servicios** a los programas de aplicación (software)
 - ▶ Administración: tareas (scheduler), memoria, red, seguridad, disco.
 - ▶ ¿Cuales conocen?
 - ▶ Windows
 - ▶ Android
 - ▶ Unix
 - ▶ Mac OS
-

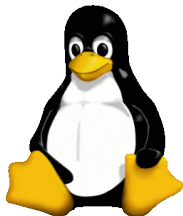
Sistemas operativos: historia y contexto

- ▶ ¿Qué es un sistema operativo?
 - ▶ Programa o conjunto de programas para **administrar** los recursos de hardware y dar **servicios** a los programas de aplicación (software)
 - ▶ Administración: tareas (scheduler), memoria, red, seguridad, disco.
- ▶ ¿Cuales conocen?
 - ▶ Windows
 - ▶ Android
 - ▶ Unix
 - ▶ Mac OS
 - ▶ BSD¹

¹Berkeley Software Distribution, desarrollado desde 1970 (Univ. de California)

Sistemas operativos: historia y contexto

- ▶ ¿Qué es un sistema operativo?
 - ▶ Programa o conjunto de programas para **administrar** los recursos de hardware y dar **servicios** a los programas de aplicación (software)
 - ▶ Administración: tareas (scheduler), memoria, red, seguridad, disco.
- ▶ ¿Cuales conocen?
 - ▶ Windows
 - ▶ Android
 - ▶ Unix
 - ▶ Mac OS
 - ▶ BSD¹
 - ▶ GNU/Linux²



¹Berkeley Software Distribution, desarrollado desde 1970 (Univ. de California)

²GNU: "GNU's Not Unix (clon de Unix), bajo licencia GPL"

Richard Stallman

Richard Matthew Stallman (nacido en [Manhattan](#), Nueva York, 16 de marzo de 1953), con frecuencia abreviado como «[rms](#)»,¹ es un [programador estadounidense](#) y fundador del [movimiento por el software libre](#) en el mundo.

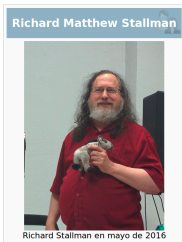
Entre sus logros destacados como programador se incluye la realización del editor de texto [GNU Emacs](#),² el compilador [GCC](#),³ y el depurador [GDB](#),⁴ bajo la rúbrica del Proyecto GNU. Sin embargo, es principalmente conocido por el establecimiento de un marco de referencia [moral, político y legal](#) para el movimiento del [software libre](#), como una alternativa al desarrollo y distribución del [software no libre](#) o privativo. Es también [inventor](#) del concepto de [copyleft](#) (aunque no del término), un método para licenciar software de tal forma que su uso y modificación permanezcan siempre libres y queden en la comunidad de usuarios y desarrolladores.

Índice [\[ocultar\]](#)

1 Biografía

1.1 Primeros años

1.2 Laboratorio de inteligencia artificial del MIT



Richard Stallman inició el proyecto GNU en enero de 1984

Linus Torvalds

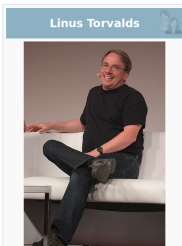
Linus Benedict Torvalds (28 de diciembre de 1969, Helsinki, Finlandia) es un [ingeniero de software](#) finés americano,¹ conocido por iniciar y mantener el desarrollo del "[kernel](#)" (en español, [núcleo](#)) [Linux](#), basándose en el sistema operativo libre [Minix](#) creado por [Andrew S. Tanenbaum](#) y en algunas herramientas, varias utilidades y los [compiladores](#) desarrollados por el [proyecto GNU](#).

Actualmente Torvalds es responsable de la coordinación del proyecto. Pertenece a la comunidad [sueco-parlante](#) de [Finlandia](#).

Sus padres tomaron su nombre de [Linus Pauling](#) (estadounidense, [Premio Nobel de Química 1954](#)). Comenzó sus andanzas informáticas a los 11 años cuando su abuelo, un [matemático](#) y [estadístico](#) de la Universidad, compró uno de los primeros microordenadores [Commodore](#) en 1980 y le pidió ayuda para usarlo.²

A finales de los [años 80](#) tomó contacto con los ordenadores [IBM](#), [PC](#) y en 1991 adquirió un ordenador con procesador modelo 80386 de [Intel](#).

