

Bienvenidos al mundo Unix

1. Un nuevo mundo

Linux, que tiene casi 30 años, ha pasado de ser un proyecto de estudiante a ser el sistema operativo más utilizado en el mundo. Desde sus primeros desarrollos en 1991 y hasta la fecha de hoy, Linux no ha dejado de evolucionar, cambiar. El mundo de la informática está vivo. Evoluciona, innova, quizás recicla antiguas ideas para mejorarlas, pero nunca se estanca. Gracias a Linux, millones de personas han encontrado por fin lo que estaban buscando.

Linux no es más complicado que cualquier otro sistema operativo. La lentitud con la que parece penetrar Linux en el gran público no está relacionada con su supuesta dificultad. El buen rendimiento admitido por numerosos usuarios principiantes o avanzados, grupos de usuarios Linux, estudiantes y profesionales en general muestra que se trata de un problema relacionado con las costumbres de la gente, habituada durante años a un sistema operativo único. Y esto sea cual sea el sistema operativo de origen, Windows, MacOS u otro. Resulta evidente que hay que modificar un poco estas costumbres para adaptarse a un entorno Linux.

Aprender Linux es aceptar salir de su zona de confort por su propio bien. El miedo a la consola hace que muchos usuarios piensen, de manera equivocada, que una interfaz gráfica hace que el uso del sistema sea más sencillo. Sin embargo para un servidor, esto no tiene mucho sentido. Los administradores de Windows se han dado cuenta de esto durante los últimos años, dedicando mucho tiempo a escribir scripts de PowerShell. ¡Embarque hacia un nuevo mundo!

2. Historia de los ordenadores

a. Complejidad de los ordenadores

Un ordenador es una máquina electrónica extremadamente compleja. El principio mismo del ordenador tal como lo conocemos no ha cambiado desde la época de Alan Turing o de Conrad Suze y data de principios de los años 40, e incluso antes (máquina de Charles

Babbage). Las evoluciones tecnológicas y la miniaturización han permitido crear máquinas cada vez más potentes y, al mismo tiempo, pequeñas. Desde los primeros ordenadores electromecánicos compuestos de miles de tubos de vacío y cuya programación se llevaba a cabo conectando cables, o girando grandes botones, al ordenador moderno de hoy, la complejidad del hardware ha sido creciente. Hubo una época en la que un ordenador ocupaba toda una planta y el circuito de agua que permitía enfriarlo (algunos eran enfriados mediante bloques de hielo) calentaba todo un edificio. Hoy en día basta con irse de compras al supermercado del barrio para adquirir un ordenador: se podría pensar de manera lógica que la sencillez en la adquisición de uno de estos equipos traería aparejada la sencillez de uso, y que el ordenador debería ser tan fácil de utilizar como su lector de DVD de salón.

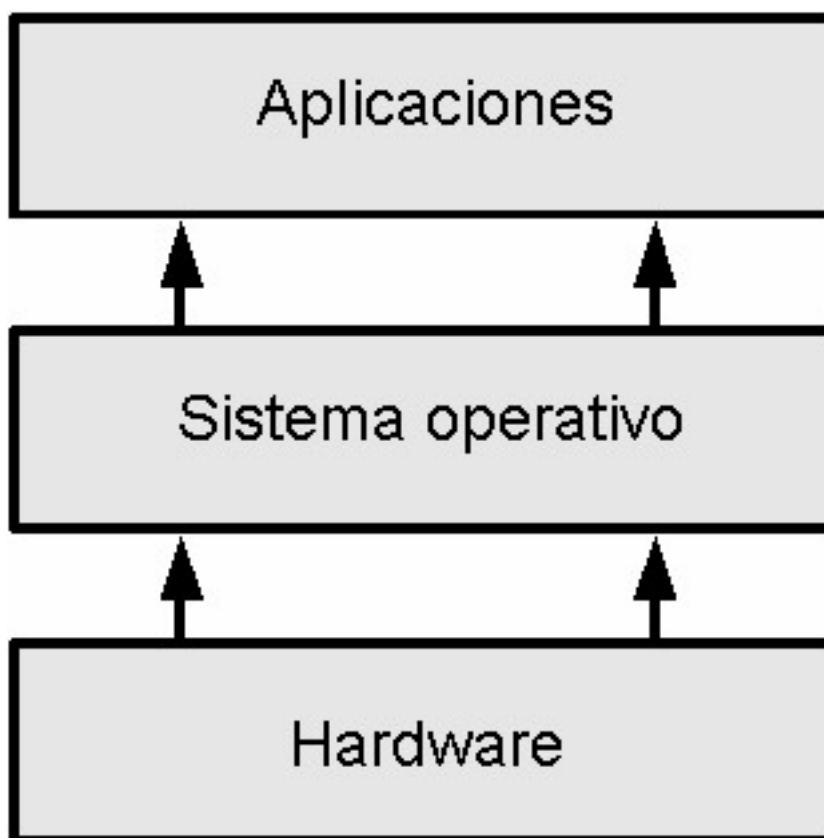
Desafortunadamente, no es el caso. **Un ordenador es una máquina muy compleja.** ¿Debemos esperar que sea tan sencillo de usar como una máquina de mecanismo simple mientras que un ordenador es un equipo completamente polivalente? ¿Debemos conocer a toda costa los entresijos técnicos de nuestro ordenador para poder utilizarlo? La respuesta varía según el uso y las personas, pero suele ser negativa. Sin embargo, hay que recordar y respetar algunas nociones y bases fundamentales.

