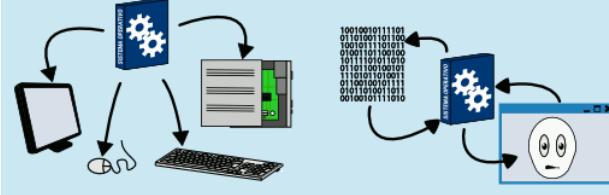


FUNCIONES DEL SISTEMA OPERATIVO



Funciones de un sistema operativo

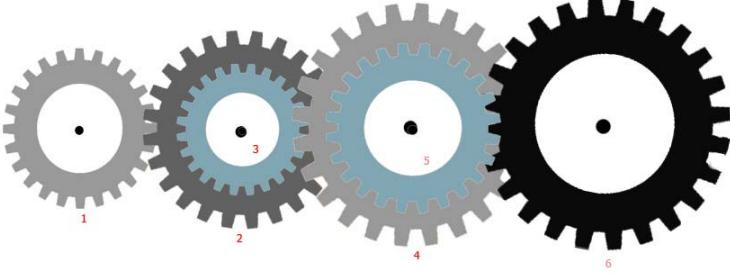
Alfredo Abad

ISO-01-03_FuncionesSO-Taxonomía.pptx

ISO-01-03

UA: 28-sep-2019

1



Controlar y gestionar el uso del hardware del ordenador: CPU, dispositivos de E/S, Memoria principal, tarjetas gráficas y el resto de periféricos

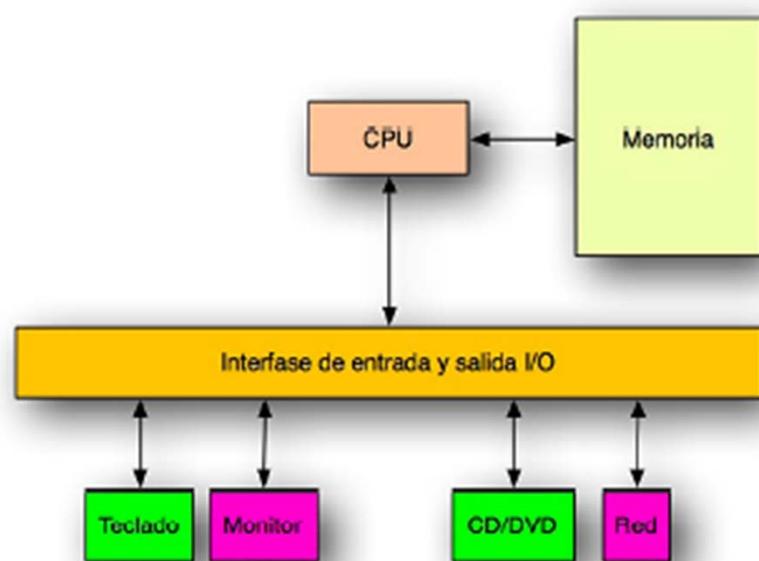
FUNCIONAMIENTO ORGÁNICO DEL HARDWARE DE UN ORDENADOR

2

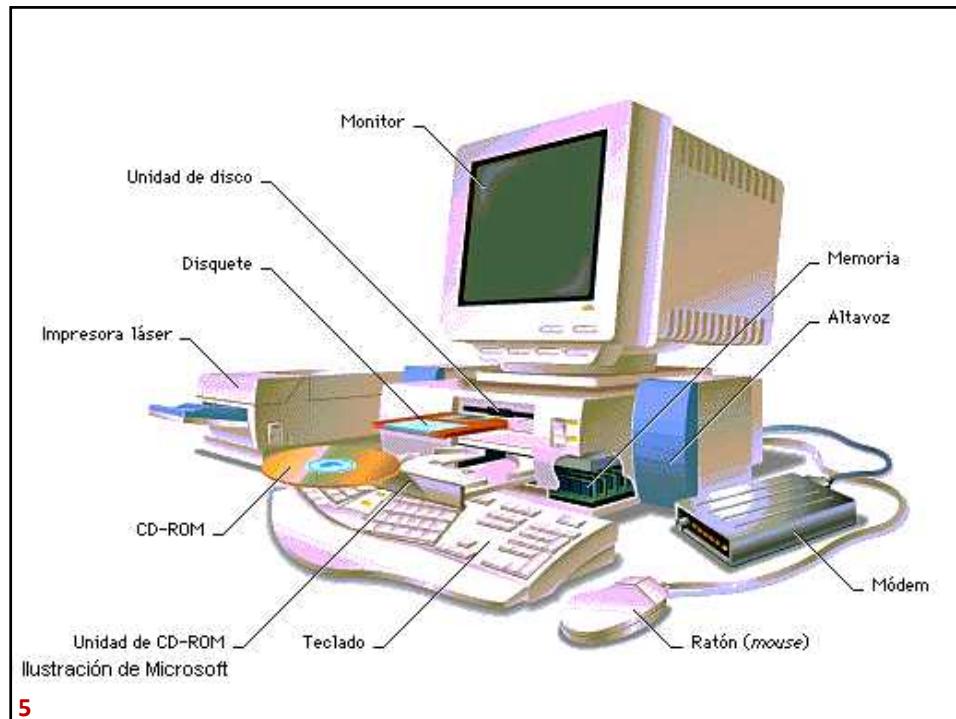
Hardware básico

- Unidad central
 - CPU, memoria, buses de comunicaciones
 - Controladores de dispositivos
 - Alimentación eléctrica
- Dispositivos de entrada y salida
 - Teclado, ratón y monitor
 - Impresoras
- Otros periféricos
- Conectividad de red