En 1988 fue admitido en la [Universidad de Helsinki](#), donde estudio [Ciencias de la Computación](#). Ese mismo año el profesor [Andrew S. Tanenbaum](#) saca a la luz el S.O. [Minix](#) con propósitos didácticos. Dos años después, en 1990, Torvalds



Linus Torvalds liberó el código del Kernel de Linux en agosto de 1991 (PC 386)

From: torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds)
Newsgroups: comp.os.minix
Subject: What would you like to see most in minix?
Summary: small poll for my new operating system
Message-ID:
Date: 25 Aug 91 20:57:08 GMT
Organization: University of Helsinki

Hello everybody out there using minix -

I'm doing a (free) operating system (just a hobby, won't be big and professional like gnu) for 386(486) AT clones. This has been brewing since april, and is starting to get ready. I'd like any feedback on things people like/dislike in minix, as my OS resembles it somewhat (same physical layout of the file-system (due to practical reasons) among other things).

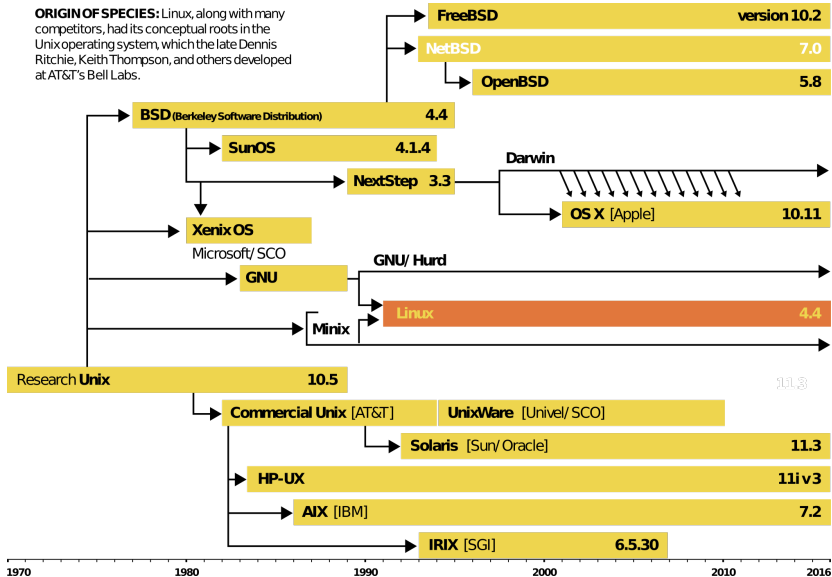
I've currently ported bash(1.08) and gcc(1.40), and things seem to work. This implies that I'll get something practical within a few months, and I'd like to know what features most people would want. Any suggestions are welcome, but I won't promise I'll implement them :-)

Linus (torvalds@kruuna.helsinki.fi)

PS. Yes - it's free of any minix code, and it has a multi-threaded fs. It is NOT protable (uses 386 task switching etc), and it probably never will support anything other than AT-harddisks, as that's all I have :-).

GNU/Linux

ORIGIN OF SPECIES: Linux, along with many competitors, had its conceptual roots in the Unix operating system, which the late Dennis Ritchie, Keith Thompson, and others developed at AT&T's Bell Labs.



