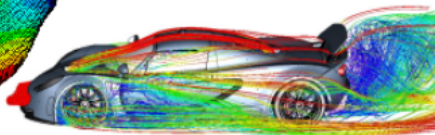
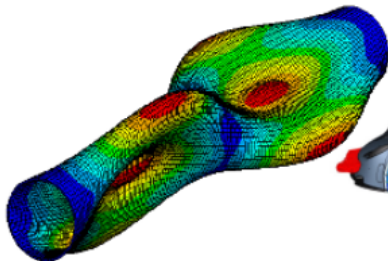
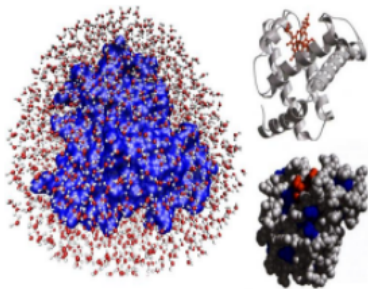


Introducción

Christopher Paredes

2 de junio de 2020

Introducción

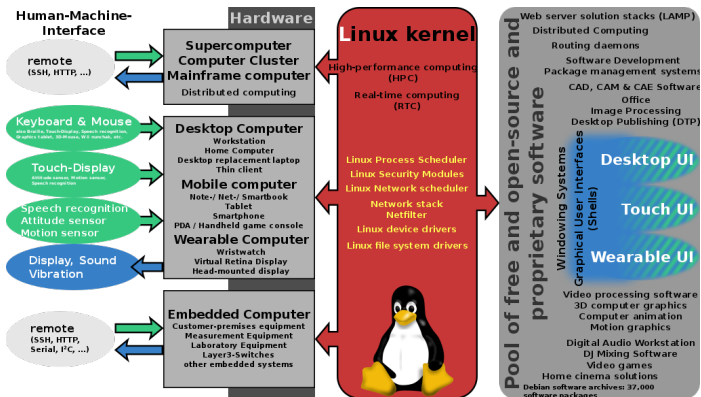


GNU/Linux - Antecedentes



- 1847 Alexander Graham Bell
- 1877 Bell Telephone Company
- 1925 Laboratorios Telefónicos Bell
- 1960 GE-645 SYSTEM -Multics
- 1969 Unix - Ken Thompson y Dennis Ritchie
- 1972 Lenguaje C - Dennis Ritchie y Ken
- 1983 Proyecto GNU (GNU's Not Unix) - Richard Stallman
- 1987 Sistema operativo Minix descrito en Libro *Operating Systems: Design and Implementation* - Andrew Tannenbaum
- 1991 Núcleo Linux versión 0 - Linus Torvalds
- 1993 Distribuciones Red Hat, Slakware y Debian.

Kernel



- Escrito en Lenguaje C.
- <https://www.kernel.org/>
- Kernel de Linus Torvalds 1991 - 10000 líneas de código.
- Kernel Linux 2020 - 27.8 millones de líneas de código. (www.linux.com)

Distribución de software basada en el núcleo Linux. La mayoría de las distribuciones que existen son GNU/Linux: herramientas y bibliotecas de GNU y núcleo Linux.

Debian (1993)	>	Ubuntu(2004)	>	Mint(2006)
	>	Knoppix(2000)	>	Kali (2013)
	>	Raspbian(2012)		
Slackware(1993)	>	Suse(1994)	>	Opensuse(2005)
Red Hat (1993)	>	Fedora(2003)	>	Tizen(2012)
	>	Mandrake(1998)	>	PCLinuxOS(2003)
	>	Scientific Linux(1995)		
Gentoo(2002)	>	Chrome OS(2011)	>	Chromium OS(2011)
Arch (2002)	>	Manjaro(2011)		
Android(2008)	>	CyanogenMod(2009)	>	LineageOS(2016)

Línea de tiempo completa: [aquí](#).

Instalación

- Elegir distribución (<https://distrowatch.com/>) y descargar imagen ISO desde la página oficial.
- Instalar en Virtual Box (<https://www.virtualbox.org/>) o en una partición aparte.

Pasos clave durante la instalación

- 1 Creación del password del usuario **root**
- 2 Creación del usuario **local** y su password.
- 3 Particionado del disco.
 - particionado guiado o manual.
 - formas de particionado LVM, RAID, cifrados, entre otros.
 - sistemas de archivos: ext4, xfs, FAT, NTFS, entre otros.
 - puntos de montaje: en una única partición o en varias.
 - archivo de intercambio (Swap)
- 4 Configuración del gestor de paquetes.
- 5 Elección de software:
 - Sistema base.
 - Escritorios: GNOME, XFCE4, LXDE, KDE,
 - Gestores de ventana: Awesome, Openbox, Wmaker, e17,...
- 6 Instalación del cargador de arranque (GRUB-LILO).