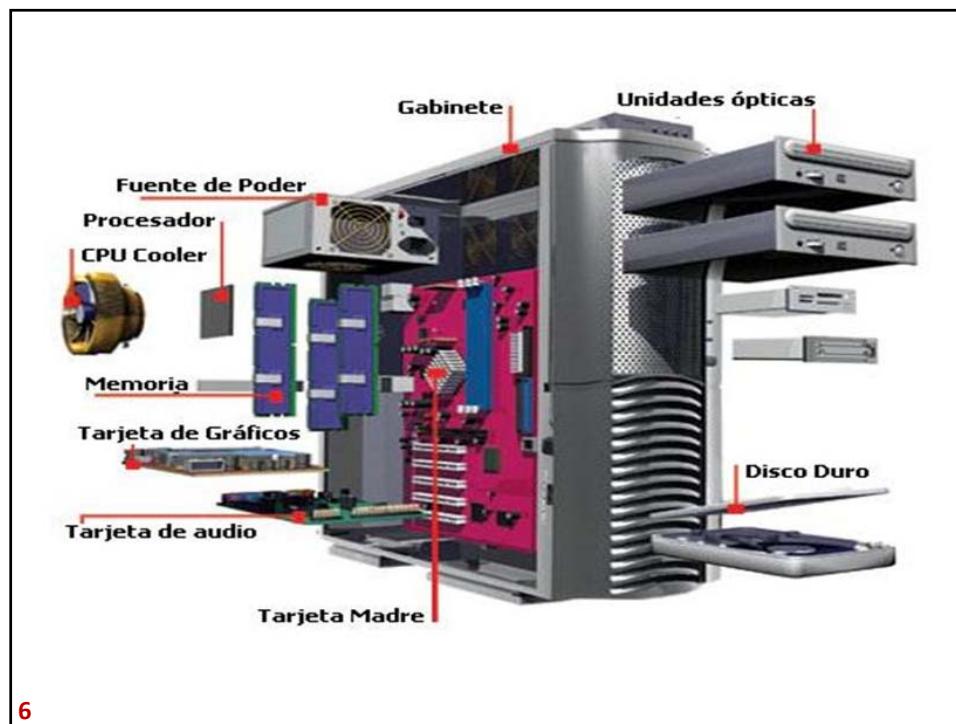
3



4



5

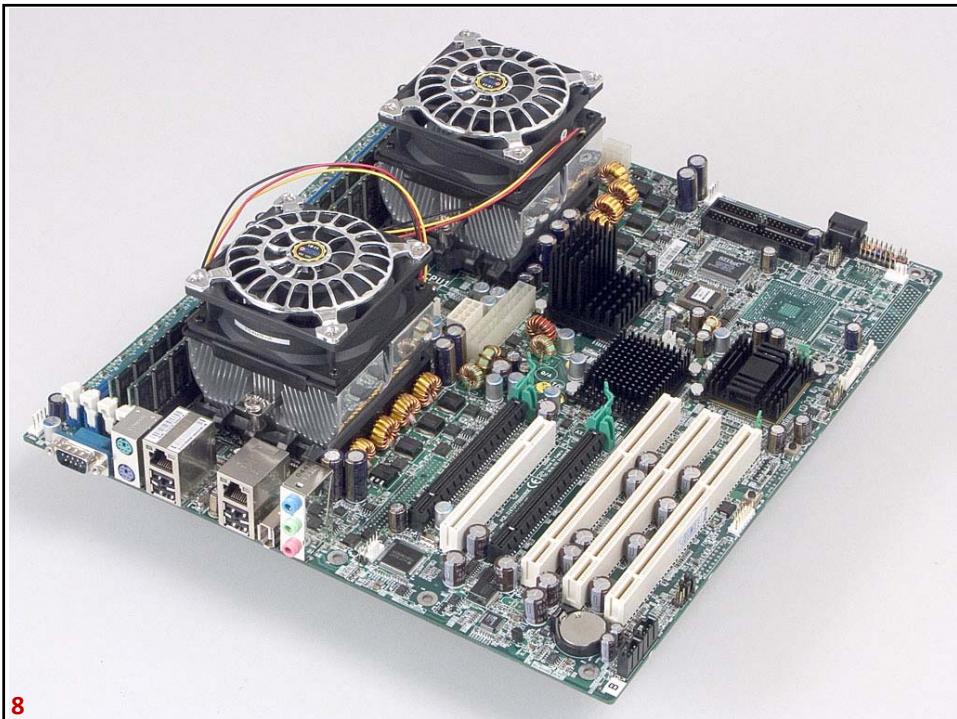


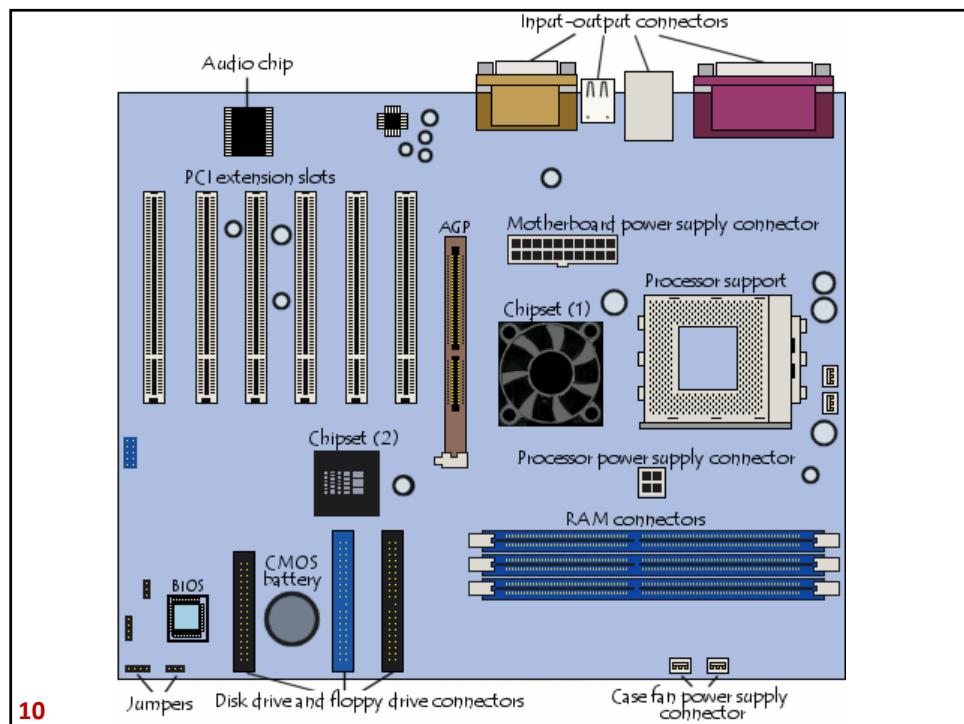
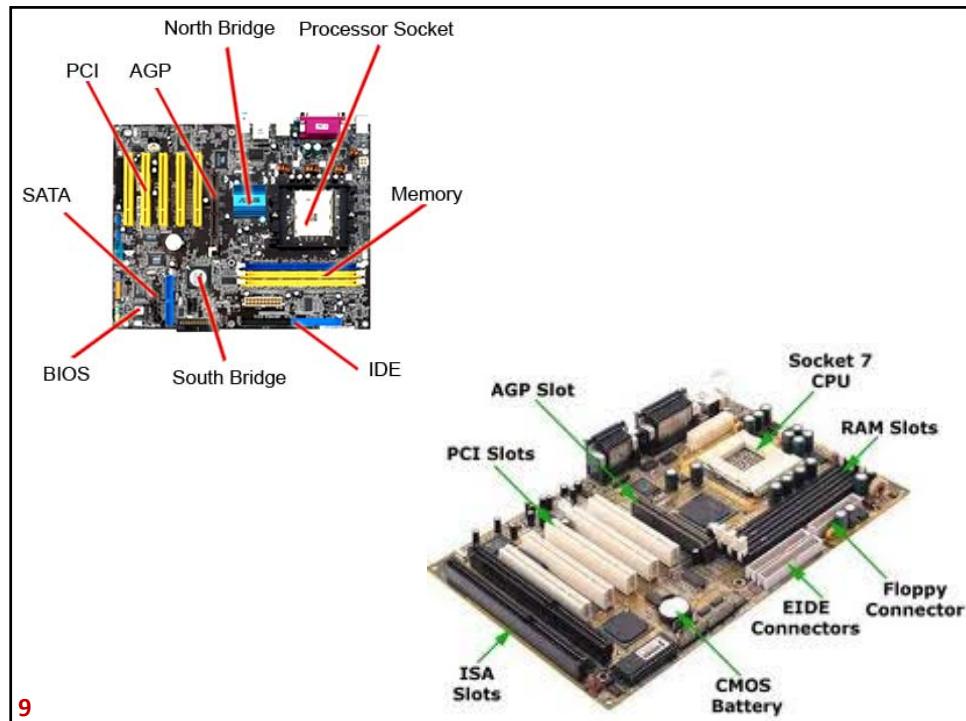
6

