

# 1. Introducción

## 1.1. ¿Qué es un sistema operativo?

Un sistema operativo es un programa, o más bien un conjunto de programas o algoritmos, que actúa como intermediario entre el usuario (en su sentido amplio) de un computador y el hardware del computador, como vemos en la Figura 1.1.

Dicho más específicamente un sistema operativo (SO) es un software que gestiona los recursos de hardware y software de las computadoras y proporciona servicios comunes para los programas.

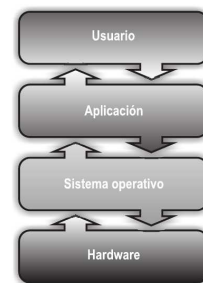
Las personas interactuamos con las aplicaciones, no directamente con el sistema operativo. A modo de ejemplo, podemos mencionar a los procesadores de texto, las hojas de cálculo, una aplicación de contabilidad, un navegador web, un reproductor multimedia, un juego. También se los suele llamar programas, aplicación o software de aplicación.

Podemos también mencionar a las **bibliotecas**, mal llamadas «librerías» porque el término proviene de *libraries*, en inglés, pero se acepta y se entiende en la jerga informática. Son un conjunto de implementaciones funcionales en la forma de rutinas (o subrutinas) que una aplicación puede invocar para realizar una operación común a muchas aplicaciones, por ejemplo *codecs* de audio, funciones matemáticas o de entrada/salida.

Para llevar a cabo la función de procesamiento, una computadora con arquitectura von Neumann (1945) está compuesta por los cuatro componentes básicos representados en la Figura 1.2.

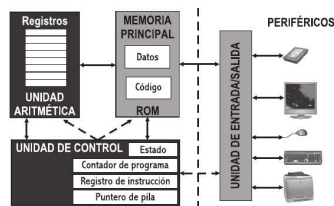
Las computadoras son máquinas de arquitectura von Neumann cuando:

- Tanto los programas como los datos se almacenan en una memoria en común. Esto hace posible la ejecución de comandos de la misma forma que los datos.
- Cada celda de memoria de la máquina se identifica con un número único, llamado dirección.
- Las diferentes partes de la información (los comandos y los datos) tienen diferentes modos de uso, pero la estructura no se representa en memoria de manera codificada.



**Figura 1.1.:** Ubicación del SO

- Cada programa se ejecuta de forma secuencial que, en el caso de que no haya instrucciones especiales, comienza con la primera instrucción. Para cambiar esta secuencia se utiliza el comando de control de transferencia.



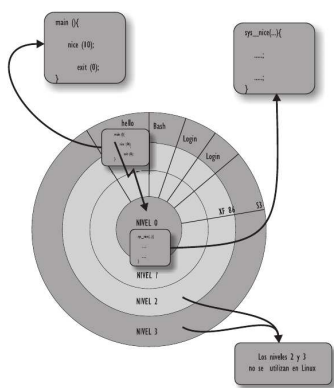
**Figura 1.2.:** Componentes de una computadora

Recordemos que este modelo describe una arquitectura de diseño para una computadora digital electrónica con partes que constan de una unidad de procesamiento que contiene una unidad aritmético lógica y registros del procesador, una unidad de control que contiene un registro de instrucciones y un contador de programa, una memoria para almacenar tanto datos como instrucciones, almacenamiento masivo externo, y mecanismos de entrada y salida.

Para lograr sincronización entre CPU y dispositivos de entrada o salida hay dos métodos:

Para lograr sincronización entre CPU y dispositivos de entrada o salida hay dos métodos:

1. Transferencia de datos por interrupciones. Una interrupción (del inglés *interrupt request* -IRQ- o pedido de interrupción) es una señal recibida por el procesador para indicarle que debe «interrumpir» el curso de ejecución actual y pasar a ejecutar código específico para tratar esta situación.
2. Transferencia de datos por acceso directo a memoria (DMA, del inglés *Direct Memory Access*). Permite a cierto tipo de componentes de una computadora acceder a la memoria del sistema para leer o escribir independientemente de la CPU principal. Permite a dispositivos de diferentes velocidades comunicarse sin someter a la CPU a una carga masiva de interrupciones.



**Figura 1.3.:** Anillos de protección

Los procesadores tienen, al menos, dos modos de operación: el sistema operativo es el encargado de operar en el modo **núcleo** o supervisor del microprocesador mientras que el resto de los programas operan en el modo **usuario** o protegido, como vemos en la Figura 1.3.

Son los llamados dominios de protección jerárquica o anillos de protección, son mecanismos para proteger datos y funcionalidad de los fallos y comportamiento malicioso.

Los anillos están dispuestos en una jerarquía desde los más privilegiados (de más confianza), usualmente numerado cero, hasta el menos privilegiado (de menos confianza), usualmente con el mayor

número de anillo.

En el anillo más privilegiado se tiene acceso a todo el conjunto de instrucciones del procesador y a todas las posiciones de memoria.

Por el contrario, en el anillo menos privilegiado sólo se tiene acceso a un conjunto restringido de instrucciones de procesador y a un área menor de posiciones de memoria.

Los procesadores suelen tener más anillos que los que usan los sistemas operativos.

Por ejemplo, tanto Windows como Linux utilizan dos anillos mientras que el sistema operativo OS/2 utilizaba tres.

Para «saltar» de un modo a otro, las aplicaciones se valen de las **llamadas a sistema** (o, *system call*, en inglés).

¿Qué es una llamada a sistema?

La llamada a sistema la invoca un proceso de usuario (o mejor dicho, un proceso que ejecuta con el procesador en modo usuario) y es servida por el núcleo, que opera con el procesador en modo supervisor.

Una llamada al sistema implica pasar o saltar del código del usuario al código del núcleo. Este salto conlleva un cambio en el modo del funcionamiento del procesador.

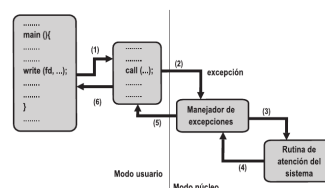


Figura 1.4.: Llamada a sistema

## 1.2. Las primeras computadoras

Ante la pregunta acerca de cuál fue la primera computadora del mundo, la respuesta parece sencilla. No obstante, no lo es. Ya que la respuesta cambia según la definición de computadora que se utilice para darla.

Si bien la historia de las herramientas para calcular nos remontan a la época del ábaco, los sistemas de cómputo previos al electrónico nos llevan a la década de 1930; veamos:

### 1.2.1. Z1

Konrad Zuse, ingeniero alemán y pionero de la computación, construyó en 1936 un equipo llamado Z1 y que era una calculadora mecánica binaria operada con electricidad y de programabilidad limitada. Leía instrucciones desde una cinta perforada. La Z1 nunca funcionó bien, debido a la falta de suficiente precisión mecánica. La Z1 y sus planos originales fueron destruidos durante la Segunda Guerra Mundial.

### 1.2.2. ABC

Entre 1937 y 1942 el Dr. John Vincent Atanasoff con la ayuda de Clifford Edward Berry construyeron el *Atanasoff Berry Computer* (ABC). Era una computadora electrónica y di-

gital automática que se usó con números y letras, para la solución de sistemas de ecuaciones algebraicas lineales. Fue en la «Iowa State University», que entonces recibía el nombre de «Iowa State College». No obstante esto, muchos historiadores de la tecnología informática cuestionan su prioridad porque no era ni programable ni «Turing completa».

Un sistema Turing completo es aquel que tiene un poder computacional equivalente a la máquina universal de Turing. En otras palabras, el sistema y la máquina universal de Turing pueden emularse entre sí.

### 1.2.3. Z2

Konrad Zuse a partir del diseño de la Z1 construyó entre 1940 y 1941 a la Z2. Ya que crear una máquina mecánica presentaba algunas dificultades, a ésta se le añadieron relés telefónicos. Esta máquina fue destruida durante un bombardeo en 1940.

### 1.2.4. Z3

La computadora Z3, creada por Konrad Zuse en 1941, era electromecánica, estaba construida con relés y fue la primera máquina programable y completamente automática. Trabajaba en binario, leía los programas de tarjetas perforadas. Fue destruida en 1943 durante un bombardeo en Berlín (Rojas (1998)).