Estructura de directorios

Directorio raíz /

Directorio desde el cual se comienza a ramificar el árbol de directorios.

```
bin    etc    initrd.img.old  lost+found  opt    run    sys    var
boot   home   lib             media       proc   sbin   tmp     vmlinuz
dev    initrd.img  lib64          mnt         root   srv    usr     vmlinuz.old
```

Estándar FHS 3.0

bin	binarios	etc	configuración global del sistema
boot	archivos (estáticos) de arranque.	devices	archivos de dispositivos
home	directorios de usuarios	lib	módulos del kernel y bibliotecas compartidas.
media	puntos de montaje para dispositivos removibles	mnt	puntos de montaje para sistemas de archivos temporales
proc	(virtual) información del sistema	root	directorio del usuario root
run	datos variables en tiempo de ejecución.	sbin	binarios del sistema
sys	(virtual) información del sistema	tmp	archivos temporales
usr	segunda jerarquía	var	datos variables
srv	datos de servicios	opt	software (add-on)

Comandos básicos

Rutas

`/directorio1/directorio2/.../archivo`

Ejecutables

En Linux los archivos ejecutables no se distinguen a partir en una extensión. cualquier archivo se puede establecer como un ejecutable

`./chmod +x archivo`

Comandos básicos

Comando	Descripción	Comando	Descripción
ls	mostrar archivos	mv	mover
mkdir	crear directorio	pwd	directorio actual
rmdir	eliminar directorio	which	ruta archivo
rm	borrar archivo	chmod	permisos
cp	copiar	man	manuals de referencia
uname -a	Versión del núcleo Linux	cat	concatena y muestra contenido de archivos

Procesos y consumo de recursos

- **top**
- **htop**
- **atop**
- **ps**

Dispositivos externos

- **lsusb**
- **pmount**
- **mount**

Recursos del sistema

- **free**
- **cat /proc/cpuinfo**
- **cat /proc/meminfo**
- **df -h**
- **lspci**

Registros

- **dmesg**
- **cat /var/log/archivo-log** (como root)

Usuarios

Usuario local

Es un usuario sin privilegios de administrador. Los datos relacionados a este usuario se encuentran en la carpeta: `/home/usuario/` En un terminal

```
usuario@nombre-maquina:ruta-actual$
```

Root

Es el administrador del sistema para ejecutar comandos como este usuario se debe ingresar al sistema como usuario root:

```
$su  
Contraseña: *****
```

En un terminal

```
root@nombre-maquina:ruta-actual#
```

Existe la opción de ejecutar comandos con permisos de administrador a través del comando **sudo**

```
$sudo comando
```

Repositorio de paquetes

Servidores de paquetes precompilados y empaquetados listos para ser instalados de manera sencilla y segura.

- **repositorio oficial:** cada distribución tiene un repositorio oficial disponible. A la fecha Debian posee alrededor de 59000 paquetes precompilados y empaquetados para ser instalados de manera sencilla y segura.
- **repositorios extras:** si alguien tiene una serie de paquetes, se puede montar un repositorio para distribuirlos.
- **replicas(mirrors):** servidores que funcionan como copias de los repositorios oficiales. Están distribuidos alrededor del mundo.

Réplicas de Debian en todo el mundo <https://www.debian.org/mirror/list.es.html>

Replica Chilena <http://ftp.cl.debian.org/debian/>

Replica Ufro <https://mirror.ufro.cl/debian/>

Más replicas Ufro <https://mirror.ufro.cl/>

Sistema de paquetes - Debian y derivados

APT (Advanced Package Tool)

Es una herramienta que permite gestionar los paquetes de software que están disponibles para Debian o distribuciones derivadas de esta.
Algunos comandos frecuentes:

<code>apt-get install paquete</code>	instala el paquete
<code>apt-get remove paquete</code>	elimina el paquete
<code>apt-get purge paquete</code>	elimina el paquete y los archivos de configuración relacionados
<code>apt-get update</code>	actualiza el listado de paquetes
<code>apt-get dist-upgrade</code>	actualiza la distribución
<code>apt-cache search patrón</code>	busca la lista de paquetes coincidencias con el patrón de búsqueda.
<code>apt-cache show paquete</code>	muestra información del paquete
<code>apt-get clean (autoclean)</code>	limpia archivos descargados (antiguos)
<code>apt-get autoremove</code>	elimina paquetes sin utilizar
<code>apt-get source paquete</code>	descarga archivos fuente del paquete
<code>apt-file search archivo</code>	busca archivos en el listado de paquetes
<code>apt-get moo</code>	apt tiene poderes de Super Vaca

Mas información: `apt-get --help`, `apt-cache --help`, `man apt-get`, `apt --help`,...