CPU

- Unidad central de proceso, procesador o microprocesador (*Central Processing Unit, CPU*)
 - Unidad aritmético-lógica (ALU)
 - Coprocesador, FPU, unidad de gráficos, etc.
 - Unidad de control (CU)
 - El reloj
 - Los registros
 - Unidad de manejo de memoria
 - Cachés
 - Buses de comunicaciones

7





Funciones de la CPU

- Proporciona la «programabilidad» del ordenador
 - Interpreta las instrucciones de programa
 - Procesa los datos
- Programa
 - Secuencia ordenada de instrucciones
 - Operan en un lenguaje apropiado para su interpretación en la CPU

11



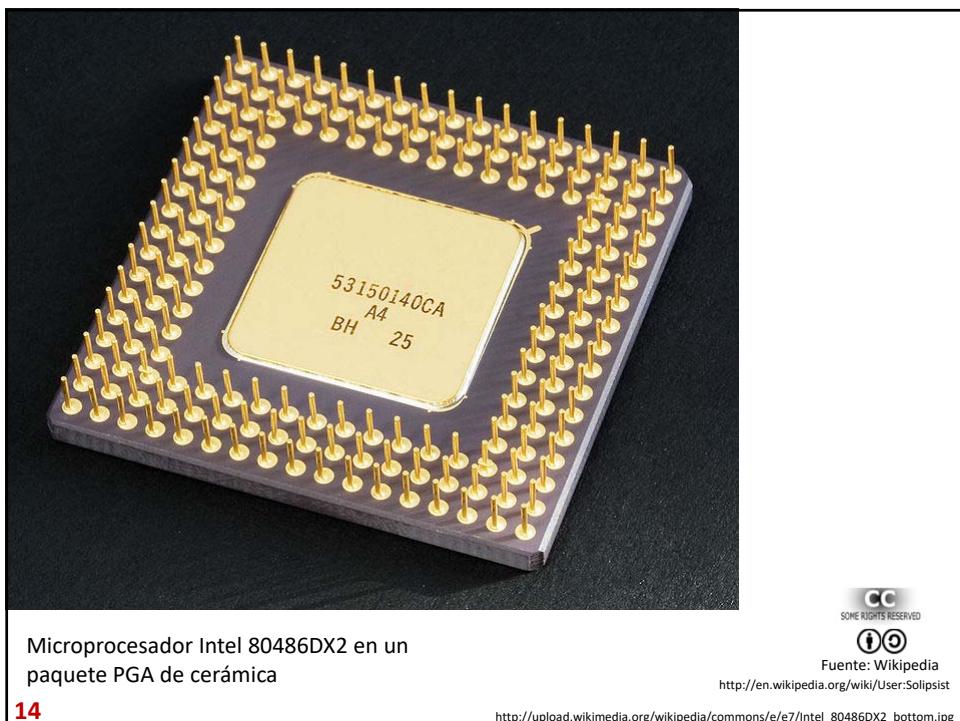
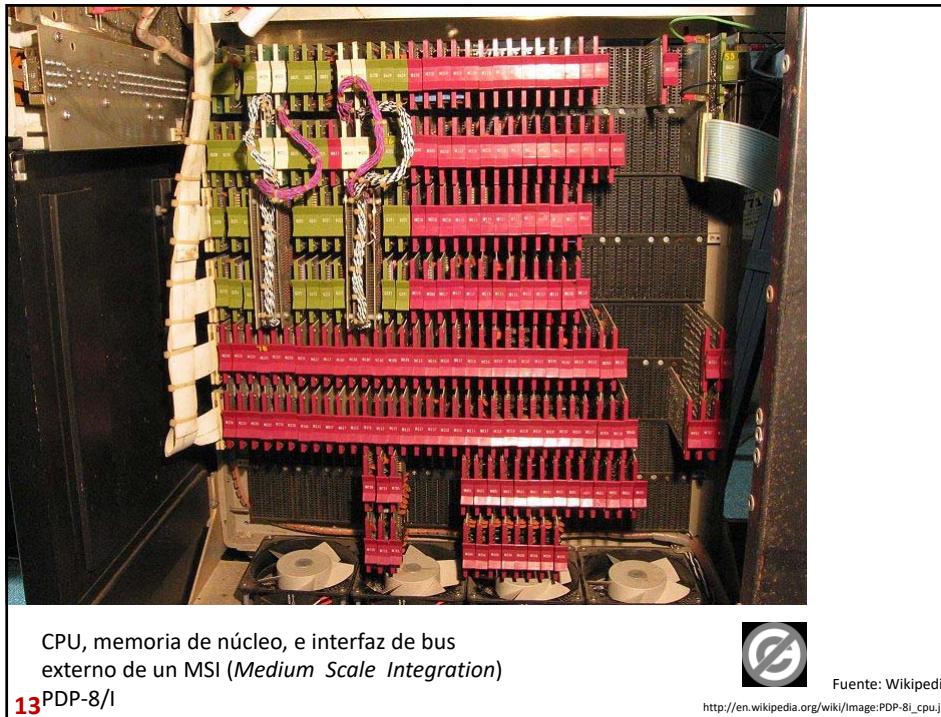
El EDVAC, uno de los primeros computadores de programas almacenados electrónicamente