b. La inteligencia

No hay nada más tonto que una computadora. Hay que decirle lo que debe hacer. El ordenador no posee ninguna inteligencia. No piensa, no reflexiona. Lo que le hace «inteligente» en el sentido de potencia de cálculo es usted y los programas que le hace ejecutar, las órdenes que le proporciona. Aunque esté inventado por humanos para humanos, el ordenador no entiende su lenguaje. Debido a su concepción electrónica, habla en binario, combinación de ceros (0) y unos (1). Estos valores combinados entre sí forman palabras y datos para el ordenador. El binario se convierte en un lenguaje llamado «lenguaje de máquina». Los microprocesadores emplean un lenguaje llamado «ensamblador» en el cual cada instrucción dispone de un equivalente en binario. Con la ayuda de este lenguaje ensamblador, los informáticos desarrollan sus programas, uno de los cuales se llama «compilador»: un traductor de lenguaje de alto nivel y comprensible esta vez por un gran número de informáticos.

3. El sistema operativo

Entre el momento en el que pulsa el botón de encendido de su ordenador y el momento en el que puede por fin trabajar y utilizar sus programas, pasa un cierto tiempo durante el cual se cargan programas en la memoria del ordenador y luego se ejecutan. El objetivo de estos programas es simplificarle la vida al usuario y al programador, al hacer las cosas más sencillas. Estos programas forman un conjunto llamado «sistema operativo». Como indica su nombre, el papel del sistema operativo consiste en sacar partido al ordenador tanto como sea posible. En otras palabras, **el sistema operativo le proporciona todos los mecanismos necesarios para aprovechar lo mejor posible los recursos de su ordenador**. Si tiene la impresión de que es el equipo el que se aprovecha de usted, entonces cambie rápido de sistema operativo.



Fundamentos del sistema operativo



Un sistema operativo es un programa o un conjunto de programas que garantiza la gestión del ordenador y de los periféricos. Sirve de interfaz entre el medio físico (hardware) y el programa (software). Es un conjunto de programas muy complejos cuya meta es hacer más sencillos los programas y el uso del ordenador.

El sistema operativo ofrece a los programadores una interfaz de programación de aplicaciones llamada **API**, *Application Programming Interface*. Todos los programadores utilizan las mismas funciones en sus programas, lo que simplifica mucho el trabajo. Pueden centrarse en el objetivo de su programa (crear un procesador de texto, por ejemplo) sin tener que escribir una y otra vez las secuencias de programas que gestionan el disco duro, la impresora o el acceso al teclado. Le corresponde al sistema operativo gestionar:

- ~ la memoria,
- ~ los accesos a los periféricos,
- ~ los datos en los discos,
- ~ los programas,
- ~ la seguridad,
- ~ la recopilación de la información.

Falta aún la interfaz gráfica. Este punto es el tema de muchas discusiones. En un producto como Microsoft Windows la interfaz gráfica está incluida en el sistema operativo. Desde hace mucho tiempo es imposible trabajar sin ella, cualquier ajuste en el sistema se realiza desde un cuadro de diálogo. Los usuarios suelen pensar que la interfaz gráfica es el sistema operativo. Sin embargo, la interfaz gráfica no siempre ha formado parte del sistema operativo. Viene como complemento. No encontrará ningún libro sobre teoría de los sistemas operativos que trate de las interfaces gráficas. ¿Por qué? ¿Cuál es el interés, más allá de aumentar la lentitud y ocupar preciosos recursos de la máquina, de tener una interfaz gráfica para gestionar un servidor de Internet? Linux ofrece interfaces, pero son programas como los demás.