De *IEEE Spectrum*, abril de 2016

GNU/Linux

- ▶ Linux 1.0 lanzado en 1994

GNU/Linux

- ▶ Linux 1.0 lanzado en 1994
- ▶ Desarrollado a partir de Minix (1987), usado por Linus (BSD no corría en PC, y GNU Hurd no estaba listo)

GNU/Linux

- ▶ Linux 1.0 lanzado en 1994
- ▶ Desarrollado a partir de Minix (1987), usado por Linus (BSD no corría en PC, y GNU Hurd no estaba listo)
- ▶ Sigue el estándar POSIX (Portable Operating System Interface for Unix): estándar para sistemas operativo tipo Unix (IEEE-CS)

GNU/Linux

- ▶ Linux 1.0 lanzado en 1994
- ▶ Desarrollado a partir de Minix (1987), usado por Linus (BSD no corría en PC, y GNU Hurd no estaba listo)
- ▶ Sigue el estándar POSIX (Portable Operating System Interface for Unix): estándar para sistemas operativo tipo Unix (IEEE-CS)
- ▶ Propiedades importantes de Linux
 1. Estabilidad
 2. Seguro
 3. No necesita rebuteo frecuente
 4. Portabilidad & Escalabilidad, etc.

GNU/Linux

- ▶ Linux 1.0 lanzado en 1994
- ▶ Desarrollado a partir de Minix (1987), usado por Linus (BSD no corría en PC, y GNU Hurd no estaba listo)
- ▶ Sigue el estándar POSIX (Portable Operating System Interface for Unix): estándar para sistemas operativo tipo Unix (IEEE-CS)
- ▶ Propiedades importantes de Linux
 1. Estabilidad
 2. Seguro
 3. No necesita rebuteo frecuente
 4. Portabilidad & Escalabilidad, etc.
- ▶ GNU (copyleft): Licenciado bajo GPL (General Public License)

GNU/Linux

- ▶ Linux 1.0 lanzado en 1994
- ▶ Desarrollado a partir de Minix (1987), usado por Linus (BSD no corría en PC, y GNU Hurd no estaba listo)
- ▶ Sigue el estándar POSIX (Portable Operating System Interface for Unix): estándar para sistemas operativo tipo Unix (IEEE-CS)
- ▶ Propiedades importantes de Linux
 1. Estabilidad
 2. Seguro
 3. No necesita rebuteo frecuente
 4. Portabilidad & Escalabilidad, etc.
- ▶ GNU (copyleft): Licenciado bajo GPL (General Public License)
- ▶ Distribuciones (Linux flavors)

Software libre – GNU/Linux

- ▶ RMS: en 1983 inicia oficialmente el proyecto GNU, en 1985 crea la Free Software Foundation, luego redacta la licencia GPL

Software libre – GNU/Linux

- ▶ RMS: en 1983 inicia oficialmente el proyecto GNU, en 1985 crea la Free Software Foundation, luego redacta la licencia GPL
- ▶ Software libre vs. privativo. Software libre vs. gratis

Software libre – GNU/Linux

- ▶ RMS: en 1983 inicia oficialmente el proyecto GNU, en 1985 crea la Free Software Foundation, luego redacta la licencia GPL
- ▶ Software libre vs. privativo. Software libre vs. gratis

Resumen GPL

- ▶ Cualquiera es libre de utilizar el Software Libre para cualquier propósito.

Software libre – GNU/Linux

- ▶ RMS: en 1983 inicia oficialmente el proyecto GNU, en 1985 crea la Free Software Foundation, luego redacta la licencia GPL
- ▶ Software libre vs. privativo. Software libre vs. gratis

Resumen GPL

- ▶ Cualquiera es libre de utilizar el Software Libre para cualquier propósito.
- ▶ Cualquiera es libre de acceder a su código fuente y estudiarlo.

Software libre – GNU/Linux

- ▶ RMS: en 1983 inicia oficialmente el proyecto GNU, en 1985 crea la Free Software Foundation, luego redacta la licencia GPL
- ▶ Software libre vs. privativo. Software libre vs. gratis

Resumen GPL

- ▶ Cualquiera es libre de utilizar el Software Libre para cualquier propósito.
- ▶ Cualquiera es libre de acceder a su código fuente y estudiarlo.
- ▶ Cualquiera es libre de distribuirlo.

Software libre – GNU/Linux

- ▶ RMS: en 1983 inicia oficialmente el proyecto GNU, en 1985 crea la Free Software Foundation, luego redacta la licencia GPL
- ▶ Software libre vs. privativo. Software libre vs. gratis

Resumen GPL

- ▶ Cualquiera es libre de utilizar el Software Libre para cualquier propósito.
- ▶ Cualquiera es libre de acceder a su código fuente y estudiarlo.
- ▶ Cualquiera es libre de distribuirlo.
- ▶ Cualquiera es libre de mejorarlo o adaptarlo y de distribuir el programa modificado.