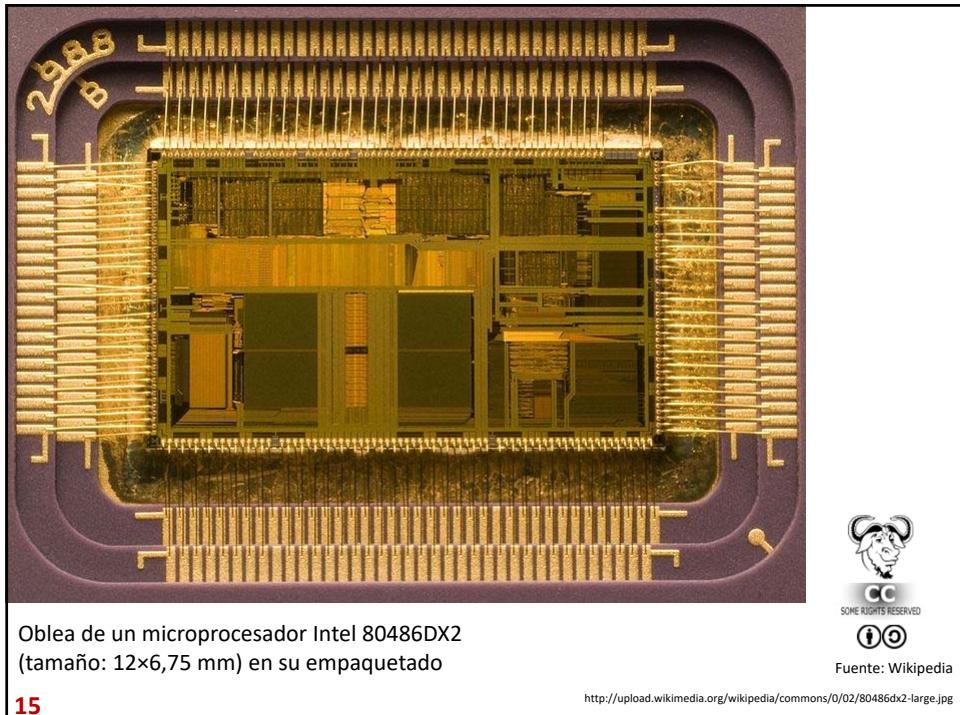


Fuente: Wikipedia

<http://upload.wikimedia.org/wikipedia/commons/1/17/Edvac.jpg>

12





Oblea de un microprocesador Intel 80486DX2
(tamaño: 12×6,75 mm) en su empaquetado



Fuente: Wikipedia

15

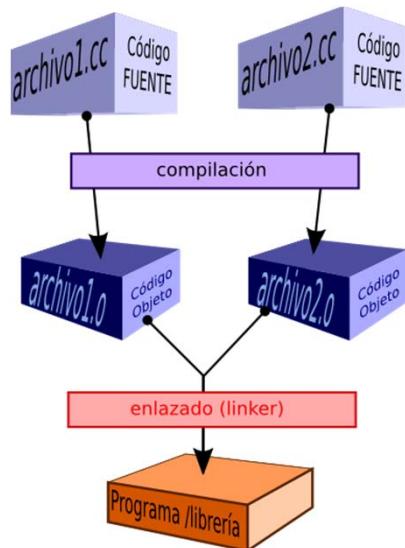
<http://upload.wikimedia.org/wikipedia/commons/0/02/80486dx2-large.jpg>

Los distintos tipos de código

- Código **fuente**
 - Lenguaje de programación amigable para el programador
- Código **objeto**
 - Compilador
 - Construcción del código legible por la CPU
- Código **ejecutable**
 - Enlace con librerías del sistema
 - Construcción del ejecutable empaquetado para ese sistema operativo
- **Compilación versus Interpretación (scripts)**