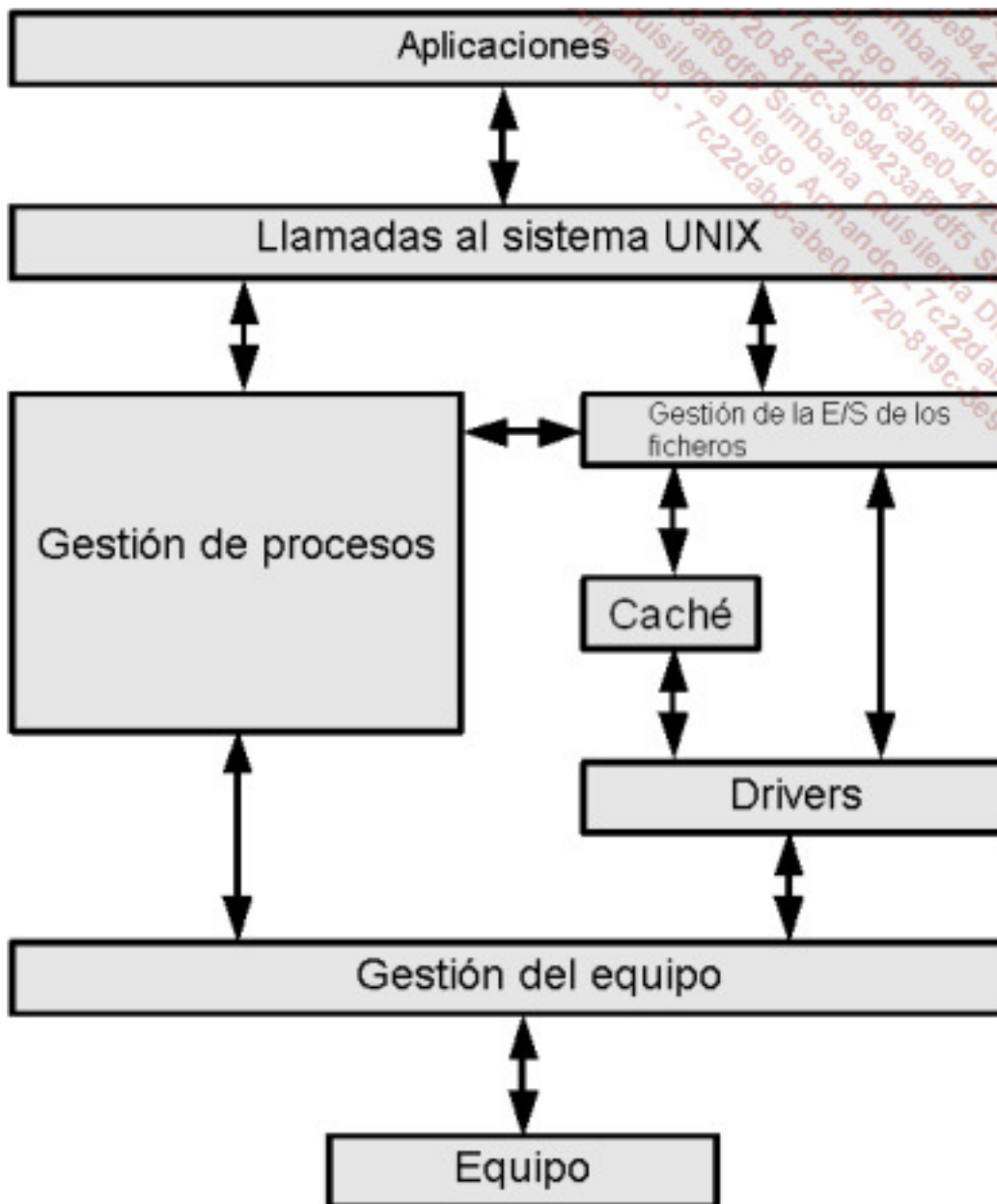


La interfaz gráfica no es un componente del sistema operativo y Linux no la necesita para funcionar correctamente. Es un conjunto de programas ejecutados de forma clásica «sobre» el sistema operativo.

Sin embargo, los controladores de la tarjeta gráfica son componentes del sistema operativo: las bibliotecas gráficas se basan en la API propuesta por el controlador para mostrar texto, ventanas o juegos.

Linux es un sistema operativo de tipo Unix. Existen decenas de sistemas operativos en esta familia. Puede sorprenderle el hecho de que MacOS es una versión de Unix como cualquier otra. Unix es un sistema operativo de la familia de los sistemas **multitarea** y **multiusuario**:

- ✓ **Multitarea:** el sistema gestiona la ejecución simultánea de varios programas llamados procesos (nota: un verdadero sistema multitarea necesita contar con varios microprocesadores o unidades de ejecución, como el Hyper Threading, por ejemplo).
- ✓ **Multiusuario:** el sistema permite la existencia de varios usuarios diferentes en una misma máquina, conectados o no (un usuario puede ejecutar un programa sin estar conectado, como por ejemplo un servidor de Internet).