### 1.2.5. Colossus

La máquina Colossus fue diseñada originalmente por Tommy Flowers en la *Post Office Research Station* (Estación de Investigación de la Oficina Postal) a comienzos de 1944. Colossus fue uno de los primeros computadores digitales, empleados por los británicos para leer las comunicaciones cifradas alemanas durante la Segunda Guerra Mundial mediante Enigma. Se basaba en la idea de universalidad de la máquina de Turing, estaba compuesta por más de 1.500 tubos de vacío, la entrada de datos era por medio de tarjetas perforadas y los resultados se almacenaban en relés temporalmente hasta que se les daba salida a través de una máquina de escribir.

### 1.2.6. ASCC

La *Automatic Sequence Controlled Calculator* (ASCC) contruida por IBM y llamada Mark I (7 de agosto de 1944) por la Universidad de Harvard, fue una computadora electromecánica de propósito general utilizada en la Segunda Guerra Mundial (Figura 1.5).

El concepto original pertenece a Howard Aiken (Cohen (2000)).

### 1.2.7. Z4

Konrad Zuse diseñó la Z4 -que fue construida por su compañía *Zuse Apparatebau* en 1945- de manera similar al Z3 pero se mejoró significativamente en varios aspectos. La memoria consistía en palabras de coma flotante de 32 bits en lugar de 22 bits. Una unidad especial llamada *Planfertigungsteil* (unidad de construcción de programas), que perforaba las cintas de programas, facilitaba mucho la programación y corrección de programas para la máquina mediante el uso de operaciones simbólicas y celdas de memoria. Los números se introducían y salían con coma flotante decimal pero el funcionamiento interno era en binario. La máquina tenía un amplio repertorio de instrucciones que incluía raíz cuadrada, MAX, MIN y seno. Las pruebas condicionales incluían pruebas de infinito.

Los primeros sistemas de cómputo electrónico similares a los actuales datan de los años 1945-1950.

Estas computadoras aún no tenían sistema operativo y el usuario que las operaba estaba muy altamente capacitado y cumplía las funciones de usuario, operador, programador y administrador.

Era necesario conocer exactamente el hardware, porque a cada programa había que darle todas las especificaciones físicas, mediante clavijas y paneles, para que ejecutara correctamente, es decir, especificar posiciones de memoria donde almacenar cada dato y además encargarse de la entrada y salida en los periféricos de la época.

A este tipo de máquinas se las conoce como «computadoras controladas por programa» (*program-controlled computers*). La «ventaja» es que disponía de toda la máquina para sí de manera real y efectiva; en nuestros tiempos, es tan sólo una ilusión que nos muestran los sistemas operativos modernos.

Por la época en la cual nacieron y porque eran tremendamente costosas, eran construidas en establecimientos militares y con fines bélicos.

### 1.2.8. ENIAC

La *Electronic Numerical Integrator And Computer* (Computador e Integrador Numérico Electrónico), que vemos en la Figura 1.6, fue una de las primeras computadoras de propósito general, presentada el 15 de febrero de 1946.

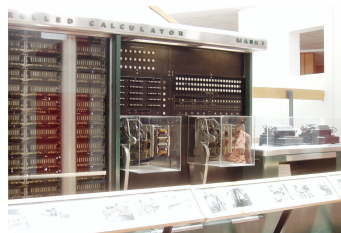
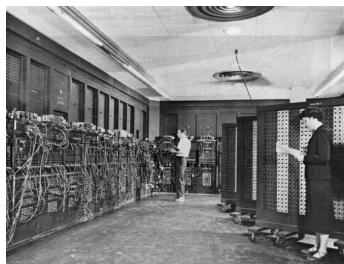


Figura 1.5.: Harvard Mark I (1944)



**Figura 1.6.:** ENIAC (1946)

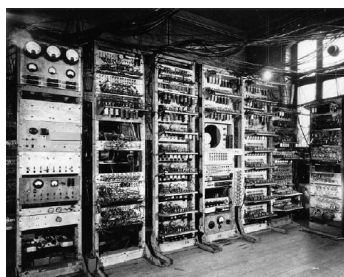
Era «Turing completa», digital y susceptible de ser reprogramada para resolver una «extensa clase de problemas numéricos».

Pero en realidad, como ya adelantamos, fue diseñada para calcular tablas de tiro de artillería destinadas al ejército de Estados Unidos.

Si bien los ingenieros John Presper Eckert y John William Mauchly se llevaron el mérito por la construcción, las encargadas de programarla eran seis mujeres.

### 1.2.9. SSEM o Manchester Baby

La Máquina Experimental de Pequeña Escala o *Small-Scale Experimental Machine* (SSEM), apodada «*Manchester Baby*» y construida en junio de 1948, fue la primera computadora con programa almacenado.



**Figura 1.7.:** SSEM (1948)

El primer programa que ejecutó fue el de encontrar a todos los divisores de  $2^{18}$  (262144) simplemente probando dividir entre todos los números enteros menor que él, hasta el 1, cuyo resto fuera cero.

La SSEM tenía una longitud de palabra de 32 bits y una memoria de 32 palabras.

Como fue diseñado para ser el computador de programa almacenado con la estructura más simple posible, las únicas operaciones aritméticas implementadas en hardware eran la resta y la negación; las otras operaciones aritméticas fueron implementadas en software.

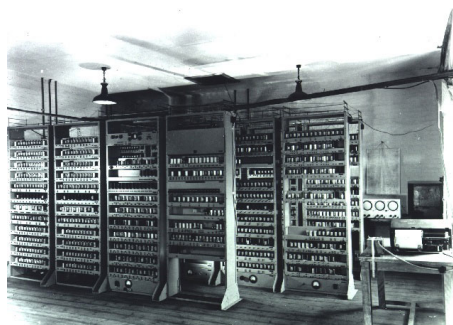
La máquina no estaba destinada a ser una computadora práctica, sino que fue diseñada como un banco de pruebas para el tubo Williams, la primera memoria de acceso verdaderamente aleatorio.

Tan pronto como demostró la viabilidad de su diseño, se inició un proyecto en la universidad para desarrollarlo en una computadora más utilizable, el Manchester Mark 1.

### 1.2.10. EDSAC

La Calculadora Automática de Almacenamiento Electrónico de Retraso (*Electronic Delay Storage Automatic Calculator*, EDSAC), que vemos en la Figura 1.8, fue una de las primeras computadoras británicas.

Inspirada en el primer borrador inicial de un informe de John von Neumann sobre el EDVAC, la máquina fue construida por Maurice Wilkes y su equipo en el Laboratorio de Matemáticas de la Universidad de Cambridge en Inglaterra. EDSAC fue la segunda computadora electrónica digital de programa almacenado que entró en servicio regular.



**Figura 1.8.:** EDSAC (1949)

El Manchester Baby precedió a EDSAC como una computadora de programa almacenado, pero fue construido como un banco de pruebas para el tubo Williams y no como una máquina para uso práctico. Sin embargo, el Manchester Mark I de 1949 (que no debe confundirse con el prototipo de 1948, el Baby) estaba disponible para uso general por otros departamentos universitarios y la empresa Ferranti en abril de 1949.

Más tarde, el proyecto fue apoyado por J. Lyons & Co. Ltd., una empresa británica, que fue recompensada con la primera computadora comercialmente aplicada, LEO I, basada en el diseño EDSAC. El trabajo en EDSAC comenzó durante 1947 y ejecutó sus primeros programas el 6 de mayo de 1949, cuando calculó una tabla de números cuadrados y una lista de números primos.

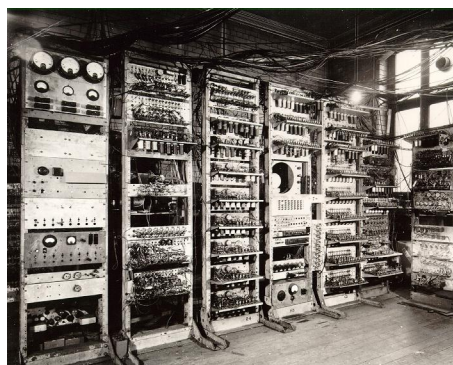
Si se utiliza una definición de computadora pensando en máquinas similares a las actuales y que hayan sido algo más que prototipos, ésta podría considerarse como la primera, adelantándose por muy pocos meses a la Manchester Mark I.