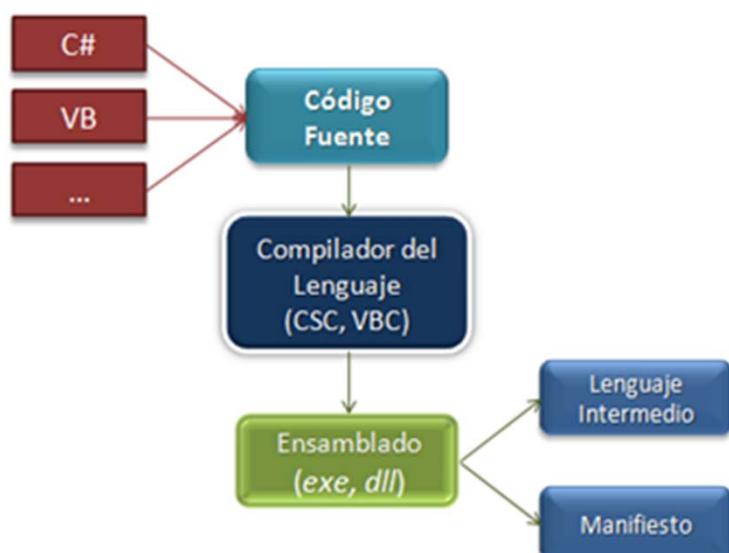
16

Compilación tradicional

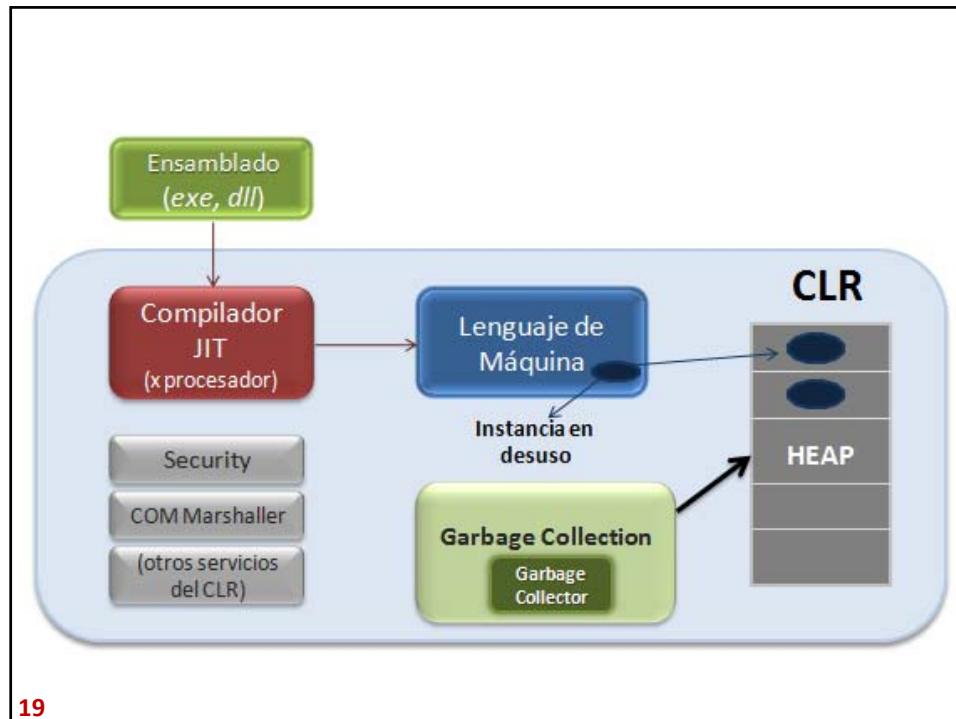


17

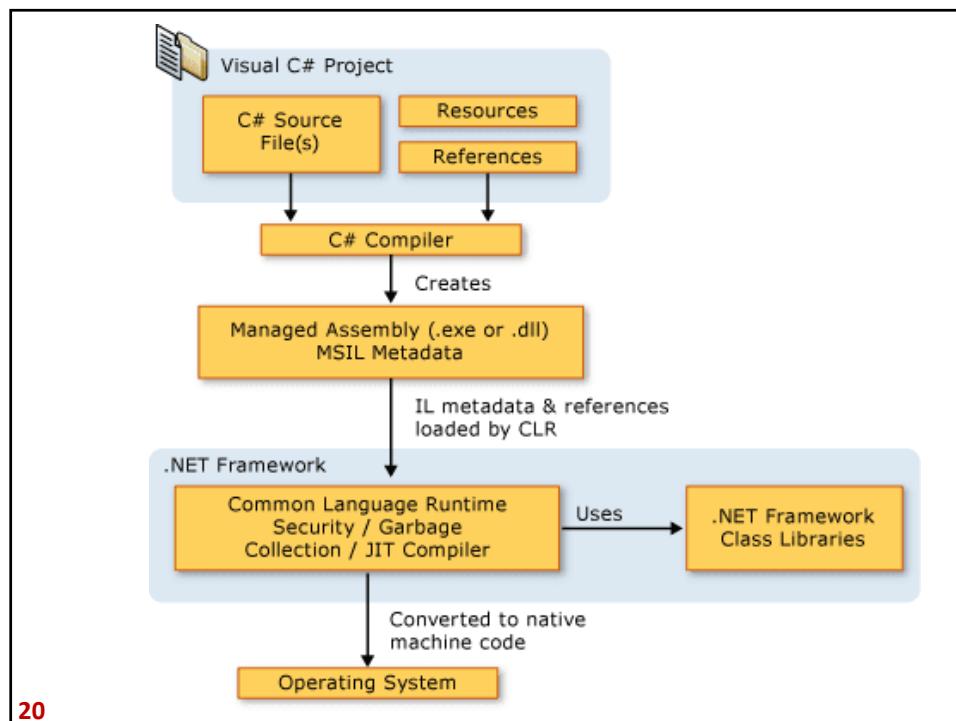
Compilación de código gestionado



18



19



20

Componentes de .NET (I)

- **CLR o Common Language Runtime:** al igual que Java, .NET utiliza un entorno (o "máquina virtual") para interpretar el código en tiempo de ejecución. Todo el código .NET se compila en un lenguaje intermedio al código nativo "Just-In-Time" antes de la ejecución.
- **CIL o Common Intermediate Language:** hablando de un lenguaje intermedio, .NET usa CIL (también conocido como MSIL). Todos los lenguajes .NET (de los cuales hay muchos) están en "assemblies" en este lenguaje intermedio. CIL es un lenguaje assembly genérico orientado a objetos que se puede interpretar en código máquina para cualquier arquitectura de hardware. Como tal, los diseñadores de lenguajes .NET no necesitan diseñar sus compiladores en torno a las arquitecturas en las que se ejecutarán. En cambio, simplemente necesitan diseñarlo para compilarlo en un idioma: CIL.

21

Componentes de .NET (II)

- **Assemblies .NET:** las aplicaciones .NET se empaquetan en .NET Assemblies. Se llaman así porque el código de su lenguaje elegido se ha "ensamblado" en CIL pero no se ha compilado realmente. Los assemblies usan una extensión del formato PE y se representan como un EXE o una DLL que contiene CIL en lugar de código de máquina nativo.
- **Dominios de aplicación:** los assemblies se ejecutan dentro de una "caja" segura conocida como dominio de aplicación. Pueden existir varios assemblies dentro de un dominio de aplicación, y pueden existir múltiples dominios de aplicación dentro de un proceso. Los AppDomains están destinados a proporcionar el mismo nivel de aislamiento entre los ensamblajes en ejecución que normalmente se proporciona para los procesos. Los subprocesos pueden moverse entre AppDomains y pueden compartir objetos a través de la organización y los delegados.

22

Fases en la operación de la CPU

- Arquitectura de **Von Neumann**
 - **Fetch** (leer)
 - **Decode** (decodificar)
 - **Execute** (ejecutar)
 - **Writeback** (escribir)

23

Fetch

- Recupera la instrucción desde la memoria de la posición indicada por el Contador de Programa (PC)
 - El tamaño de la instrucción puede variar
 - Arquitectura **CISC**
 - Arquitectura **RISC**
- Después el PC se autoincrementa para apuntar a la instrucción siguiente
- La sentencia JUMP (saltar) modifica directamente el valor del PC

24

Decode

- Se analiza la instrucción leída para prepararse para su ejecución
 - Operation Code (indica qué operación debe realizarse: sumar, restar, mover, etc.)
 - Cero o más direcciones de operandos
 - Otros parámetros (immediate value)