Arquitectura lógica de un sistema UNIX

El esquema anterior es una síntesis simplificada de la estructura interna de un sistema operativo Unix. Abajo se encuentra el hardware; encima, los programas que ejecuta su equipo. Entre ambos, los diferentes componentes del sistema operativo garantizan su correcto funcionamiento:

- ✓ Las **llamadas al sistema** son utilizadas por los programas para comunicarse con el sistema operativo Unix.
- ✓ La **gestión de los procesos** se encarga de la conmutación de las tareas y de su

prioridad. Este componente se encarga, por lo tanto, de la multitarea.

- ✓ La **gestión de las entradas y de las salidas de archivos** se encarga tanto de la lectura y escritura de los datos en sus discos duros como también en sus periféricos (tarjeta de sonido, impresora, red, etc.).
- ✓ Se puede ubicar cierta información en una zona de memoria llamada **caché**. En lugar de escribir datos directamente en el soporte de almacenamiento (un proceso por lo general lento), Unix va a escribirlos provisionalmente en una zona de memoria de acceso más rápido, para luego escribirlos en el disco, después de unos segundos, mientras que usted se ocupa de otras cosas. Así, la lectura de estos datos es más rápida porque ya están en memoria y el programa no pierde tiempo en esperar el fin de la escritura de los datos.
- ✓ Los **controladores** tienen como papel gestionar al más bajo nivel el hardware o las estructuras lógicas del hardware (por ejemplo, los datos de una partición).



Una aplicación bien programada en un sistema operativo bien programado no puede cortocircuitar este esquema: no «discute» nunca con el hardware, sino que dialoga obligatoriamente con las API correspondientes.

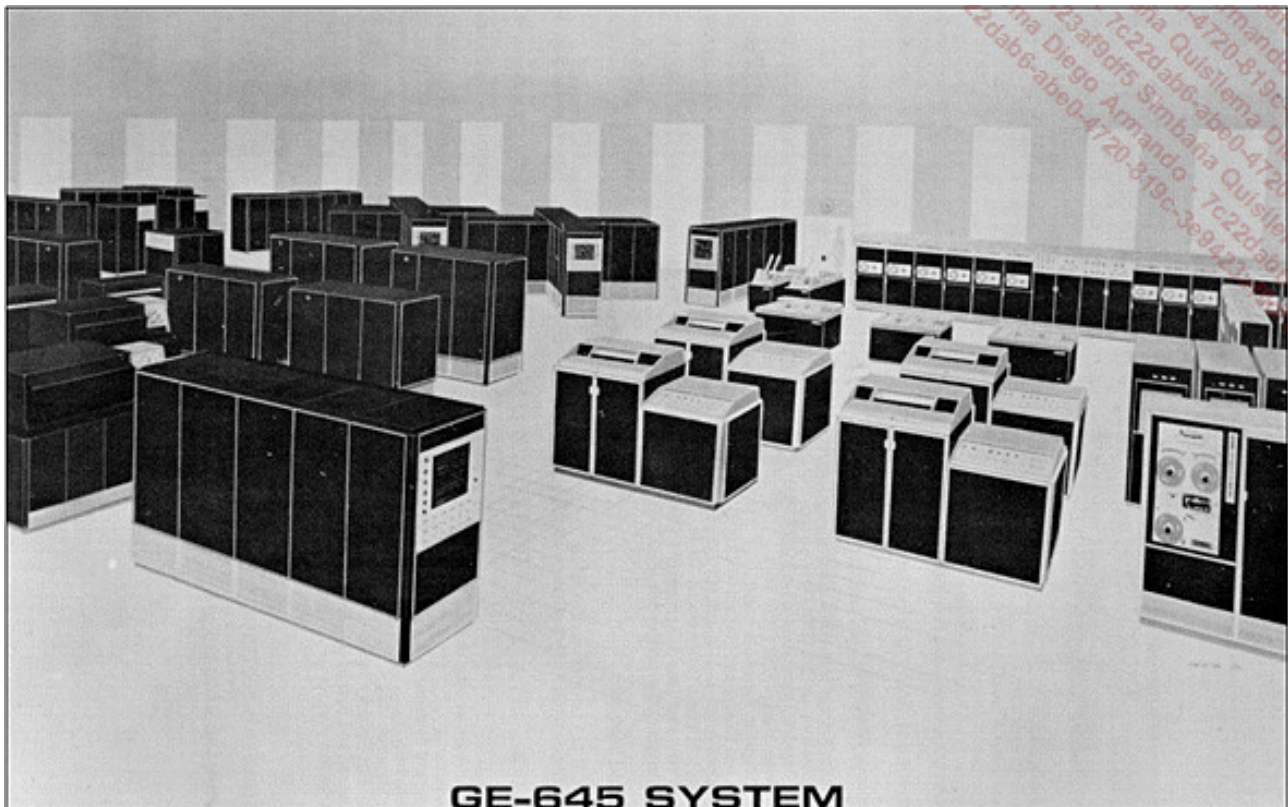
4. El sistema Unix, una breve historia

a. De MULTICS a UNIX

La historia de Unix empieza en **1964**, cuando el MIT, el laboratorio Bell Labs de AT&T y General Electric comienzan a desarrollar el proyecto experimental **MULTICS** (*Multiplexed Information and Computing Service*). El proyecto Multics responde a estas nuevas necesidades:

- ✓ posibilidad de ser utilizado por varias personas a la vez,
- ✓ posibilidad de lanzar procesos en segundo plano,
- ✓ una gestión más adecuada de la seguridad.

Multics se desarrolla en un enorme sistema GE-645 de General Electric, equipado con dos procesadores que pueden tratar cada uno 435 000 instrucciones por segundo, tres unidades de memoria de 1 MB cada una y 136 MB de almacenamiento. Funcionó en el MIT hasta 1988, con 82 estaciones de trabajo, y un máximo de 200 usuarios y llegó a trabajar de forma simultánea en General Electric. La última instalación Multics que ha sido desactivada es la del Ministerio de Defensa Canadiense, el 30 de octubre de 2000.



El GE-645

Sin embargo, si bien Multics alcanzó pronto un grado de estabilidad lo suficientemente aceptable como para pasar a producción, resultó que su rendimiento era menor de lo esperado. En **1969** Bell Labs abandona el proyecto para dedicarse al desarrollo de otro sistema llamado GECOS.

Ken Thompson, desarrollador en Bell, continuó trabajando en el GE-645 y se propuso crear un juego llamado Space Travel. Al ejecutarse en Multics resultó ser muy lento y costoso, más si se le medía en tiempo compartido (antes de la multitarea, el tiempo de máquina era seccionado y cada una de estas era contabilizada y facturada).



Ken Thompson

Ken reescribe entonces el juego en un ensamblador para el miniordenador DEC PDP-7. En esta misión le ayuda **Dennis Ritchie**, procedente también de Bell Labs. Esta experiencia, combinada con la obtenida durante el diseño de Multics, empuja a los dos hombres y a su equipo a crear un nuevo sistema operativo para el PDP-7. **Rudd Canaday**, también de Bell Labs, estaba desarrollando justamente un nuevo sistema de archivos que diseña como si fuera un sistema operativo; de ahí que Unix sea un sistema orientado a archivos, en el cual todo (o casi) es un archivo. Le añadieron un intérprete de comandos y algunas herramientas. Llamaron al sistema **UNICS** (*Uniplexed Information and Computing System*), según una idea de **Brian Kernighan**. Desde el principio el proyecto podía gestionar dos usuarios al mismo tiempo en modo multitarea real.



La DEC PDP-7



El origen de la palabra es tema de numerosas leyendas, todas las cuales tienen probablemente su parte de verdad. UNICS es un juego de palabras en clave de humor de MULTICS, cuya arquitectura es el motivo de numerosas críticas en esta época: «MULTICS (múltiple) hacía la misma cosa de varias maneras, mientras UNICS (único) hacía cada cosa de la misma manera». Además, en inglés UNICS se pronuncia como «eunuchs», o sea, «eunuco», **un sistema Multics «castrado»**.

UNICS retoma los conceptos esenciales desarrollados para MULTICS y los mejora. En particular, sus diseñadores proponen un sistema de comunicación completamente nuevo entre los programas, en el cual un primer programa puede enviar sus datos a otro programa. Rápidamente, el CS fue sustituido por un X, una letra menos para la misma presentación. La leyenda **UNIX** acababa de nacer.

Todo hubiera podido finalizar aquí, ya que los equipos trabajaban sin financiación. Bell Labs se deshizo de Multics y de sus sucesores. Para proseguir los trabajos, Thompson y Ritchie proponen a Bell Labs añadir un programa de procesador de textos para el PDP-11/20 en UNIX. La compañía acepta, se pone a disposición la máquina y el equipo obtiene financiación y respaldo oficial. La herramienta *runoff* (que se convertirá en *roff* y luego en *troff*) y el editor *ed* son desarrollados y por primera vez en **1970** se emplea la denominación **Unix Operating System**. Bell utiliza entonces Unix como un sistema de procesador de texto para la redacción de sus patentes. El primer manual de programación Unix data del 3 de noviembre de 1971.