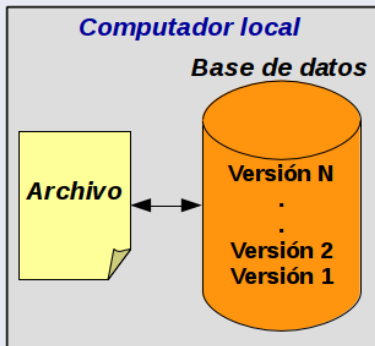
Alternativas: **apt**, **aptitude**, **synaptics** y cualquier otra interfaz para gestionar paquetes de software.

Gestores de paquetes de otras distribuciones: yum (RedHat), portage (Gentoo),
pacman(ArchLinux),...

Control de versiones

Sistema que registra, recupera y mezcla cambios realizados sobre un archivo o un grupo de archivos a lo largo del tiempo.

Control de versiones local.



- Simple de implementar.
- Si no se utiliza un software especializado este método está propenso a errores.
- Para gestión de archivos personales o proyectos simples.
- Ejemplo: rcs GNU

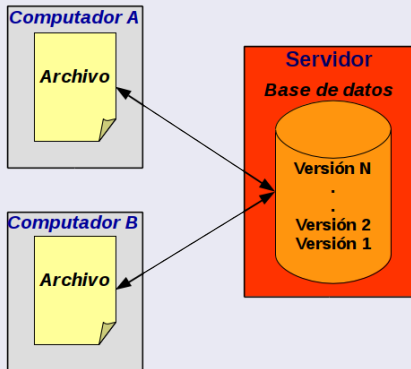
```
#apt-get install rcs  
$mkdir RCS
```

```
$ci prueba.c (almacena)  
$co -l prueba.c (recupera)
```

<https://www.gnu.org/software/rcs/manual/rcs.html>

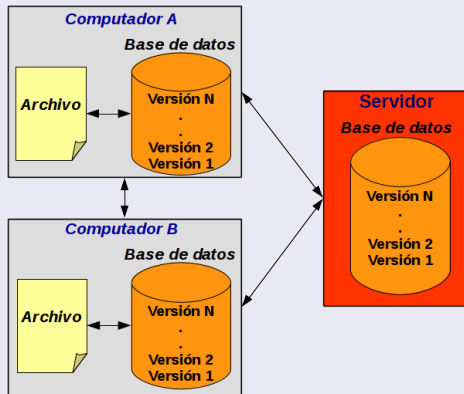
Control de versiones

Control de versiones centralizado



- Mejor administración.
- Si falla el servidor se pierde la colaboración.
- Ejemplos: CVS y subversion (svn)

Control de versiones distribuido.



- Cada clon es una replica completa del repositorio.
- Flujos de trabajo colaborativo que no se pueden lograr con un repositorio centralizado.
- Ejemplos: GIT, Mercurial y Bazaar.

- 2005 - Linus Torvalds



- **Proyectos que utilizan GIT**

- Kernel de linux
<https://git.kernel.org>
- Aplicaciones de GNU: Emacs, gcc, ...
<http://git.savannah.gnu.org>

- **Forjas**

- Wikis para el proyecto.
- Red social y de colaboración.
- Gráficas y herramientas, entre otros.

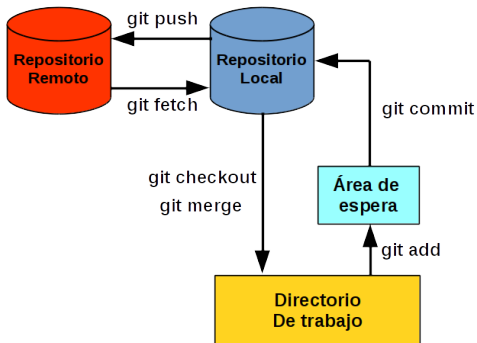




Figura: Teletipo (TTY: acrónimo de teletype)

- `/dev/tty`: dispositivo que actúa como un teletipo. Se comunica directamente con el computador a través del hardware y el núcleo.
- `pty`: pseudo terminal, es un dispositivo que hace las veces de terminal para procesar las entradas y salidas pero es gestionado por una aplicación (`xterm`, `tilda`, `terminator`,...).
- `shell`: interprete de lenguaje de comandos.