MIPS32 Add Immediate Instruction

001000	00001	00010	0000000101011110
OP Code	Addr 1	Addr 2	Immediate value

Equivalent mnemonic: **addi \$r1, \$r2, 350**

25

Execute

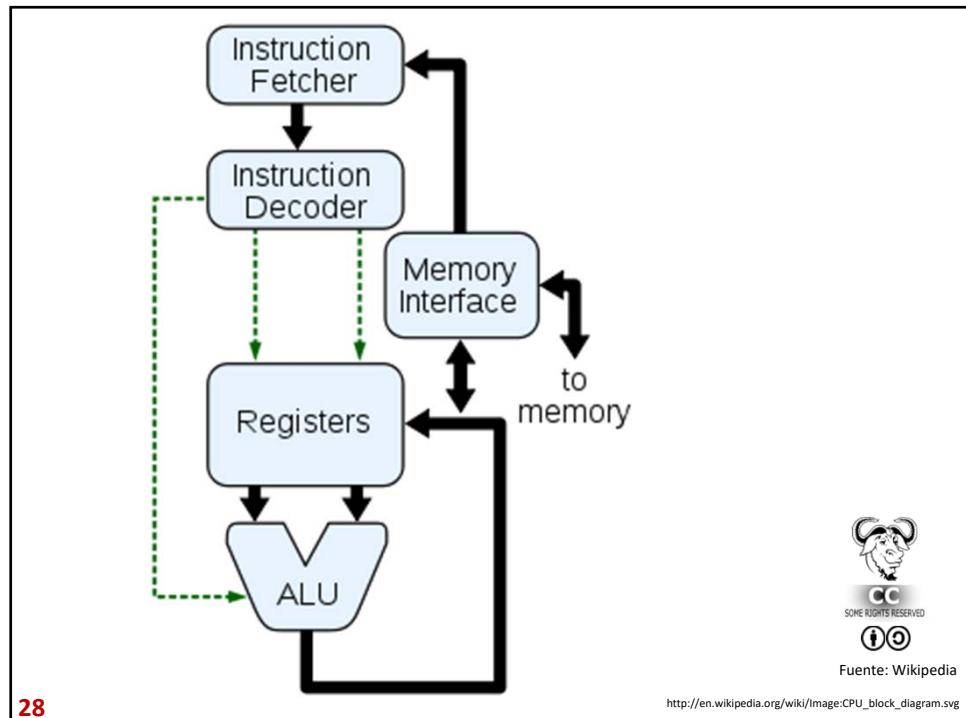
- Activa las señales eléctricas necesarias para activar los circuitos electrónicos encargados de efectuar la operación
- La ejecución puede producir errores
 - Desbordamientos (*overflow* y *underflow*)
 - *Traps* aritméticos
 - División por cero
 - Excepciones

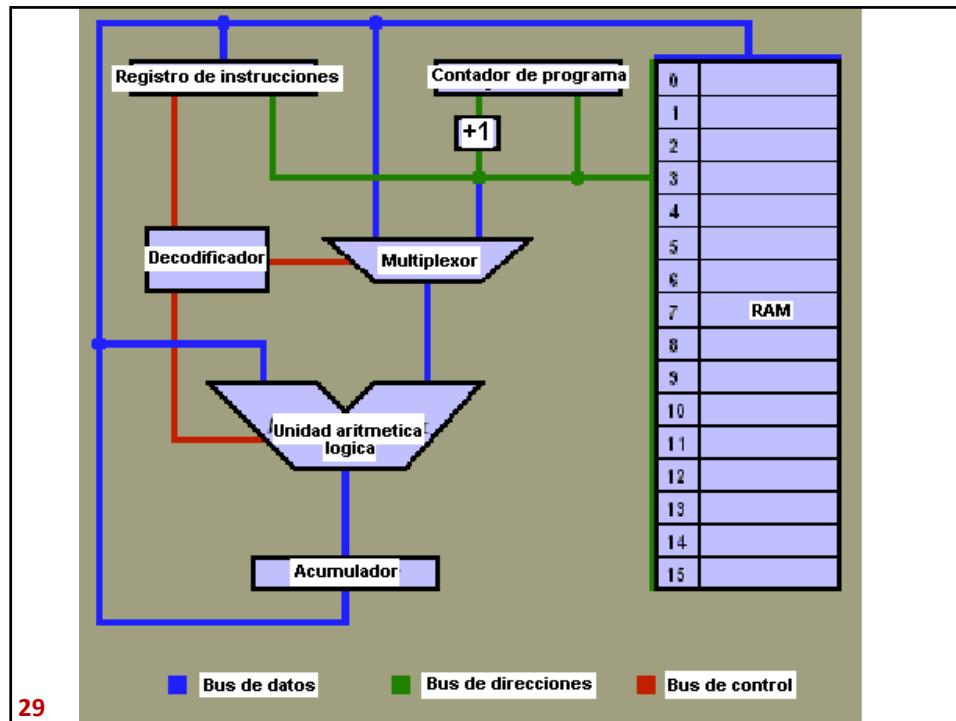
26

Writeback

- Escribe los resultados del paso de ejecución a una cierta forma de memoria
- Frecuentemente este resultado se escribe en algún registro de la CPU para ser utilizado en sucesivos cálculos
- También pueden ser escritos directamente en la memoria central

27



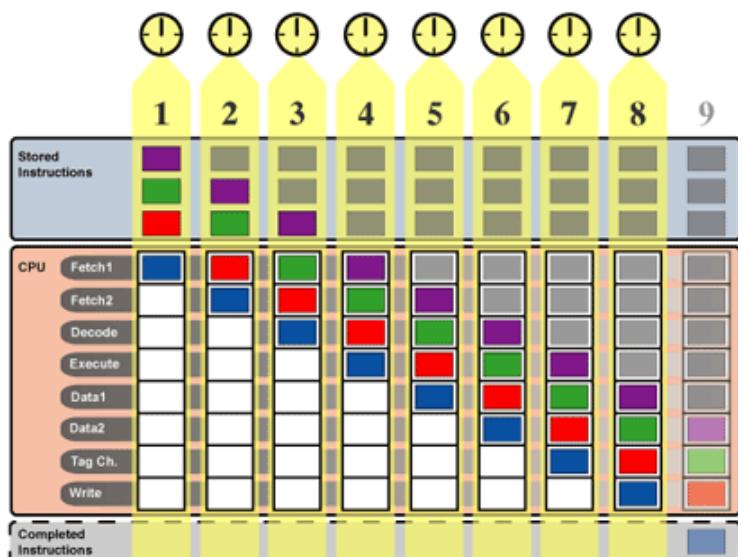


Parámetros que hay que considerar

- Direccionamiento
 - Número de bits de la CPU: 32, 64
- Anchura de buses
- Frecuencia de reloj
 - Medido en Gigahertzios (GHz)
- Paralelismo
 - Pipeline (varias instrucciones a la vez, cada una en su propia fase)
 - Varias unidades aritmético-lógicas
 - Varias CPUs
 - Varios sistemas