Software libre – GNU/Linux

- ▶ RMS: en 1983 inicia oficialmente el proyecto GNU, en 1985 crea la Free Software Foundation, luego redacta la licencia GPL
- ▶ Software libre vs. privativo. Software libre vs. gratis

Resumen GPL

- ▶ Cualquiera es libre de utilizar el Software Libre para cualquier propósito.
- ▶ Cualquiera es libre de acceder a su código fuente y estudiarlo.
- ▶ Cualquiera es libre de distribuirlo.
- ▶ Cualquiera es libre de mejorarlo o adaptarlo y de distribuir el programa modificado.
- ▶ La única obligación es que si se distribuye, haya que hacerlo bajo la misma licencia GPL.

Software libre – GNU/Linux

- ▶ RMS: en 1983 inicia oficialmente el proyecto GNU, en 1985 crea la Free Software Foundation, luego redacta la licencia GPL
- ▶ Software libre vs. privativo. Software libre vs. gratis

Resumen GPL

- ▶ Cualquiera es libre de utilizar el Software Libre para cualquier propósito.
 - ▶ Cualquiera es libre de acceder a su código fuente y estudiarlo.
 - ▶ Cualquiera es libre de distribuirlo.
 - ▶ Cualquiera es libre de mejorarlo o adaptarlo y de distribuir el programa modificado.
 - ▶ La única obligación es que si se distribuye, haya que hacerlo bajo la misma licencia GPL.
-
- ▶ **FOSS:** Free and Open Source Software

Software libre – GNU/Linux

- ▶ RMS: en 1983 inicia oficialmente el proyecto GNU, en 1985 crea la Free Software Foundation, luego redacta la licencia GPL
- ▶ Software libre vs. privativo. Software libre vs. gratis

Resumen GPL

- ▶ Cualquiera es libre de utilizar el Software Libre para cualquier propósito.
 - ▶ Cualquiera es libre de acceder a su código fuente y estudiarlo.
 - ▶ Cualquiera es libre de distribuirlo.
 - ▶ Cualquiera es libre de mejorarlo o adaptarlo y de distribuir el programa modificado.
 - ▶ La única obligación es que si se distribuye, haya que hacerlo bajo la misma licencia GPL.
-
- ▶ **FOSS**: Free and Open Source Software
 - ▶ **OSHW**: Open Source Hardware

Distribuciones – versiones/flavors

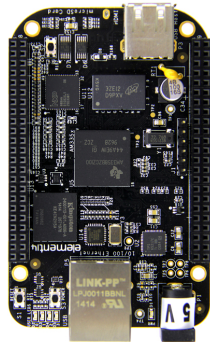
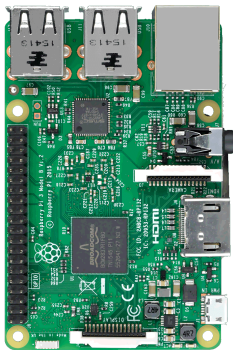
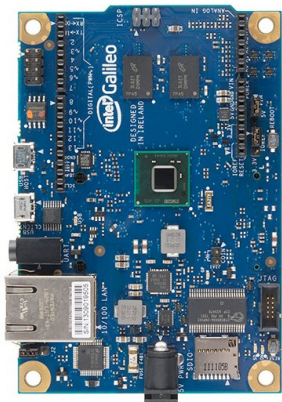