### 1.2.11. Manchester Mark I

Manchester Mark I o *Manchester Automatic Digital Machine* (MADM) fue una de las primeras computadoras de programa almacenado, construida a partir de agosto de 1948 en la misma Universidad de Manchester en el Reino Unido. En abril de 1949 ejecutó un programa para encontrar números primos de Marsenne, entre el 16 y 17 de junio de 1949. La vemos en la Figura 1.9.

La Manchester Mark I no tenía aún lenguaje ensamblador (los programas se debían escribir en binario), ni sistema operativo.

La computadora es especialmente importante históricamente debido a su inclusión pionera de registros de índice (*index registers*), una innovación que facilitó la



**Figura 1.9.:** Manchester Mark I

lectura secuencial de un programa a través de una serie de palabras en la memoria.

A esta máquina le siguió la Ferranti Mark 1, reconocida como la primera computadora de propósito general y disponible comercialmente en el mundo.

### 1.2.12. EDVAC

Los inventores de ENIAC, John Mauchly y J. Presper Eckert, propusieron la construcción de la EDVAC (**E**lectronic **D**iscrete **V**ariable **A**utomatic **C**omputer) en agosto de 1944. En abril de 1946 se firmó un contrato para construir la nueva computadora, que fue entregada al Laboratorio de Investigación y Balística (*Ballistic Research Laboratory*) en 1949.

EDVAC era una computadora serial binaria con suma, resta, multiplicación, división programada y verificación automática con una memoria serial ultrasónica con capacidad para 1000 palabras de 34 bits. El tiempo de adición promedio de EDVAC fue de 864 microsegundos y su tiempo de multiplicación promedio fue de 2900 microsegundos.

La famosa monografía sobre la EDVAC de John Von Neumann, «*First Draft of a Report on the EDVAC*», (von Neumann (1945)) propuso la mejora principal de su diseño que incorpora el concepto principal de «programa almacenado» que ahora llamamos la arquitectura de Von Neumann. Este fue el almacenamiento del programa en la misma memoria que los datos. Las computadoras británicas EDSAC en Cambridge y Manchester Baby fueron las primeras computadoras que siguieron este diseño. Y ha sido seguido por la gran mayoría de todas las computadoras fabricadas desde entonces. Tener el programa y los datos en diferentes memorias ahora se llama la **arquitectura de Harvard** para distinguirlo.

### 1.2.13. Ferranti Mark 1

La computadora Ferranti Mark 1 o Manchester Electronic Computer o Manchester Ferranti, de febrero de 1951, estaba basada en la Manchester Mark 1, con mejoras en el tamaño del almacenamiento primario y secundario, un multiplicador más rápido e instrucciones adicionales. Uno de ellos permitía emitir un sonido (retroalimentación auditiva) que, al alterarlo, tocaba melodías sencillas. La sucedió la Ferranti Mercury.

### 1.2.14. UNIVAC I

La UNIVAC I (**UNIV**ersal **A**utomatic **C**omputer **I**, Computadora Automática Universal I) fue la primera computadora comercial fabricada en Estados Unidos, entregada el 31 de marzo de 1951 a la oficina del censo. Fue diseñada principalmente por J. Presper Eckert y



John William Mauchly, autores de la primera computadora electrónica estadounidense, la ENIAC. Durante los años previos a la aparición de sus sucesoras, la máquina fue simplemente conocida como «UNIVAC». Se donó finalmente a la universidad de Harvard y Pensilvania.

La Z3 de 1941 fue el primer computador operativo, los posteriores fueron proyectos militares como el ENIAC de 1946 u ordenadores de uso específico, mientras que el UNIVAC I compitió por ser el primer ordenador de uso general vendido comercialmente, pero perdió por un par de meses ante el británico Ferranti Mark I vendido en febrero de 1951.

### 1.2.14.1. **Compilador A-0**

De forma paralela al desarrollo del hardware, está también el desarrollo de los lenguajes de programación. Grace Murray Hopper fue una científica de la computación, programadora de la Harvard Mark I (ASCC), y se le reconoce haber desarrollado el primer «compilador»<sup>1</sup> para un lenguaje de programación (compilador «A», 1951). Posteriores mejoras llevaron al compilador a la versión B-0 (FLOW-MATIC), que fue tomado como base para COBOL.

El sistema A-0 (*Arithmetic Language version 0*), escrito por Grace Murray Hopper entre 1951 y 1952 para el UNIVAC I, fue una de las primeras herramientas relacionadas con el compilador desarrolladas para computadoras electrónicas. El A-0 funcionaba más como un cargador o enlazador que la noción moderna de un compilador. El programa era especificado como una secuencia de subrutinas y argumentos. Las subrutinas se identificaban mediante un código numérico y los argumentos de las subrutinas se escribían directamente después de cada código de subrutina. El sistema A-0 convirtió la especificación en código máquina que podía ser introducido en el ordenador por segunda vez para ejecutar dicho programa.

### 1.2.15. **ORDVAC**

La ORDVAC o *Ordnance Discrete Variable Automatic Computer* fue la primera computadora construida por la Universidad de Illinois para el Ballistic Research Laboratory en Aberdeen Proving Ground. Estaba basado en la arquitectura IAS desarrollada por John von Neumann, que fue conocida como la Arquitectura de von Neumann. La ORDVAC fue la primera computadora en tener un compilador: el Forast (cf. Campbell y Beck (1965)). Empezó a funcionar en la primavera de 1951 en Aberdeen Proving Ground en Maryland.

Era una máquina asíncrona, lo que significa que no estaba regulada por un reloj central: cada instrucción empezaba cuando acababa la anterior.

Estaba diseñada para calcular trayectorias balísticas para el Ejército de Estados Unidos.

---

<sup>1</sup>si bien su funcionamiento no se ajustaba a lo que hoy entendemos como «compilador».

### 1.2.16. IBM 704

La máquina IBM 704 fue la primera computadora producida en masa con hardware basado en aritmética de coma flotante, introducida por IBM en abril de 1954 (cf. IBM (1954)).

Fueron inicialmente desarrollados para la 704 los lenguajes de programación FORTRAN y LISP, así como el MUSIC 1, que se puede decir es el primer programa de música para computadora, de Max Mathews.

Como la programación de la entrada y salida era una tarea bastante común y repetitiva, y para ayudar a la programación en aquellas épocas, esas rutinas de entrada y salida comenzaron a escribirse a modo de lo que hoy llamaríamos «bibliotecas» (*libraries*) para que pudieran volver a utilizarse por otros programas. Estas bibliotecas fueron las semillas germinales del sistema operativo.

Las computadoras eran mucho más costosas que en estos tiempos, de manera que había que encontrar formas de hacerlas rendir más. También era necesario llevar la contabilidad del tiempo consumido por los programas, del uso de la CPU, la memoria, el espacio de almacenamiento y el papel impreso. Todas estas funciones se fueron incorporando paulatinamente al sistema operativo. De a poco también fue delineándose la figura del operador, distinguiéndola de la del usuario/programador, que todavía seguía siendo el mismo.

## 1.3. Los sistemas operativos

### 1.3.1. GM-NAA I/O

Se acepta comúnmente que el GM-NAA I/O haya sido el primer sistema operativo de la historia (1956) que ejecutaba en una IBM 704. Se los suele denominar más propiamente «monitores» o «programas monitores» del computador. Eran desarrollados por los mismos programadores para poder trabajar más cómodamente sobre el equipo. Eran específicos para la máquina sobre la cual ejecutaban.

La función principal del GM-NAA I/O era ejecutar automáticamente un nuevo programa una vez finalizado el que se estaba ejecutando (procesamiento por lotes). Estaba formado por rutinas compartidas a los programas que proporcionaban acceso común a los dispositivos de entrada/salida.