Algunas propiedades comunes que tienen las shell según el estándar IEEE Std 1003.1, 2004 Edition

- Las shell leen sus entradas desde archivos o una cadena de comandos (ej: `sh -c "ls"`).
- Sus entradas se separan en **palabras** y **operadores**.
- Procesa entradas en forma de **comandos simples** y **comandos compuestos**.
- Permite expansión de comandos.
- Permite redirecciones de comandos.
- Las funciones (del usuario o integradas en la shell), ejecutables o script reciben parámetros posicionales numerados entre 1 y n, dejando el número 0 para el nombre del script (o función).

Algunas Shell

1971	T shell	Thompson Shell
1978	csh	C Shell
1979	sh	Bourne Shell
1983	tcsh	TENEX C Shell
1983	ksh	KornShell
1989	bash	GNU Bash (Bourne-again shell)
1989	ash	Almquist shell
1990	zsh	Z shell
1997	dash	Debian Almquist shell

<code>~/.bash_history</code>	historial de comandos
<code>~/.bashrc</code>	configuración del usuario local. Se ejecuta cada vez que se ejecuta bash.
<code>~/.bash_profile</code>	configuración de inicialización del usuario local. Se ejecuta cuando el usuario inicia sesión.
<code>/etc/profile</code>	igual que <code>bash_profile</code> pero para todos los usuarios
<code>/etc/bashrc</code>	igual que <code>bashrc</code> pero para todos los usuarios

```
$script.sh
$chmod +x script.sh
```

shebang (#!)

```
#!/bin/bash
```

Mayor portabilidad:

```
#!/usr/bin/env bash
```

Redireccionamiento



fd	nombre	stream	símbolo	archivo	periférico
0	Standard Input	stdin	0<	/dev/stdin	teclado
1	Standard Output	stdout	1>	/dev/stdout	pantalla
2	Standard Error	stderr	2>	/dev/stderr	pantalla

fd (file descriptor): indicador abstracto para acceder a un archivo o un dispositivo de entrada/salida. Los fd se encuentran en `/proc/self/fd/`.

Redireccionamiento de la salida estándar

```
comando > destino (sobre escribe el contenido)
comando >> destino (agrega el contenido al final)
```

Entrada estándar alternativa

```
comando < origen
```

Redireccionamiento

Redireccionamiento de error estándar

```
comando 2> destino  
comando 2>> destino  
comando 2> /dev/null
```

Combinaciones

```
comando 2>&1 destino  
comando < origen > destino
```

Tuberias

```
comando1 | comando2 | comando3 .....
```

Más información en: <http://www.tldp.org/LDP/abs/html/io-redirection.html>

Procesamiento de archivos de texto

grep

Imprime líneas que coinciden con un patrón.

```
grep patrón archivo          # líneas que incluyan coincidencias con "patrón"
grep --color patrón archivo  # activa colores
grep -n patrón archivo       # incluye el número de línea
grep -v patrón archivo       # líneas que no incluyan coincidencias con "patrón"
grep -r patrón directorio    # busca recursivamente dentro del directorio
grep -l patrón directorio    # muestra solamente el nombre de archivo
grep -L patrón directorio    # igual que -v pero muestra solamente el nombre de archivo
grep 'expresión regular' archivo
```

sed

sed [opciones] archivo

```
sed 's/palabra/reemplazo/g' archivo      #reemplaza todas las ocurrencias
sed 's/palabra/reemplazo/ng' archivo     #reemplaza desde la n-ocurrencia
sed 'n s/palabra/reemplazo/' archivo     #reemplaza en una línea específica
sed 'n,m s/palabra/reemplazo/' archivo   #reemplaza en un rango de líneas
echo "Hola Mundo" | sed 's/\([A-Z]\)/\(\1\)/g' #(H)ola (M)undo
```

Expresiones regulares (Regex)

Una expresión regular es un patrón que describe un conjunto de cadenas de texto. Las expresiones regulares se construyen empleando operadores para combinar pequeñas expresiones. Formalmente las expresiones regulares permiten escribir estructuras sintácticas de los lenguajes regulares.