Distribuciones – versiones/flavors

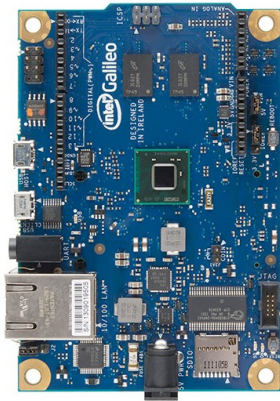


Linux Distribution Timeline

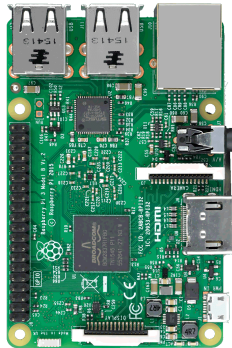
SBC, Single Board Computer



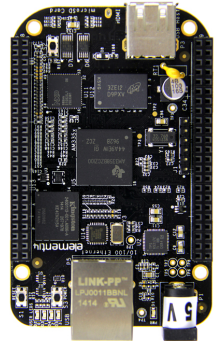
SBC, Single Board Computer



Intel Galileo



RaspberryPi 3



Beaglebone Black

SO GNU/Linux – Componentes

Componentes de un sistema GNU/Linux:

1. El kernel/núcleo (Linux)

SO GNU/Linux – Componentes

Componentes de un sistema GNU/Linux:

1. El kernel/núcleo (Linux)
2. El shell o intérprete de comandos
(`bash`, `ash`, `csch`, etc.)

SO GNU/Linux – Componentes

Componentes de un sistema GNU/Linux:

1. El kernel/núcleo (Linux)
2. El shell o intérprete de comandos
(`bash`, `ash`, `csch`, etc.)
3. Procesos y archivos/sistema de
archivos (File System)

SO GNU/Linux – Componentes

Componentes de un sistema GNU/Linux:

1. El kernel/núcleo (Linux)
2. El shell o intérprete de comandos
(`bash`, `ash`, `csch`, etc.)
3. Procesos y archivos/sistema de
archivos (File System)
4. Entorno de escritorio

SO GNU/Linux – Componentes

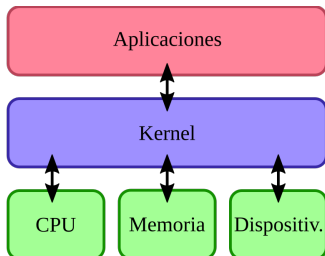
Componentes de un sistema GNU/Linux:

1. El kernel/núcleo (Linux)
2. El shell o intérprete de comandos
(`bash`, `ash`, `csch`, etc.)
3. Procesos y archivos/sistema de
archivos (File System)
4. Entorno de escritorio
5. Aplicaciones

SO GNU/Linux – Componentes

Componentes de un sistema GNU/Linux:

1. El kernel/núcleo (Linux)
2. El shell o intérprete de comandos (`bash`, `ash`, `csch`, etc.)
3. Procesos y archivos/sistema de archivos (File System)
4. Entorno de escritorio
5. Aplicaciones



Kernel de Linux

- ▶ El Kernel se comunica con el hardware

Kernel de Linux

- ▶ El Kernel se comunica con el hardware
- ▶ No tiene ninguna vinculación con el usuario

Kernel de Linux

- ▶ El Kernel se comunica con el hardware
- ▶ No tiene ninguna vinculación con el usuario
- ▶ Coordina las funciones internas y administra los recursos del sistema

Kernel de Linux

- ▶ El Kernel se comunica con el hardware
- ▶ No tiene ninguna vinculación con el usuario
- ▶ Coordina las funciones internas y administra los recursos del sistema

Funciones principales del Kernel

- ▶ Administración de procesos mediante la planificación del tiempo de corrida de cada uno y los privilegios que tienen (administración del microprocesador y memoria)

Kernel de Linux

- ▶ El Kernel se comunica con el hardware
- ▶ No tiene ninguna vinculación con el usuario
- ▶ Coordina las funciones internas y administra los recursos del sistema

Funciones principales del Kernel

- ▶ Administración de procesos mediante la planificación del tiempo de corrida de cada uno y los privilegios que tienen (administración del microprocesador y memoria)
- ▶ Administración el uso de dispositivos de hardware a través de los controladores (drivers) necesarios para su funcionamiento

Kernel de Linux

Está escrito en lenguaje C (`gcc`) con algunas pequeñas secciones de código en ensamblador.

Kernel de Linux

Está escrito en lenguaje C (`gcc`) con algunas pequeñas secciones de código en ensamblador.