La empresa IBM (*International Business Machines*) tuvo un gran protagonismo en esta época. Desarrollaba el hardware y el software para el equipo, tanto el sistema operativo como el software de soporte. En la década de 1960 IBM se abocó al desarrollo de la serie System/360. Todos los modelos de la serie usaban la misma arquitectura de instrucciones, para los cuales IBM intentó desarrollar el mismo sistema operativo, el OS/360, de 1964.

### 1.3.2. Multics

Un hito muy importante en la historia de los sistemas operativos es la aparición de Multics (***M**ultiplexed **I**nformation and **C**omputing **S**ervice* o «Servicio de Computación e Información Multiplexado») en el año 1969, luego de varios años de desarrollo.

Multics fue un influyente sistema operativo de tiempo compartido que se basaba en el concepto de una memoria de un solo nivel. Prácticamente todos los sistemas operativos modernos están fuertemente influenciados por Multics - a menudo a través de Unix, que fue creado por algunas de las personas que habían trabajado en Multics - ya sea directamente (Linux, macOS) o indirectamente (Windows NT).

Su licencia de uso originalmente fue propietaria pero se liberó en 2007. La última versión (12.6f) apareció en diciembre de 2016.

Era un proyecto muy ambicioso que no tuvo el éxito esperado pero tuvo mucha influencia en la historia.

### 1.3.3. UNIX

Los Laboratorios Bell (*Bell Labs*), de AT&T, participaron del proyecto Multics pero se retiraron para utilizar sus recursos en otros proyectos. Ken Thompson trabajaba en Bell Labs y junto con Dennis Ritchie comenzaron a desarrollar en 1969 otro sistema en una computadora PDP-7, al que Brian Kernighan jocosamente llamó Unics -en contraste con Multics-, pero luego cambió el nombre a UNIX.

Contrariamente a lo que se decía en aquella época acerca de que un sistema operativo, dada su complejidad y dependencia con el hardware, debía escribirse en assembler por razones de desempeño, ellos lo escribieron en lenguaje C, logrando que UNIX fuera mucho más portable que otros sistemas operativos que lo precedieron y sin problemas de desempeño. UNIX se distribuía originalmente con su código fuente, principalmente en universidades, por lo que se lo utilizó mucho como material didáctico para la enseñanza del sistema operativo.

### 1.3.4. VMS

El 25 de Octubre de 1977 aparece la primera versión del sistema operativo VMS (Virtual Memory System) de la empresa DEC (Digital Equipment Corporation). Era un sistema operativo de 32 bits; su diseñador fue David Cutler, quien fue luego diseñador del Windows NT.

Poco antes de 1980 aparecen las computadoras hogareñas; sus procesadores eran de 8 bits, a diferencia de los 32 que estaban usando la VAX y otros.

### 1.3.5. Linux

Como veremos en el artículo «El Proyecto GNU», de Richard Stallman, en enero de 1984, él renunciaba a su trabajo en el Laboratorio de Inteligencia Artificial del MIT para comenzar a desarrollar software para el Proyecto GNU, cuyo anuncio se había efectuado el 27 de septiembre de 1983.

En el año 1987 aparece el sistema operativo Minix, creado por Andrew S. Tanenbaum en la Vrije Universiteit de Amsterdam para poder ejemplificar los principios de un sistema operativo. Es un sistema operativo tipo UNIX, que está basado en una arquitectura micronúcleo y su intención es netamente educativa.

El 14 de mayo de 1991 sale a la luz el núcleo del sistema operativo Linux versión 0.01, desarrollado por Linus Torvalds. Era un sistema operativo tipo UNIX, de arquitectura monolítica, y ejecutaba en procesadores compatibles con el 80386 de Intel y hardware de PC. No fue sino hasta el 14 de marzo de 1994 que apareció la siguiente versión de hito, Linux 1.0.

Varios elementos se conjugaron en la historia de los sistemas operativos para que apareciera Linux. Por una parte, el Proyecto GNU, que había iniciado Richard Stallman como se dijo, había ya creado una gran cantidad de herramientas y aplicaciones tales como el compilador GNU C, bibliotecas y editores de texto. El sistema de ventanas X Window no había sido desarrollado por el Proyecto GNU; aún así, tenía una licencia compatible con el mismo, pero el desarrollo del núcleo del sistema operativo GNU Hurd estaba atascado e incompleto. Por otra parte, estaba disponible el código fuente del sistema operativo Minix, pero no estaba permitida su modificación y redistribución. Tanenbaum se mostraba renuente a introducirle modificaciones porque quería mantenerlo sencillo para que siguiera teniendo utilidad a sus fines educativos.

Por ello, Linus Torvalds, siendo aún estudiante en la Universidad de Helsinki en Finlandia, se decide a desarrollar un sustituto de Minix pero con una licencia GNU (a partir de la versión 0.99) y arquitectura monolítica.

En definitiva, lo que Linus Torvalds desarrolló fue el núcleo, el cual en sí mismo no es usable, sino a través de sus aplicaciones, las cuales provenían en su mayoría del Proyecto GNU.

Y como tanto el núcleo como las bibliotecas y aplicaciones para Linux (o sistema GNU/Linux) se ofrecen en forma de archivos fuente, no ejecutables, el sistema no sería usable si no fuera por las **distribuciones**, que son una gran cantidad de software ya compilado por alguna organización de personas y que se entrega en algún medio de almacenamiento, por ejemplo DVD y que, no obstante esto, también está disponible en sitios de Internet a los que se les llama **repositorios**, que suelen contener paquetes de software y actualizaciones de los mismos. Las primeras distribuciones comenzaron a aparecer en 1992, MCC Interim Linux en febrero, luego TAMU, SLS e Yggdrasil.

### 1.3.6. Evolución de Windows

En abril de 1980, Tim Paterson comenzó a trabajar en el sistema operativo QDOS (Quick and Dirty Operating System), copiando las llamadas a sistema de CP/M y simplificándolo. El sistema debía correr sobre un procesador 8086 de Seattle Computer Products (SCP), que tenía originalmente sólo al BASIC de Microsoft. Este sistema operativo pronto cambió de nombre a 86-DOS y fue posteriormente comprado por Microsoft para comercializarlo bajo el nombre de MS-DOS. SCP le había solicitado esto a Paterson porque sus ventas languidecían y DRI (Digital Research Incorporated) tardaría un tiempo en lograr una versión de CP/M para él mismo.

Microsoft, en diciembre de 1980, le compró una licencia no exclusiva a SCP del sistema 86-DOS y le pidió a Tim Paterson que lo portara al IBM-PC, que usaba un procesador 8088 de Intel. Luego, Microsoft le compró a SCP la totalidad de los derechos del 86-DOS y, a su vez, le otorgó una licencia a IBM para ese computador. IBM le encontró unos 300 errores, escribió los manuales y llamó a ese sistema operativo IBM Personal Computer Disk Operation System 1.0, o PC-DOS 1.0, y lo lanzaron en 1981.

A partir de la misma fecha, Microsoft comenzó a comercializar el mismo sistema operativo bajo el nombre de MS-DOS, pero conservando la totalidad de los derechos y el código fuente.

El 20 de noviembre de 1985, Microsoft presentó a Windows 1.0, que era un ambiente de operación gráfico de 16 bits, es decir que no era un sistema operativo en sí mismo, era una aplicación que ejecutaba sobre sistema operativo DOS.

En el año 1988 aparece Windows 2.0, como un ambiente de operación gráfico de 16 bits, que ejecutaba como una aplicación de MS-DOS. Sus aplicaciones eran la calculadora, el calendario, el reloj y el block de notas, entre otras.

Windows 3.0 aparece el 22 de mayo de 1990 con mejoras técnicas, para permitir un mejor uso de las capacidades de administración de la memoria de los procesadores 80286 y 80386 de Intel. Aparecen el Administrador de Programas (Program Manager), el Administrador de Archivos (File Manager) y el juego de cartas Solitario.

En marzo de 1992 aparecería Windows 3.1. Windows 3.11 no fue una versión aislada de la serie sino que fue una actualización de Windows 3.1. Pero siempre como aplicaciones gráficas que ejecutaban con MS-DOS como sistema operativo subyacente.