b. El lenguaje C

Rápidamente, los ingenieros deben enfrentarse a un nuevo problema. Desarrollado en ensamblador y por lo tanto en lenguaje máquina, Unix debe ser reescrito en parte para cada nuevo modelo de ordenador DEC, y programar en ensamblador es un arte difícil. Es el momento de enfrentarse al reto de la portabilidad, ya que cada nuevo equipo es diferente del anterior. En 1970, Thompson se interesa por el problema. Primero piensa en desarrollar Unix en lenguaje TMG para crear y utilizar Fortran. Al encontrar el lenguaje incompleto, se asocia con Dennis Ritchie para crear el **lenguaje B**, que proviene del

lenguaje **BCPL**. Pero tampoco les parece el más adecuado (problema con el tipado de las variables y los números reales). Ritchie parte del lenguaje B y desarrolla el **New B**, que llama lógicamente el **lenguaje C**. C es volcado a lenguaje máquina tras pasar una etapa de compilación. La escritura de los programas resulta más rápida.



Dennis Ritchie (1941-2011)

Unix es reescrito en lenguaje C a partir de **1973**. Para pasar un sistema Unix de una máquina a otra, basta con que haya un compilador C instalado en la nueva máquina. Es mucho más sencillo y eficiente escribir un compilador C (él mismo escrito en gran parte en C) que escribir de nuevo todo un sistema operativo en ensamblador. Únicamente los elementos que tenían mayor contacto con la arquitectura física de la máquina fueron escritos directamente en lenguaje máquina. Unix se hace portable y eso permite acelerar su desarrollo.

c. Las licencias y el advenimiento de BSD y System V

Un hecho de gran calado va a contribuir en aquel entonces a consolidar la gran difusión de Unix (la palabra «gran» puede parecer exagerada cuando se está hablando de decenas de copias del sistema). AT&T, de la cual depende Bell Labs, fue objeto en 1956 de un decreto antimonopolio que le prohibía comercializar cualquier producto ajeno a su sector industrial: las telecomunicaciones. No puede vender Unix (lo cual, vista la utilización actual del sistema, sería discutible). AT&T (que ni siquiera le ve futuro comercial) decide en **1974** liberar completamente el sistema UNIX, pero sólo con fines educativos a las universidades y las empresas y bajo una licencia que resulta ser muy poco restrictiva. Sólo el código fuente (*el texto del programa no compilado aún*) del núcleo en ensamblador no se incluirá en la liberación, al menos oficialmente. UNIX cuenta ya con algunas versiones, de entre las cuales las más difundidas son la sexta, de **1975**, y la séptima, de **1978**. La siguiente versión, Unix v7, fue la primera específicamente diseñada con la intención manifiesta de ser llevada a otras máquinas además de las PDP, en particular al modelo VAX 11/780. Se considera que la v7 fue la última versión completamente común a todos los Unix siguientes.

Por aquel entonces se produce otro gran acontecimiento. Unix iba a celebrar sus diez años y las universidades estadounidenses se estaban implicando con fuerza en su difusión y mejora cuando AT&T modifica la licencia de Unix haciéndola más restrictiva. La entidad que comercializa Unix tiene la autorización de vender licencias del código fuente. Las tarifas, prohibitivas, obligan a las universidades a continuar, para lo bueno y lo malo, sus desarrollos tomando como punto de partida las versiones anteriores al cambio de la licencia. Una de estas universidades se encuentra en California y se llama **Berkeley**. Berkeley es la mayor contribuidora de Unix desde que empezó a trabajar en él, en 1974. La primera versión UNIX de **BSD** (*Berkeley Software Distribution*) está basada en Unix v6, de **1977**, y recibe el nombre comercial de **1BSD**. La versión **2BSD** está basada en Unix v7 y data de 1978.

A partir de ahí, se asiste al enfrentamiento de dos escuelas UNIX. La primera, en teoría la oficial, es la de AT&T, que va a seguir desarrollando las versiones 8, 9 y 10 durante los años 80 del siglo XX, con fines científicos. De forma paralela, desarrolla un Unix totalmente comercial llamado **Unix System III**, que se comercializa a partir de **1982**. En **1983** AT&T desarrolla y vende las primeras versiones Unix System V. **Unix SystemV release 4.2**, data de **1993**. Esta versión de Unix se conoce con la abreviatura SVR4 y su principal característica es que su código fuente está disponible bajo licencia. Esto significa que un organismo está autorizado a comprar una y a desarrollar su propia

versión comercial.

Una versión SVR5 fue publicada en 1997, desarrollada por Santa Cruz Operation (SCO), sin embargo, el código asociado nunca fue reutilizado por otras compañías. Desde aquel entonces, no ha habido ninguna nueva versión SVR. SVR4 es la versión de referencia.