Algunas arquitecturas de $\mu\text{P}/\mu\text{C}$ soportadas:

- ▶ x86: IA-32 y x86-64
- ▶ ARM: Freescale i.MX, gumstix, Qualcomm Snapdragon
- ▶ MIPS: PlayStation2, Broadcom Wireless
- ▶ PowerPC: PlayStation3, Nintendo
- ▶ SPARC
- ▶ RISC-V
- ▶ etc.

Kernel de Linux – versión

Identificación para la versión del Kernel

1. Antes de la versión 1.0

Kernel de Linux – versión

Identificación para la versión del Kernel

1. Antes de la versión 1.0

- ▶ Primer versión: 0.01, seguido por la 0.02, 0.03, 0.10, 0.11, y 0.12 (primer versión GPL), etc. hasta la versión 1.0

Kernel de Linux – versión

Identificación para la versión del Kernel

1. Antes de la versión 1.0

- ▶ Primer versión: 0.01, seguido por la 0.02, 0.03, 0.10, 0.11, y 0.12 (primer versión GPL), etc. hasta la versión 1.0

2. Luego de la versión 1.0 y antes de las 2.6:

Kernel de Linux – versión

Identificación para la versión del Kernel

1. Antes de la versión 1.0

- ▶ Primer versión: 0.01, seguido por la 0.02, 0.03, 0.10, 0.11, y 0.12 (primer versión GPL), etc. hasta la versión 1.0

2. Luego de la versión 1.0 y antes de las 2.6:

- ▶ La numeración tenía la forma a.b.c, donde: a era la versión del Kernel, b eran las revisiones principales y c las revisiones menores.

Kernel de Linux – versión

Identificación para la versión del Kernel

1. Antes de la versión 1.0

- ▶ Primer versión: 0.01, seguido por la 0.02, 0.03, 0.10, 0.11, y 0.12 (primer versión GPL), etc. hasta la versión 1.0

2. Luego de la versión 1.0 y antes de las 2.6:

- ▶ La numeración tenía la forma **a.b.c**, donde: **a** era la versión del Kernel, **b** eran las revisiones principales y **c** las revisiones menores.
- ▶ En algunos casos se agregó un cuarto número (**a.b.c.d**), también la variante RC (release candidate)

Kernel de Linux – versión

Identificación para la versión del Kernel

1. Antes de la versión 1.0

- ▶ Primer versión: 0.01, seguido por la 0.02, 0.03, 0.10, 0.11, y 0.12 (primer versión GPL), etc. hasta la versión 1.0

2. Luego de la versión 1.0 y antes de las 2.6:

- ▶ La numeración tenía la forma **a.b.c**, donde: **a** era la versión del Kernel, **b** eran las revisiones principales y **c** las revisiones menores.
- ▶ En algunos casos se agregó un cuarto número (**a.b.c.d**), también la variante RC (release candidate)

3. A partir de la versión 2.6.0 (2004) comenzó una discusión de un nuevo mecanismo de numeración para períodos de releases más cortos.

Kernel de Linux – versión

Identificación para la versión del Kernel

1. Antes de la versión 1.0

- ▶ Primer versión: 0.01, seguido por la 0.02, 0.03, 0.10, 0.11, y 0.12 (primer versión GPL), etc. hasta la versión 1.0

2. Luego de la versión 1.0 y antes de las 2.6:

- ▶ La numeración tenía la forma **a.b.c**, donde: **a** era la versión del Kernel, **b** eran las revisiones principales y **c** las revisiones menores.
- ▶ En algunos casos se agregó un cuarto número (**a.b.c.d**), también la variante RC (release candidate)

3. A partir de la versión 2.6.0 (2004) comenzó una discusión de un nuevo mecanismo de numeración para períodos de releases más cortos.

- ▶ En 2011 se liberó la versión 3.0 (2.6.39)

Kernel de Linux – versión

Identificación para la versión del Kernel

1. Antes de la versión 1.0

- ▶ Primer versión: 0.01, seguido por la 0.02, 0.03, 0.10, 0.11, y 0.12 (primer versión GPL), etc. hasta la versión 1.0

2. Luego de la versión 1.0 y antes de las 2.6:

- ▶ La numeración tenía la forma $a.b.c$, donde: a era la versión del Kernel, b eran las revisiones principales y c las revisiones menores.
- ▶ En algunos casos se agregó un cuarto número ($a.b.c.d$), también la variante RC (release candidate)

3. A partir de la versión 2.6.0 (2004) comenzó una discusión de un nuevo mecanismo de numeración para períodos de releases más cortos.