Windows NT 3.1 fue el primer sistema operativo de la línea NT (New Technology) para servidores y computadoras de escritorio y fue hecho público por Microsoft el 27 de julio de 1993. El número de versión fue elegido así para hacerlo coincidir con Windows 3.1 pero, a diferencia de éste, no fue desarrollado tomando como base a MS-DOS. Para su desarrollo, Microsoft contrató a un grupo proveniente de DEC (Digital Equipment Corporation), liderados por Dave Cutler; por eso, muchos elementos de su diseño reflejan aspectos interiores del VMS y el RSX-11. Fue un diseño y desarrollo nuevo, es decir que no se utilizó la base de código escrita para Windows 3.x.

Microsoft continuaría desarrollando ambas líneas de sistemas operativos durante varios años, hasta que finalmente no se desarrolló más la línea de DOS y Windows 3.x y sólo se siguió con la línea de NT.

La siguiente versión de la línea NT fue Windows NT 3.5, la cual se conoció el 21 de septiembre de 1994. El objetivo principal era que el usuario lo percibiera más rápido que el anterior. Nueve meses después, el 30 de mayo de 1995, aparece Windows NT 3.51 al cual Microsoft le dió soporte hasta el 31 de diciembre de 2001.

Sobre la otra línea, el 24 de agosto de 1995, Microsoft presenta a Windows 95 como un sistema operativo basado en una GUI, a diferencia de sus predecesores que eran ambientes de operación gráfica, es decir, aplicaciones que ejecutaban sobre el sistema operativo MS-DOS. Pero al parecer se trataba de un MS-DOS 7.0 con un Windows 4.0, es decir, otra aplicación gráfica pero finamente enlazada con un sistema operativo MS-DOS, porque había una manera de instalarlo con otra versión de DOS, que era la de Digital Research Inc. -conocida como DR-DOS-. Caldera, una distribución de Linux que en aquella época tenía los derechos de DR-DOS, demandó a Microsoft: ambas compañías llegaron a un acuerdo económico extra judicial y la demanda fue retirada.

El 29 de julio de 1996 hace su aparición Windows NT 4.0, que es un sistema operativo de núcleo híbrido, para procesadores de 32 bits, con desalojo, gráfico y orientado a las empresas. Está diseñado para trabajar ya sea con uniprosesores como multiprosesores simétricos. Una diferencia significativa con las versiones previas de Windows NT es que el Graphics Device Interface, o GDI, está incorporado dentro del núcleo para acelerar la GUI, dando como resultado una mejora sustancial en el desempeño, comparado con Windows NT 3.51.

El 25 de junio de 1998, Microsoft presentó a Windows 98 como un sistema operativo gráfico sucesor de Windows 95. Como tal, era un producto monolítico híbrido de 16 y 32 bits, basado en MS-DOS. Fue el primero en utilizar el Windows Driver Model o WDM, que se siguió utilizando en Windows 2000 y Windows XP.

Windows 2000 fue una línea de sistemas operativos producidos por Microsoft para ser usados en computadoras de escritorio, notebooks y computadoras para servidores. Fue anunciado en diciembre de 1999, pero estuvo disponible el 17 de febrero del 2000 y fue el sucesor de Windows NT 4.0.

Por otra parte, el 14 de setiembre de 2000, Microsoft presentó a Windows Millennium Edition, más conocido como Windows ME, el cual fue el último de la línea de Windows monolíticos híbridos de 16 y 32 bits, basados en MS-DOS, es decir que fue el sucesor de Windows 98. Microsoft cesó su soporte el de julio de 2006.

En agosto de 2001 se anuncia, pero recién el 25 de octubre de ese mismo año, Microsoft presenta a Windows XP (eXPerience), que fue el sucesor de Windows 2000 Professional y de Windows ME. Es otro sucesor de la línea de NT, por lo tanto, no tiene código proveniente de Windows 98.

Microsoft anunció la aparición de Windows Vista el 22 de julio de 2005, el desarrollo se completó el 8 de noviembre de 2006 y quedó completamente disponible a partir del 30

de enero de 2007. El objetivo primordial establecido por Microsoft fue el de mejorar la seguridad del sistema operativo.

El desarrollo de Windows 7 se completó el 22 de julio de 2009 y se confirmó su fecha de venta oficial para el 22 de octubre de ese mismo año, junto a su equivalente para servidores Windows Server 2008 R2. Desarrollado para ser usado en PCs de escritorio y hogareñas, equipos portátiles, tablet PC, netbooks y media centers, fue concebido como una actualización incremental y focalizada de Vista y su núcleo NT 6.0 para mantener cierto grado de compatibilidad con aplicaciones y hardware en los que Vista ya era compatible. Es decir que no presentó el salto arquitectónico y de características que tuvo su predecesor Windows Vista respecto de Windows XP.

### 1.4. Propiedad intelectual y derecho de autor

#### 1.4.1. ¿Qué es la propiedad intelectual?

Es el conjunto de derechos de autor, personales (morales) y patrimoniales (económicos) que corresponden a los autores sobre las obras de su creación.

En otras jurisdicciones y para la OMPI (Organización Mundial de la Propiedad Intelectual) la expresión «*intellectual property*» engloba tanto los derechos de propiedad industrial (marcas, patentes, diseño industrial, denominaciones de origen) como los derechos de propiedad intelectual (derechos de autor y derechos afines).

En nuestra República Argentina, la propiedad intelectual está regulada por la LEY 11.723 sancionada el 28 de septiembre de 1933 y su posterior modificación: Propiedad intelectual, Ley 25036 que modifica los artículos 1º, 4º, 9º y 57º e incorpora el artículo 55 bis a la Ley 11723. Fue sancionada el 14 de Octubre de 1998 y Promulgada en Noviembre del mismo año.

#### 1.4.2. ¿Qué son los derechos de autor?

Los derechos de autor son de dos clases:

- Derechos morales, irrenunciables e inalienables, como el derecho de reconocimiento de autoría y el derecho de integridad de la obra, entre otros.
- Derechos económicos, transferibles y de duración limitada en el tiempo, básicamente los derechos de explotación, aunque hay otros (por ejemplo el derecho a remuneración por copia privada).

#### 1.4.3. ¿Quién es el autor de una obra?

Se considera autor a la persona natural que aparece como tal en la obra. En algunos casos previstos por la ley las personas jurídicas pueden tener algunos derechos económicos de

propiedad intelectual.

En las obras en colaboración los derechos pertenecen a todos los autores, sin embargo en las obras colectivas corresponden a la persona bajo cuya iniciativa y coordinación se edita y divulga la obra.

#### **1.4.4. ¿Qué obras son objeto de propiedad intelectual?**

Todas las creaciones originales literarias, artísticas o científicas expresadas por cualquier medio o soporte, tangible o intangible, actualmente conocido o que se invente en el futuro. Es decir: libros, folletos, impresos, conferencias; obras musicales, dramáticas, coreografías, obras cinematográficas y audiovisuales; esculturas, pinturas, dibujos, comics; proyectos, planos, maquetas, bocetos; mapas, fotografías y programas de computadoras.

#### **1.4.5. ¿Qué se excluye del objeto de propiedad intelectual?**

En general lo que no puede incluirse en la definición anterior, como son las ideas, la información, y todo conocimiento que es patrimonio común y no es susceptible de apropiación.

Expresamente la ley excluye las disposiciones legales y reglamentarias, sus correspondientes proyectos, las resoluciones de órganos jurisdiccionales, actos, acuerdos, deliberaciones y dictámenes de organismos públicos y traducciones oficiales de todos ellos.

#### **1.4.6. ¿Cuáles son los derechos de explotación?**

Los derechos de explotación son sinónimo de los derechos de patrimonio.

La ley establece como derechos de explotación los derechos de reproducción, distribución, comunicación pública y transformación.

Los derechos patrimoniales o de explotación facultan al autor a decidir sobre el uso de su obra, que no podrá llevarse a cabo sin su autorización, salvo en determinados casos previstos en la vigente Ley de Propiedad Intelectual, que se conocen como límites o excepciones (ejemplo: derecho de cita).