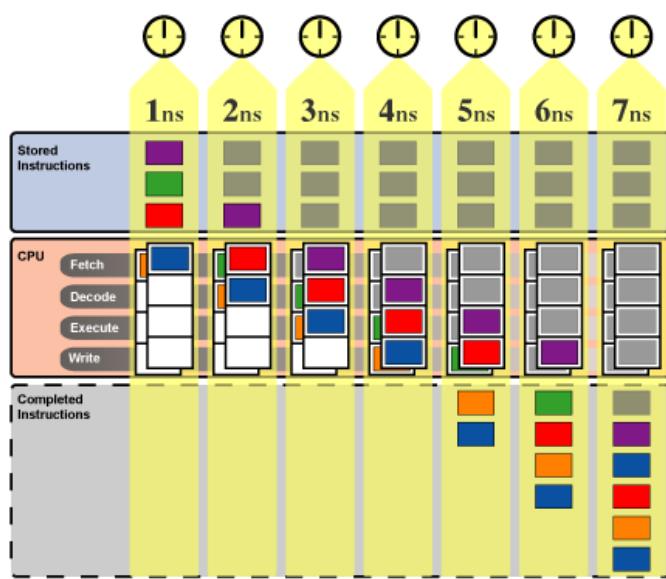
30

Pipeline de 8 etapas



31

Pipeline de 4 etapas superescalar



32



Administrar la ejecución de los procesos. Planificación

GESTIÓN DE PROCESOS EN UN SISTEMA OPERATIVO

33

Proceso o tarea

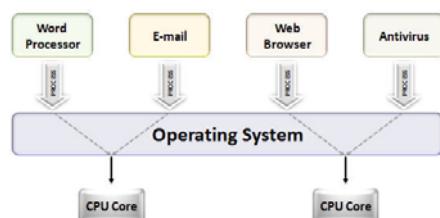
- Es un programa en ejecución
 - Reside en memoria
 - Compite por los recursos de CPU, memoria, I/O, etc.
- Una aplicación puede necesitar de varios procesos para cumplir su función
- A veces, los procesos son distribuidos entre varios sistemas

Story	To Do	In Process	To Verify	Done
As a user, I... 8 points	Code the... 9 Test the... 8	Code the... 4 DC	Test the... SC 6	Code the... Test the... Test the... SC Test the... SC Test the... SC 6
	Code the... 2 Test the... 8	Test the... SC 8		Test the... SC Test the... SC Test the... SC 6
	Test the... 8 Test the... 4			Test the... SC Test the... SC Test the... SC 6

34

Gestión del proceso

- Multitasking (multitarea)
- Scheduling
- Herencia de un proceso
 - Proceso padre
 - Procesos hijos
- Terminación de un proceso
 - Salida programada
 - Salida por excepción
 - Salida por interrupción del operador (“abortar de la tarea”)



35

Planificación de los procesos

- No todas las aplicaciones en todos los sistemas
- No todos los procesos en todas las aplicaciones
- Tener en cuenta el régimen de explotación del sistema
- Adecuar los procesos a las cargas de los sistemas
 - Reparto de recursos



36



Controlar el proceso de organización de la información. Creación, acceso (ubicación física) y borrado de archivos

ORGANIZACIÓN DE LA INFORMACIÓN

37

Algunas clasificaciones sobre tipos de información

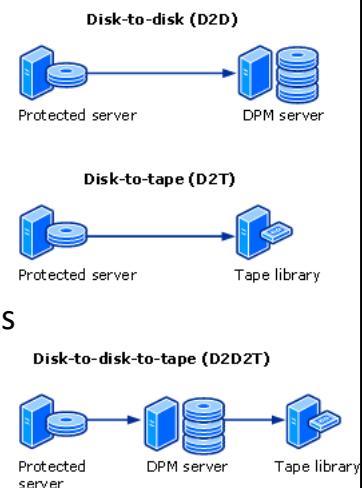
- En cuanto a su operatividad
 - Programas
 - Datos
- En cuanto a su valor o durabilidad
 - Información temporal
 - Información a conservar segura
- En cuanto a su inmediatez
 - Online
 - Offline



38

¿Dónde reside la información?

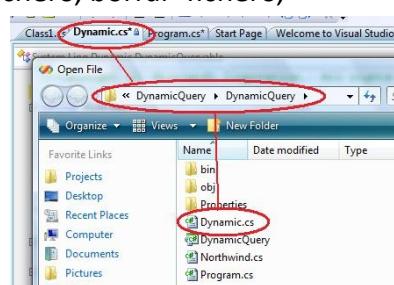
- Establemente
 - En discos y cintas
 - Dispositivos de estado sólido
- De manera volátil
 - En memoria RAM
- En transición
 - En las redes de comunicaciones



39

Organización de la información

- Sobre discos
 - Ficheros o archivos
 - Operaciones básicas: lectura, escritura, borrado, modificación, ejecución
 - Directorios
 - Operaciones básicas: añadir fichero, borrar fichero, listar ficheros, crear y borrar subdirectorios
- Sobre RAM
 - Estructuras de datos



40



Controlar el acceso de los programas o los usuarios a los recursos del sistema.
Proporcionar interfaces de usuario: en modo texto y gráficos.

CONTROL DE ACCESO A LOS RECURSOS

41

Decisiones sobre los accesos

- No todo usuario tiene que poder acceder a cualquier programa
- No todo programa tiene que tener acceso a cualquier recurso
- No todo usuario tiene que poder acceder a cualquier sistema
- Los sistemas tienen que poder controlar el acceso a sus recursos tanto de usuarios como de programas



42

La administración del sistema

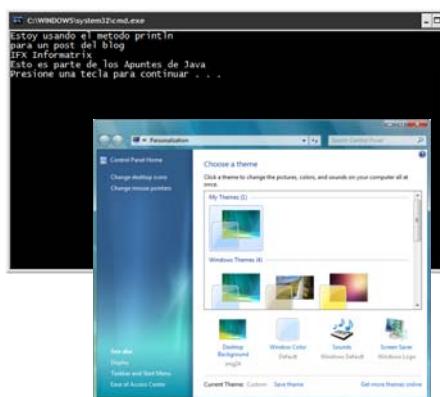
- Gestión de recursos
- Gestión de usuarios
- Gestión de procesos y aplicaciones
- Gestión de cambios y actualizaciones
- Aseguramiento de la información
- Gestión de la seguridad



43

Interfaz de acceso al sistema

- Las consolas de acceso
 - Textuales
 - Gráficas
- Tipos de acceso
 - Según su localización
 - Local
 - Remoto
 - Según su autorización
 - Invitado
 - Autenticado



44

Identificación frente al sistema

- No requiere identificación
 - Accesos invitados
 - Accesos no invitados, pero con autenticación delegada en otros sistemas
- Accesos identificados
 - Mediante usuario y contraseña
 - Mediante certificación digital y/o tokens de seguridad
 - Mediante biometría
 - Combinaciones de los anteriores



45



Servicios soporte: actualizaciones de software, controladores para nuevos periféricos, etc.