Durante este tiempo, la universidad de Berkeley no descansa y sigue desarrollando BSD como alternativa Open Source al Unix System III y V, cuyas fuentes no puede usar por carecer de derechos. Es en BSD donde se va a implementar por primera vez el protocolo **TCP/IP**, base del Internet moderno, gracias a la financiación del Ministerio estadounidense de Defensa. La última versión oficial de BSD es la **4.4BSD** y data de junio de **1994**.

d. La guerra de los Unix

El período que va desde la mitad de los años 80 hasta 1994 no ha estado exento de sobresaltos. Los efectos de la separación de Unix en dos ramas ha resultado ser un desastre y a punto ha estado de significar su pérdida total. Los dos campos (AT&T con su System V y Berkeley con su BSD) no se ponen de acuerdo en un estándar común. La consecuencia, además de la celebración de múltiples juicios (hasta 1993) sobre el uso del nombre y herramientas derivados de Unix, es el hecho de que han crecido como la mala hierba multitud de versiones de Unix comerciales y sobre todo incompatibles entre ellas. Es a partir de esta época cuando surgen los grandes nombres de los clones Unix, entre los cuales cabe destacar **Solaris, AIX, OSF1 / Digital Unix / tru64, Xenix, HP-UX, IRIX, Ultrix, Unixware, A/UX**. Todos son incompatibles con el vecino, pero claman alto y claro su pertenencia a Unix. Esta guerra de los Unix se conoce realmente como el período oscuro de las *Unix wars*. Pero lo más triste es que nadie parece interesado en ponerse de acuerdo sobre una base y un estándar común. El efecto directo de esta guerra ha sido la creación de un vacío en el mercado aprovechado por Microsoft para colocar con fuerza su sistema operativo **Windows NT** (que a su vez, pero poca gente lo sabe, deriva de Unix).

En **1984** un grupo de editores comerciales de Unix intenta una primera estandarización creando X/Open Standards y publicando un documento llamado **X/Open Portability Guide**, que describe un estándar abierto (accesible a todos) para Unix. Este comité obtiene un gran espaldarazo en **1987**, cuando Sun Microsystems y AT&T deciden trabajar sobre un Unix unificado, fusión de BSD y de System V. El resultado es en realidad System V Release 4.

El refrán dice que pican más los celos que las pulgas. La competencia acusa a Sun de

dirigir el cotarro y funda *Open Software Foundation*, o sea, **OSF**, en **1988**. OSF declara que también persigue un estándar abierto para Unix, pero basado de manera exclusiva en BSD. Publicando sus especificaciones en **1990**.

Como respuesta, AT&T y su grupo crean *Unix International* en **1989**, en un enésimo intento de unificación, sin éxito. En medio de este embrollo, AT&T decide deshacerse de Unix, del que sigue siendo propietario oficialmente, y crea para ello una empresa llamada *Unix System Laboratories* en **1992**. Todos los derechos de Unix se transfieren a USL.

e. La estandarización

Un año más tarde, y cuando la situación amenaza con estancarse, aparece un nuevo actor que pretende triunfar donde los otros fracasaron. La empresa **Novell** compra USL el año de su creación y se convierte en el propietario de SVR4.2. En **1993** Novell cede la marca Unix a X/Open. Unix International desaparece en **1994** y se reestructura OSF. Finalmente, en **1995** Novell cede la licencia operativa del código fuente de Unix a la empresa **SCO** *Santa Cruz Operations* (que se convertirá en Caldera y luego, de nuevo, en SCO). El mismo año X/Open y OSF se fusionan de manera definitiva y forman **The Open Group**. Unix es una marca registrada perteneciente a The Open Group, encargada por Novell de explotar la marca. Después de un proceso de litigio entre Novell y SCO acerca de la propiedad intelectual del código fuente, la justicia americana establece en 2006, tras las múltiples apelaciones siguientes (hasta 2011), que dicha propiedad es de Novell. Luego, Novell fue comprada por Attachmate y Attachmate por Micro Focus.

Ya sólo existe un único organismo de estándar Unix. Unix es, por lo tanto, un sistema operativo abierto: sus especificaciones son conocidas y cada editor de sistema Unix comercial o gratuito que desea asegurar una compatibilidad con el conjunto de los Unix debe implementar este estándar, aunque sigue siendo libre de programar este estándar como desee, ya que una misma función puede escribirse de varias maneras.

f. Unix es un estándar

Para asegurarse de que todos los Unix siguen las mismas recomendaciones, The Open Group publica las normas (*Single Unix Specification*, *Unix95*, *Unix98*, *Linux Standard Base*, etc.) y puede proponer certificaciones. Estas normas se apoyan, en parte, sobre las definidas desde **1988** por el **IEEE** *Institute of Electrical and Electronics Engineers* (que se pronuncia «I3E») y en particular en la **IEEE 1003**, llamada también **POSIX** (*Portable*

Operating System Interface). La X es al mismo tiempo una herencia y un reconocimiento al trabajo efectuado en Unix. IEEE 1003 se compone de 15 documentos que agrupan por temas todo lo que debe contener un Unix (comandos básicos, intérprete de comandos, interfaces de usuario, funciones de programación, etc.) para estar conforme al estándar POSIX. POSIX no se limita a Unix. Windows NT es compatible con POSIX para algunos de sus componentes. POSIX no es un estándar abierto. Las especificaciones de *The Open Group* están abiertas y accesibles a todos y los editores prefieren referirse a ellas. La última revisión POSIX es el IEEE Std 1003.1-2017, mejor conocido como POSIX:1-2017.