Estos derechos de explotación, que pueden cederse a terceros, son, según la Ley de Propiedad Intelectual:

- Reproducción: acto de fijación de la obra en un medio que permita su comunicación y la obtención de copias de todo o parte de ella.
- Distribución: acto de puesta a disposición del público del original o copias de la obra mediante su venta, alquiler, préstamo o cualquier otra forma.
- Comunicación pública: acto por el cual una pluralidad de personas pueda tener acceso a la obra sin previa distribución de ejemplares a cada una de ellas.



- **Transformación:** acto de traducción, adaptación y cualquier otra modificación de una obra en su forma de la que se derive una obra diferente. En el caso de las bases de datos, se considera transformación su reordenación.

La Ley también reconoce otros derechos de carácter patrimonial a los autores:

**Derecho de remuneración por copia privada:** la reproducción de una obra (divulgada en forma de libros o publicaciones asimiladas, entre otras, excepto los programas de computadora) realizada exclusivamente para uso privado del copista, mediante aparatos o instrumentos técnicos no tipográficos, originará una remuneración equitativa y única a favor de sus titulares de derechos de estas creaciones. Esta reproducción no tiene que contar con la autorización previa de sus titulares para su realización.

La remuneración compensatoria por copia privada se determinará para cada tipo de obra en función de los equipos, aparatos y materiales idóneos para realizar su reproducción. Debe ser abonada obligatoriamente por los fabricantes e importadores de estos instrumentos, y recaudada y repartida por las entidades de gestión.

Las copias para uso privado son las que se efectúan en el ámbito doméstico, sin fin lucrativo, ni uso colectivo ni se distribuyen mediante precio.

**Colecciones escogidas u obras completas:** la cesión de los derechos de explotación sobre sus obras no impedirá al autor publicarlas reunidas en colección escogida o completa.

### **1.4.7. ¿A quién corresponden los derechos de explotación?**

Los derechos de explotación corresponden exclusivamente al autor y no pueden ser realizados sin su autorización, salvo en los casos previstos en la ley.

### **1.4.8. ¿Qué son las excepciones a los derechos de autor?**

También llamados límites, son casos en los que la ley autoriza el ejercicio de actos de explotación sin necesidad de una autorización por parte del titular de los derechos.

### **1.4.9. ¿Qué es el dominio público?**

Cuando los derechos de explotación se extinguen, normalmente por el transcurso de su plazo de duración, las obras pasan al dominio público y pueden ser utilizadas por cualquiera siempre que se respete su autoría e integridad.

### **1.4.10. ¿Es obligatorio inscribir las obras en el Registro de la Propiedad Intelectual?**

No. Los derechos de autor nacen en el momento de la creación de la obra. El Registro es sólo un medio de protección y de prueba de los derechos.

### 1.4.11. ¿Qué es el «copyright»?

Es la fórmula anglosajona para designar únicamente los derechos de explotación de una obra, no hace relación a los derechos morales. El símbolo © asociado a un nombre indica titularidad de derechos de explotación. Normalmente, aunque no necesariamente, va seguido de la expresión «todos los derechos reservados». «Copyright» es un vocablo formado por los términos ingleses «copy» (copia) y «right» (derecho, en el sentido jurídico). Sucede que «right» también significa lado derecho, lo que propicia el juego de palabras al sustituirlo por lado izquierdo, que en inglés se dice «left».

### 1.4.12. ¿Qué es el «copyleft»?

Es un movimiento social y cultural alternativo al sistema tradicional del *copyright* que aboga por el uso de licencias libres para compartir y reutilizar las obras de creación. «Left» es también el participio del verbo «to leave» (abandonar), de ahí el juego de palabras. Las cláusulas copyleft exigen que el software derivado se distribuya con la misma licencia y en las mismas condiciones que el original.

Una licencia es una cesión de derechos por parte del propietario del mismo hacia el licenciatario.

Toda licencia tiene dos aspectos:

- Aseguran los derechos del usuario, detallando las autorizaciones para el uso del software.
- Protege los derechos del titular, al especificar los derechos no cedidos y las condiciones que debe cumplir el usuario.

Hay diferentes tipos de licencias libres entre las que se puede elegir según el ámbito que se trate (software, obra científica, música, arte,...). Las más utilizadas son las licencias Creative Commons de origen norteamericano, pero también existen otros modelos, por ejemplo las licencias Colouriuris, de iniciativa española, que se están abriendo paso en la comunidad de habla hispana.

### 1.4.13. ¿Qué son las licencias Creative Commons?

Son licencias en las que el autor otorga a la comunidad una mayor libertad de uso sobre su obra aunque bajo determinadas condiciones. Estas condiciones son escogidas por el propio autor, de modo que frente una obra con «todos los derechos reservados» las licencias Creative Commons proponen «algunos derechos reservados».

### 1.4.14. Software libre o «free software»

En este contexto «free» quiere decir libre, no gratis. Libre como el sol cuando amanece, no como en barra libre. El término fue acuñado por Richard Stallman quien además fundó la Free Software Foundation (FSF) (cf. GNU Project (1983)). Para que un software sea considerado libre, el autor (en uso de sus derechos patrimoniales sobre su obra) debe otorgar la licencia o permiso de uso al usuario. Estos permisos, licencias o «libertades», como las llama la FSF son:

1. Usar el programa. Sin restricciones por objetivo, modo de uso, ubicación, etc.
2. Estudiarlo y adaptarlo. Para ello se necesita acceso al código fuente.
3. Distribuir copias. Gratis o por un precio.
4. Hacer mejoras y publicarlas. Para compartir esas mejoras, gratis o por un precio.

Al distribuir copias (originales o modificadas) se debe hacer bajo los mismos términos, sin quitar ni alterar ninguna de estas cuatro libertades. No se puede crear software privativo a partir de software libre.

Estas libertades se aplican en la distribución pero no obliga a distribuir el software.

### 1.4.15. Software de fuente abierta «open source»

La principal diferencia entre las distintas licencias de software libre está en las condiciones que imponen a la redistribución del software.

La licencia «open source» fue redactada por Eric S. Raymond y es la que apoya la OSI (*Open Source Initiative*):

1. Redistribución libre
2. Código fuente
3. Obras derivadas
4. Integridad del código del autor
5. No discriminación de usuarios
6. No discriminación de actividad
7. Distribución de licencia
8. Independencia del producto
9. No limitar otro software
10. Neutra respecto a tecnología

La licencia de software libre y la de fuente abierta difieren en su filosofía. La primera pone énfasis en la libertad mientras que la segunda lo pone en los aspectos prácticos de contar con el código fuente.

### 1.4.16. Software privativo

Es lo contrario del software libre o de fuentes abiertas. Es aquel en el que falta alguna de las libertades de la FSF y/o no cumple alguno de los criterios de la OSI. En este sentido, no son software libre:

- Freeware
- Shareware
- Versiones «demo» limitadas
- Código fuente (solamente) disponible
- Cualquier licencia que prohíba la redistribución

## 1.5. Algunas comparaciones entre Windows y Linux

### 1.5.1. Históricas

Como ya vimos, el sistema operativo Multics (Multiplexed Information and Computing Service) fue todo un hito en la historia de los sistemas operativos. Era un proyecto muy ambicioso, ya en su nombre «Servicio de Computación e Información Multiplexado» se puede ver que se concebía a la misma como un servicio que se podría prestar como otros servicios públicos, tales como la energía eléctrica. Ese sistema operativo iba a ejecutar en grandes mainframes, y la gente iba a tener acceso al servicio a través de terminales conectadas al equipo. No tuvo el éxito esperado -tal vez por lo ambicioso de su objetivo- pero tuvo mucha influencia en la historia. Esto provocó que Ken Thompson y Dennis Ritchie se propusieran un objetivo mucho menos ambicioso, pero alcanzable, al que Brian Kernighan jocosamente le llamó Unics, y posteriormente UNIX.