- ▶ En 2011 se liberó la versión 3.0 (2.6.39)
- ▶ Sistema de versionado $x.y.z$

Kernel de Linux – versión

Identificación para la versión del Kernel

1. Antes de la versión 1.0

- ▶ Primer versión: 0.01, seguido por la 0.02, 0.03, 0.10, 0.11, y 0.12 (primer versión GPL), etc. hasta la versión 1.0

2. Luego de la versión 1.0 y antes de las 2.6:

- ▶ La numeración tenía la forma $a.b.c$, donde: a era la versión del Kernel, b eran las revisiones principales y c las revisiones menores.
- ▶ En algunos casos se agregó un cuarto número ($a.b.c.d$), también la variante RC (release candidate)

3. A partir de la versión 2.6.0 (2004) comenzó una discusión de un nuevo mecanismo de numeración para períodos de releases más cortos.

- ▶ En 2011 se liberó la versión 3.0 (2.6.39)
- ▶ Sistema de versionado $x.y.z$
- ▶ En enero de 2019 se anunció la versión 5.0 (de la 4.20)

The Linux Kernel Archives

[About](#)[Contact us](#)[FAQ](#)[Releases](#)[Signatures](#)[Site news](#)

Protocol

[HTTP](#)[GIT](#)[RSYNC](#)

Location

<https://www.kernel.org/pub/><https://git.kernel.org/><rsync://rsync.kernel.org/pub/>

Latest Stable Kernel:



4.20.12

mainline:	5.0-rc7	2019-02-18	[tarball]	[patch]	[inc. patch]	[view diff]	[browse]	
stable:	4.20.12	2019-02-23	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse] [changelog]
longterm:	4.19.25	2019-02-23	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse] [changelog]
longterm:	4.14.103	2019-02-23	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse] [changelog]
longterm:	4.9.160	2019-02-23	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse] [changelog]

La shell de Linux

- ▶ Es la cara visible al usuario, interpreta las órdenes que recibe y las transmite al Kernel mediante *system calls*.
- ▶ Se la conoce también como *interpretes de comandos*

La shell de Linux

- ▶ Es la cara visible al usuario, interpreta las órdenes que recibe y las transmite al Kernel mediante *system calls*.
- ▶ Se la conoce también como *interpretes de comandos*

Algunas shell son:

`sh` : Bourne shell, escrita por Steve Bourne cuando estaba en Bell Labs

La shell de Linux

- ▶ Es la cara visible al usuario, interpreta las órdenes que recibe y las transmite al Kernel mediante *system calls*.
- ▶ Se la conoce también como *interpretes de comandos*

Algunas shell son:

`sh` : Bourne shell, escrita por Steve Bourne cuando estaba en Bell Labs

`csh` : C-shell, escrita por Bill Joy

La shell de Linux

- ▶ Es la cara visible al usuario, interpreta las órdenes que recibe y las transmite al Kernel mediante *system calls*.
- ▶ Se la conoce también como *interpretes de comandos*

Algunas shell son:

`sh` : Bourne shell, escrita por Steve Bourne cuando estaba en Bell Labs

`csh` : C-shell, escrita por Bill Joy

`ash` : Almquist shell, reemplazo de Bourne shell con licencia BSD

La shell de Linux

- ▶ Es la cara visible al usuario, interpreta las órdenes que recibe y las transmite al Kernel mediante *system calls*.
- ▶ Se la conoce también como *interpretes de comandos*

Algunas shell son:

`sh` : Bourne shell, escrita por Steve Bourne cuando estaba en Bell Labs

`csh` : C-shell, escrita por Bill Joy

`ash` : Almquist shell, reemplazo de Bourne shell con licencia BSD

`dash` : Debian Almquist shell, reemplazo de ash en Debian

La shell de Linux

- ▶ Es la cara visible al usuario, interpreta las órdenes que recibe y las transmite al Kernel mediante *system calls*.
- ▶ Se la conoce también como *interpretes de comandos*