El estándar POSIX a veces se modifica debido a controversias o por razones de sentido común. El proyecto GNU está detrás de la iniciativa de un cambio de las herramientas `du` y `df`, ya que por defecto el resultado de estos comandos es múltiplo de 512 bytes, en lugar de 1 KB, que es mucho más fácil de leer. Esto ha conducido a la incorporación de una variable llamada `POSIXLY_CORRECT`, que todo el mundo utiliza.

Para poder utilizarse en algunas administraciones estadounidenses, un sistema operativo Unix debe ser compatible con estándar POSIX. De hecho, cuando Linux tuvo que usarse, el gobierno de Bill Clinton hizo que el Tesoro estadounidense financiara la totalidad de la certificación **PCTS** (*Posix Conformance Test Suite*). Esto no significa que Linux esté certificado por POSIX, sino que en un momento dado algunas versiones de distribuciones Linux han sido certificadas. Una certificación tiene un coste muy alto y su obtención requiere tiempo. **Recordemos sin embargo que Linux es un sistema operativo que respeta el estándar POSIX sin estar certificado.**



Las últimas versiones oficiales de BSD y System V datan de 1994. Los Unix concebidos a partir de 1995 implementan las recomendaciones de The Open Group. Sin embargo, históricamente algunas versiones siguen más «orientadas» a BSD, o más a System V en lo que se refiere a su configuración, o a veces a los dos, como Linux (y según la distribución).

g. Unix en los ordenadores personales

El primer Unix para ordenador personal, en el sentido de ordenador de tipo IBM PC es

Xenix. Proviene de Unix v7 y salió en **1983** para PC (algunas versiones han estado disponibles incluso antes en otras arquitecturas físicas). Microsoft ha llevado Xenix, a costa de numerosas modificaciones, a otras máquinas. La versión 2 de Xenix data de 1985 y está basada en Unix System V. Cuando en 1987 IBM lanzó el sistema operativo OS/2 en asociación con Microsoft, este último transfirió los derechos de Xenix a SCO. La versión 2.3.1 de este mismo año soporta el 386, SCSI y TCP/IP. Xenix se convierte en SCO Unix en 1989 y luego desaparece en favor de SVR4.

El norteamericano **Andrew Stuart Tanenbaum** (apodado Andy) es investigador y profesor de informática de la Universidad libre de Ámsterdam. También es autor de obras de referencia en informática sobre la teoría de los sistemas operativos. En **1987**, con fines pedagógicos, concibe y escribe el sistema operativo Minix. Utiliza 20 MB de espacio en disco y necesita pocos recursos (2 MB de memoria física). Minix tendrá una gran importancia para Linux. Minix sigue existiendo y la versión 3.4.0rc6 salió en mayo de 2017.



Andrew Tanenbaum

Hay otras versiones de BSD adaptadas al PC. Una de ellas es el **386BSD**, de octubre de

1989 y que deriva de 4BSD. Su sucesor es el **NetBSD** (versión 7.0.2 en octubre de 2016), que deriva de 4.3BSD y de 4.4BSD. NetBSD es el sistema operativo más portado a otras arquitecturas físicas. **OpenBSD** (versión 6.0 en septiembre de 2016) está basado en 4.4BSD y muy orientado a la seguridad: en ocho años sólo se pudo encontrar un fallo de seguridad. **FreeBSD** (versión 11.0 en octubre de 2016) deriva también de 4.4BSD y proviene directamente de la época de los juicios entre BSDI y AT&T. Gracias a ello, FreeBSD es completamente libre y abierto.

Solaris, el Unix de Sun Microsystems, está disponible desde hace varios años en PC. Su última versión, la 11.3, salió en octubre de 2015. Solaris se ofreció como versión libre entre 2005 y 2010, año en el cual Oracle compró Sun y regresó al sistema propietario (Closed Source), condenando de esta forma a OpenSolaris. Desde entonces, el proyecto **illumos** intenta engendrar un derivado (fork).

Por último, **Linux** es sin duda el Unix libre más conocido y más extendido en el PC. La historia de su creación merece destacar algunos detalles adicionales.