A partir del año 1983, y durante varios años, tuve la oportunidad de trabajar en una VAX 11/780 con sistema operativo VMS; y debo admitir que cuando tuve la oportunidad de probar el UNIX en aquellos años, me decepcionó un poco. Me resultó sencillo, pero hasta podría decir elemental y rudimentario. Extrañaba la potencia del intérprete de comandos del VMS en el que se podían colocar los comandos de forma abreviada -siempre y cuando no resultara ambiguo- y si faltaba algún parámetro imprescindible para la correcta interpretación se lo preguntaba. Cada capa de software y cada componente del sistema tenía su propio conjunto de errores. De manera que si el comando que estaba ejecutando involucraba la ejecución de distintas capas o componentes del sistema -lo cual se cumplía para la mayoría de ellos- y se producía un error, se veían varios mensajes de error en la pantalla, cada uno proveniente de las distintas capas o componentes involucrados. Su assembler no sólo sacaba ventaja de cada aptitud del procesador sino que permitía redacciones propias de un lenguaje de alto nivel, lo que lo hacía muy potente. El número de versión como parte del nombre completo del archivo (por ejemplo:

«archivo.dat;1», «archivo.dat;2») era una ventaja que no ha sido incorporada a los sistemas operativos posteriores. No obstante todo esto, los pequeños elementos que, como ladrillos, construían al UNIX, la comunicación entre procesos y el enfoque de ver a todo como si se tratara de archivos, dejaban vislumbrar que se convertiría en un sistema que pasaría merecidamente a la historia.

Fueron muy audaces al animarse a desarrollar un sistema operativo escribiéndolo en lenguaje C, contrariamente a lo que se decía en aquella época acerca de que un sistema operativo debía escribirse en assembler. Esto logró que UNIX fuera mucho más portable que los otros sistemas operativos y, al distribuirlo originalmente con su código fuente en las universidades en las que se enseñaba Sistemas Operativos como materia, logró que se difundiera muchísimo y fuera tomado como modelo para explicar ciertos conceptos.

Un ejemplo notable de esto es el Berkeley Software Distribution, BSD o Berkeley UNIX. Esta portabilidad y licencia le permitió volverse la elección para una segunda generación de minicomputadoras y la primera generación de estaciones de trabajo o workstations.

Tan fuerte era esta influencia, no sólo en las bondades de UNIX y su portabilidad, sino sobre todo en su licencia que llevó a Richard Stallman a lanzar el Proyecto GNU, cuando se vio amenazada la libertad del UNIX. Esto provocó que el desarrollo de software ya no estuviera confinado a unas pocas personas, sino que provocó una gran avalancha de desarrollo de software, cada vez de mejor calidad porque era mejorado por otros programadores.

Otra prueba de lo expuesto es la creación del Minix, como sistema operativo tipo UNIX pero ejecutando esta vez sobre arquitecturas tipo IBM PC; acompañado de un libro de texto universitario para enseñar los principios de los sistemas operativos.

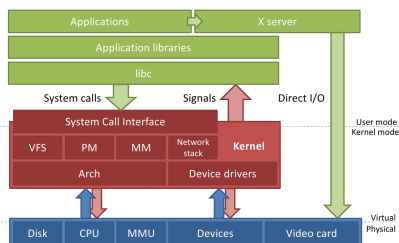
Todos estos elementos fueron los que se conjugaron en la historia de los sistemas operativos para que apareciera Linux. Por todo esto, vemos que Linux no sólo es un sistema operativo tipo UNIX, sino también que tiene una fuerte herencia del mismo y esto tiene muchas connotaciones a la hora de intentar compararlo con Windows. Hereda una fuerte tradición proveniente de los mainframes, las minicomputadoras y las workstations, frente a las PC que han sido dominio de Windows.

Como vimos, Windows nace como sistema operativo de PC, monousuario, con gran énfasis en la interfaz gráfica para que le resulte muy fácil y amistoso al usuario final con pocos conocimientos de informática. Si hay algo que agradecerle -gracias también a la alianza con Intel- es haber popularizado el uso de las computadoras. Operaciones que en un tiempo estuvieron reservadas a unos pocos altamente capacitados, ahora pueden efectuarlas intuitivamente muchísimas personas.

### 1.5.2. Los núcleos

Lo primero que vemos de un sistema operativo es su aspecto exterior, su interfaz con el usuario, que puede ser una interfaz de línea de comandos o una interfaz gráfica. Y son bastante reconocibles las interfaces gráficas de Windows y las de Linux (si bien cada

distribución de Linux presenta una interfaz que la diferencia de las otras). Pero eso no es lo que las diferencia en esencia, sino sus núcleos que, paradójicamente, son invisibles a los ojos. No obstante esto, algunas distribuciones se han esmerado en hacer parecer sus interfaces gráficas, escritorios y fondos de pantalla a las de Windows.



**Figura 1.10.:** Arquitectura del núcleo Linux

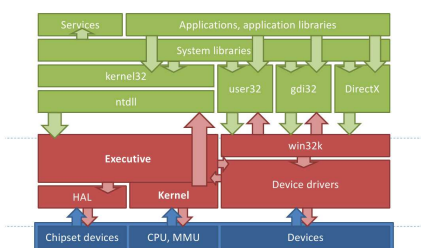
El núcleo de Linux tiene una estructura monolítica, como decíamos, en el que se concentran todas las funcionalidades, pero con una estructura de módulos cargables y descargables que lo hacen más funcional, como apreciamos en la Figura 1.10

El núcleo Linux no presenta una Interfaz Binaria de Aplicación (**Application Binary Interface** o ABI, en inglés), en cambio Windows NT posee una ABI del núcleo relativamente estable.

La gestión de procesos y el modelo de seguridad de Linux es semejante a la del UNIX, con soporte de red nativo en el núcleo, mediante *sockets* y tuberías. En cambio en Windows NT no hay soporte para redes en el núcleo mismo. Las llamadas a sistema «select» y «poll» que se usan en las redes no son llamadas a sistema dentro de Windows NT sino un ioctl al «sistema de archivos socket». Los *sockets* y las tuberías son provistas por dos sistemas de archivos especiales. Hay una capa adicional en modo usuario de abstracción de sockets: Winsock solía ser un componente de terceras partes (Trumpet Winsock).

El núcleo Linux provee una abstracción del sistema de archivos, que es el Sistema de Archivos Virtual (**Virtual File System** o VFS, en inglés).

La arquitectura del núcleo Windows NT es híbrida. El diseño del núcleo es casi idéntico al RSX-11 y VMS de DEC, con influencia de DOS, OS/2 y Windows 95. RSX-11, VMS y Windows NT fueron diseñados por el mismo ingeniero, David Cutler.



**Figura 1.11.:** Arquitectura del núcleo Windows

En el núcleo Windows NT no hay abstracción de dispositivos. La abstracción se provee mediante manejadores estándares de sistema, que son los manejadores de puerto o clase (*class or port drivers*, en inglés).

Con respecto a los sistemas gráficos, Windows NT tiene subsistemas gráficos y de ventanas nativos (ejecutando en modo núcleo) con una API estándar. En cambio en Linux el subsistema gráfico (X server) está en su propia categoría. El núcleo Li-

linux tiene «puertas traseras» para permitirle al servidor X hablarle directamente al hardware para mantener la complejidad de los manejadores de video fuera del ambiente sensible del modo núcleo.

Además hay características únicas en la arquitectura de Windows. Los dispositivos chip-set, tales como el temporizador o timer, el controlador de interrupciones, el administrador de energía, los colectores o buses, etc., son abstraídos por un componente del núcleo llamado Capa de Abstracción del Hardware (**H**ardware **A**bstraction **L**ayer o HAL, en inglés).

Windows tiene llamadas inversas a sistema (devolución de llamadas, retrollamadas o *callbacks*, en inglés). De manera que, por ejemplo, los subsistemas de ventana y gráficos pueden devolver la llamada a modo usuario.

Estas son algunas diferencias que podemos encontrar entre los núcleos de Linux y Windows, tal vez difíciles de terminar de entender en esta etapa de nuestro estudio, pero que se irán volviendo evidentes a medida que avancemos porque volveremos a hacer referencia a ellas desde distintas perspectivas.

### 1.5.3. Licencias de uso

Como ya sabemos, Windows utiliza una licencia propietaria y Linux otorga la Licencia Pública GNU o GPL por sus siglas en inglés. Esta diferencia de licencias de uso tiene muchas consecuencias, y muchas no son evidentes. La GPL es una licencia que permite libertad para usar un software de calidad sin quebrantar la ley. Microsoft siempre ha considerado que la única manera de brindarles a los usuarios un software de calidad es si éste tiene una licencia propietaria de uso de los ejecutables, sin entregarles los fuentes y prohibiéndoles su redistribución o modificación. Pero la realidad nos ha mostrado que esto no es tan así, mucho software con licencia GPL es de muy alta calidad y se lo utiliza para tareas críticas.