.	coincidencia con cualquier carácter único
^	coincidencia al inicio de línea
?	el ítem precedente es opcional
[0-9]	dígito
foo\$	líneas que terminan en "foo"
^foo\$	líneas que solamente contiene "foo"
^\$	líneas en blanco
[fF]oo	coincidencias con foo o Foo
foo[0-9][0-9]	foo seguido de dos números (ej.: foo15 o foo93)
~/[/*]	líneas que comiencen por // o /*

Mas información `man grep` o `info grep`

```
cat ./ejemplo.dat | grep --color -n '[a]'
```

Lenguaje C

- 1965 - PD7-7
- 1969 - Unix 0 en lenguaje ensamblador PD7-7.
- 1969 - Lenguaje B - K. Thompson
- 1972 - Lenguaje C - D. Ritchie
- 1978 - The C Programming Language, B. Kernighan y D. Ritchie (versión K&R).
- 1983 - C++ - Bjarne Stroustrup
- 1989 - C89 - ANSI C - 1er estándar oficial.
- 1990 - C90 - estándar ANSI C publicado como ISO/IEC 9899:1990.
- 1999 - C99 - ISO/IEC 9899:1999.
- 2011 - C11 - ISO/IEC 9899:2011.
- 2018 - C17 - ISO/IEC 9899:2018.
- 2022 - C2x



Hola mundo en PD7-7

<https://www.youtube.com/watch?v=pvaPaWyuLA>



Bell Laboratories

SUBJECT: **Users' Reference to B** - Case 39199-11

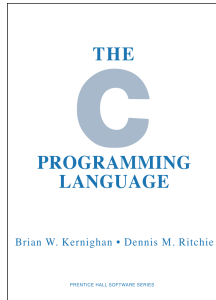
DATE: January 7, 1972
FROM: K._Thompson

MM-72-1271-1

MEMORANDUM FOR FILE

1.0 Introduction

B is a computer language directly descendant from BCPL [1,2]. B



THE C PROGRAMMING LANGUAGE

Brian W. Kernighan • Dennis M. Ritchie

PRENTICE-HALL SOFTWARE SERIES

Lenguaje C - definiciones previas

- Bit

0	Apagado
1	Encendido

- Byte

Conjunto ordenado de bits, en la mayoría de los sistemas tiene 8 bits.

0 1 0 0 0 0 0 1

- Código ASCII

Utiliza 7 bits

0 1 0 0 0 0 0 1

!"#\$%&'()*+,-./0123456789:;<=>
?@ABCDEFGHIJKLMNOPQRSTUVWXYZ
[\]^_`abcdefghijklmnopqrstuvwxyz
{|}~

- Código ASCII - extendido

0 1 0 0 0 0 0 1



Lenguaje C

Dirección de memoria



- Bus de 32 bits - 2^{32} ubicaciones de memoria

$$2^{32} \text{ byte} = 4 \text{ GiB.}$$

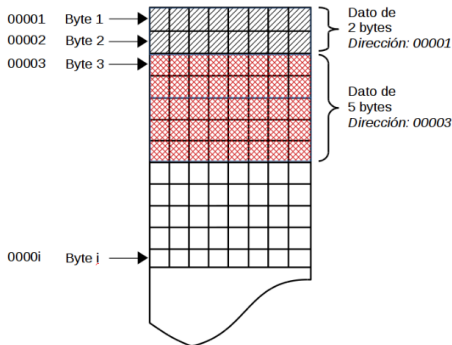
Hex: 00000000H-FFFFFFFFH

- Bus de 64 bits

$$2^{64} \text{ byte} = 16 \text{ EiB}$$

Hex: 0000000000000000H-
FFFFFFFFFFFFFFFFFH

- **Objeto** región de almacenamiento de datos en un entorno de ejecución, cuyo contenido representa algo.
- **Valor** significado preciso del contenido de un objeto cuando se interpreta como un tipo específico.
- **Alineamiento**: requisito de que los objetos de un tipo particular sean ubicados en los límites de almacenamiento con direcciones múltiplos de una dirección de memoria.
- **Acceso**: acción en tiempo de ejecución para leer o modificar el valor de un objeto.



Lenguaje C - definiciones previas

Implementación: conjunto particular de software, que se ejecuta en un entorno de traducción en particular bajo opciones de control particulares, que realiza la traducción de programas y apoya la ejecución de funciones en un entorno de ejecución en particular. Una implementación traslada código fuente C y ejecuta programas C en dos entornos de sistema de procesamiento de datos:

- Entorno de traducción
 - Archivos fuente
 - Unidad de traducción de preprocesamiento
 - Unidad de traducción
- Entorno de ejecución
 - Freestanding
 - Hosted

N1570

Committee Draft — April 12, 2011

ISO/IEC 9899:201x

INTERNATIONAL STANDARD

©ISO/IEC

ISO/IEC 9899:201x

Programming languages — C

ABSTRACT

(Cover sheet to be provided by ISO Secretariat.)

This International Standard specifies the form and establishes the interpretation of programs expressed in the programming language C. Its purpose is to promote portability, reliability, maintainability, and efficient execution of C language programs on a variety of computing systems.