Algunas shell son:

`sh` : Bourne shell, escrita por Steve Bourne cuando estaba en Bell Labs

`csh` : C-shell, escrita por Bill Joy

`ash` : Almquist shell, reemplazo de Bourne shell con licencia BSD

`dash` : Debian Almquist shell, reemplazo de ash en Debian

`bash` : Bourne-Again shell, se escribió como parte del proyecto GNU

La shell de Linux

- ▶ Es la cara visible al usuario, interpreta las órdenes que recibe y las transmite al Kernel mediante *system calls*.
- ▶ Se la conoce también como *interpretes de comandos*

Algunas shell son:

`sh` : Bourne shell, escrita por Steve Bourne cuando estaba en Bell Labs

`cs`h : C-shell, escrita por Bill Joy

`ash` : Almquist shell, reemplazo de Bourne shell con licencia BSD

`dash` : Debian Almquist shell, reemplazo de ash en Debian

`bash` : Bourne-Again shell, se escribió como parte del proyecto GNU

`zsh` : Z shell, superconjunto de sh, ash, bash, csh, ksh, y tcsh

Entornos gráficos

- ▶ **Display Manager:** LightDM, GDM, KDM, LXDM, etc.
- ▶ **Window Manager:** Compiz, Metacity, Mutter, W9dk, fluxbox, etc.
- ▶ **Desktop Environment:** Gnome (GTK), KDE (Qt), LXDE (Lightweight X11 Desktop Environment), XFCE
- ▶ **Graphical User Interface (GUI)**

Algunas aplicaciones

Oficina: LibreOffice, L^AT_EX, LyX



Algunas aplicaciones

Oficina: LibreOffice, L^AT_EX, LyX

Edición imágenes, diagramas, etc.: Gimp,
Inkscape, Blender



Algunas aplicaciones

Oficina: LibreOffice, L^AT_EX, LyX

Edición imágenes, diagramas, etc.: Gimp,
Inkscape, Blender

Programación: GCC, make, ...



Algunas aplicaciones

Oficina: LibreOffice, L^AT_EX, LyX

Edición imágenes, diagramas, etc.: Gimp,
Inkscape, Blender

Programación: GCC, make, ...

IDE de programación: Gedit, Geany,
Code::Blocks, Eclipse CDT, Qt
Creator, Vim, Emacs



Algunas aplicaciones

Oficina: LibreOffice, L^AT_EX, LyX

Edición imágenes, diagramas, etc.: Gimp,
Inkscape, Blender

Programación: GCC, make, ...

IDE de programación: Gedit, Geany,
Code::Blocks, Eclipse CDT, Qt
Creator, Vim, Emacs

Diseño de circuitos (esquem. y PCB), simulación, etc.:
KiCAD, QUCS



Algunas aplicaciones

Oficina: LibreOffice, L^AT_EX, LyX

Edición imágenes, diagramas, etc.: Gimp,
Inkscape, Blender

Programación: GCC, make, ...

IDE de programación: Gedit, Geany,
Code::Blocks, Eclipse CDT, Qt
Creator, Vim, Emacs

Diseño de circuitos (esquem. y PCB), simulación, etc.:
KiCAD, QUCS

Matemática: GNU Octave, Maxima
(wxMaxima), Scilab, R, ...



Algunas aplicaciones

Oficina: LibreOffice, L^AT_EX, LyX

Edición imágenes, diagramas, etc.: Gimp,
Inkscape, Blender

Programación: GCC, make, ...

IDE de programación: Gedit, Geany,
Code::Blocks, Eclipse CDT, Qt
Creator, Vim, Emacs

Diseño de circuitos (esquem. y PCB), simulación, etc.:
KiCAD, QUCS

Matemática: GNU Octave, Maxima
(wxMaxima), Scilab, R, ...

Otros: Firefox, Iceweasel, Evince,
Okular, VLC, Audacity, ...