La GPL nos permite entender todo lo que pasa en el computador. En cambio el software con licencia propietaria no nos permite ver cómo se hacen ciertas tareas y, frente a un fallo, el usuario del software no puede tener una idea clara de las causas del mismo.

Otra consecuencia de la licencia libre que utiliza Linux son las distribuciones, que ya se cuentan por cientos. Y como son tan adaptables, cualquiera podría armarse su propia distribución.

En este sentido, tendemos a pensar en las distribuciones Linux como en las que se usan en las computadoras de escritorio y portátiles, pero la verdad es que todas las personas usan Linux sin saberlo: si utilizan una conexión ADSL para conectarse a Internet, o si usan Google o Facebook. Estos sitios utilizan Linux en sus servidores. Además, nuestro tráfico Internet pasa por muchos dispositivos de red que suelen tener Linux como su sistema operativo. Sin olvidar que el sistema Android de los teléfonos móviles está basado en Linux.

Muchas confusiones se han generado al traducir el término «*free*» del inglés a «gratis» en castellano, en vez de «libre», como sería más apropiado. Piense en esto: vender Linux no es ilegal, como tampoco lo es distribuir (gratuitamente o cobrando) copias compradas. Un Linux comercial no viola la licencia GPL ni tampoco el espíritu de libertad. Pero no caiga en la trampa: si usted sabe que está utilizando algún tipo de dispositivo que tiene

un Linux dentro, aún si no se lo cobraron y vino «gratuitamente», usted debería reclamar el software fuente, y el fabricante no debería ponerle ningún tipo de obstáculo porque eso sí que es una violación a la licencia GPL, cuyas reglas de juego el fabricante aceptó al incluir a Linux como software en su producto.

#### 1.5.4. Interfaces

Y tratando el tema de la plataforma sobre la cual ejecuta y en el caso particular de las computadoras de escritorio y portátiles, surge el tema de las interfaces. Como vimos, al hablar acerca de cómo funcionan sus respectivos núcleos, la interfaz gráfica (GUI) de Linux es opcional, en cambio la de Windows es un componente integral del sistema operativo. La interfaz gráfica de Linux es un conjunto de aplicaciones que ejecutan (obviamente, con el procesador en modo usuario) en el modelo cliente-servidor.

Muy ligado a la instalación optativa de la interfaz gráfica está la de las distintas distribuciones de Linux que ofrecen una versión «Server» y otra «Desktop» o «de Escritorio». Estas diferencias son más bien nominales porque los distintos programas que hacen que una distribución se comporte como un «Server» o un sistema para «Escritorio» se instalan y desinstalan como cualquier aplicación. Dicho de otro modo: usted puede instalar una distribución Linux de escritorio para luego agregarle las aplicaciones servidoras, y obtendrá un completo servidor con una interfaz gráfica para administrarlo cómodamente. No obstante esto, una buena distribución de Linux debería ser mucho más que un conjunto de paquetes instalados. Es decir, si se la presenta como una versión para servidores debería tener al núcleo compilado con todas las optimizaciones posibles para esos procesos servidores; la actualización del software debería ser automática porque será un equipo no interactivo. Por ende, debería tener un sistema de registro de actividades muy completo con notificaciones al administrador.

#### 1.5.5. Comparaciones mal hechas

Muchas veces podrá encontrarse con comparaciones mal hechas a la hora de mostrar las diferencias entre Linux y Windows. Linux es un sistema operativo multitarea real y multiusuario real, como tal habría que compararlo con alguna versión de Windows Server.

Cuando empecé a dar clases de Linux, allá por el año 1999, me tocaba «comparar» a Linux con Windows 95 o Windows 98, y me costaba hacerles entender a mis alumnos que estábamos comparando peras con manzanas. Ninguno de ellos, como era de esperar, utilizaba alguna versión de Windows NT Server en sus computadoras personales como para que entendieran cuáles eran las verdaderas diferencias. Como ya vimos, esas versiones de Linux «para escritorio» no son sistemas monousuario.

Tradicionalmente, primero se fabricaban las computadoras (con todos sus componentes) y se les desarrollaba un sistema operativo que se encargaba de gestionarles el hardware y ofrecía al programador una máquina más sencilla de programar. Pero desde hace



un tiempo ya, algunos fabricantes están desarrollando hardware para hacerlo funcionar con cierto sistema operativo (léase «Windows»). Esto no sólo pervierte el sano enfoque de las cosas sino que deliberadamente se hace para sacar de competencia a otro sistema operativo. La mayor o menor (también podríamos decir «la mejor o peor») detección de hardware no es culpa de Linux, como tampoco lo es la mejor o peor calidad de los manejadores o drivers. Es culpa de los fabricantes, no de Linux. Ellos seguramente se excusarán con el pretexto de que no pueden divulgar las especificaciones técnicas para que los programadores de los manejadores Linux hagan su trabajo porque quieren mantener sus derechos intelectuales sobre el hardware.

Otros también hacen malas comparaciones al afirmar que existen más y mejores juegos en Windows que en Linux, así como comparaciones acerca de la mejor o peor calidad de estos juegos en uno o en otro. Los juegos en Windows en su mayoría utilizan las bibliotecas DirectX privativas, no es una culpa atribuible a Linux.

Otra comparación mal hecha, en desmedro de Windows, se refiere a los virus. Es cierto que esto es todo un problema en Windows. Como con las brujas, se dice que también existen virus para Linux pero, en mi experiencia personal en todos estos años -desde 1995 hasta ahora-, nunca pude encontrarme con uno. El argumento de los virus funciona porque la gente usa erradamente a Windows con una cuenta con privilegios. Por suerte, esta situación puede cambiar si los usuarios cambian la forma de usar Windows, si luego de ser instalado queda habilitada una cuenta sin privilegios y para poder efectuar alguna operación que la requiera el sistema notifique al usuario. También ayudaría si los desarrolladores de software programaran las aplicaciones sin necesidad de escribir en carpetas que no fueran otras que las del usuario, ni intentaran efectuar ninguna otra operación que requiriera privilegios administrativos sobre el sistema. La larga tradición de estos dos sistemas operativos tiene mucho que ver. La herencia del UNIX sólo le permite al usuario «usar» el sistema pero no modificarlo. En cambio la herencia del DOS le hace sentir que la máquina es toda para él.

### 1.5.6. Un Windows libre

UNIX fue en sus comienzos un sistema operativo libre -antes de que la definición de «software libre» fuera acuñada por la FSF- y luego su licencia tornó a privativa. Esto provocó la reacción de Richard Stallman y fundó la FSF, como ya vimos. Se desarrolló mucho software libre y el núcleo Linux con esta licencia. El UNIX privativo continuó paralelamente durante varios años más.

En la actualidad existe mucho software libre que ejecuta sobre un Windows privativo, como en los comienzos de la FSF. ¿Qué pasaría si a alguien se le ocurriera desarrollar un núcleo Windows desde cero y distribuirlo bajo los términos de la licencia GPL? Tendríamos un Windows libre ejecutando aplicaciones libres.

Puede parecer una idea loca -no menos loca que la que se le ocurrió a Richard Stallman- pero tal cosa ya existe. Se trata del proyecto ReactOS.

Si bien no está suficientemente listo como para usarlo ampliamente, está bastante avanzado. Cuando esté completamente listo servirá como modelo para exponer otro enfoque de sistema operativo, otra forma de administrar el mismo hardware, otra manera de presentar al programador de aplicaciones una máquina sencilla más fácil de programar que el puro metal.

Muchas de estas diferencias entre Windows y Linux dejarán de serlas. Permanecerán las diferencias en cuanto a la arquitectura de sus núcleos y las que estén relacionadas con ellas.

Finalmente, si le gustan las tablas comparativas, puede ver una en Wikipedia bajo la entrada «Comparación de sistemas operativos» o «Comparison of operating systems» en inglés.