SERVICIOS DE SOPORTE

46

Mantenimiento del sistema

- Instalación del sistema
- Actualización del sistema
- Resolución de incidencias
- Averías del hardware
- Averías del software
- Copias de seguridad
- Gestión de la seguridad de acceso
- Hacer rentable la explotación del sistema



47

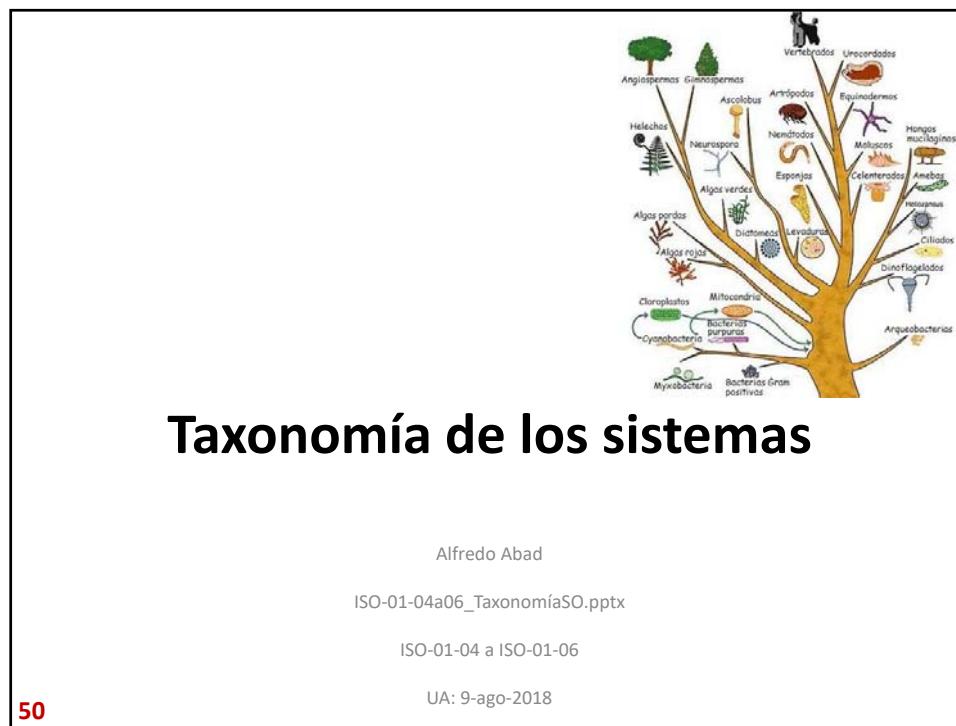
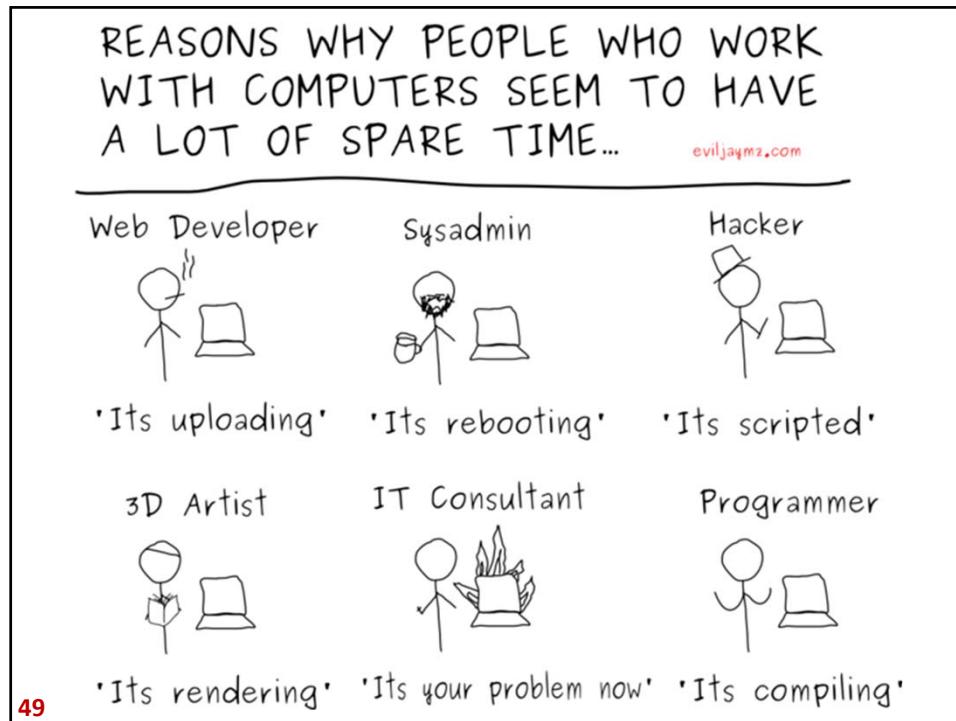
Actualizaciones del sistema

- Gestión de actualizaciones y parches
 - Updates
 - Upgrades
- Instalación de nuevos periféricos
 - Gestión de controladores
 - Recursos periféricos compartidos
- Help Desk
- Documentación



Actualizaciones

48



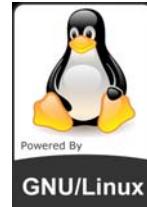


TIPOS DE SISTEMAS OPERATIVOS

51

Atendiendo a los usuarios

- **Monousuario**
 - Sólo pueden atender a un usuario
 - Este usuario opera desde una consola
 - Ejemplo: MS-DOS, Windows Phone, etc.
- **Multiusuario**
 - Atienden a varios usuarios simultáneamente
 - Se gestionan en tiempo real o compartido
 - Requieren de múltiples terminales o deben poder gestionar consolas remotas
 - Ejemplos: Windows, GNU/Linux, MacOSX, etc.



52

Atendiendo a su localización

- **Centralizado**

- Todos los componentes del sistema se localizan en el mismo sitio

- Ejemplo: una estación de trabajo

- **Distribuido**

- Diversos componentes se instalan en localizaciones remotas

- Ejemplo: una gran base de datos en cluster



53

Atendiendo a la gestión de procesos

- **Monotarea**

- Sólo pueden ejecutar una única tarea
 - Sirven para propósitos específicos

- **Multitarea**

- Pueden ejecutar más de una tarea gestionadas por un despachador de tareas
 - Suelen servir a propósitos generales
 - Gestión de hilos o *threads*



54

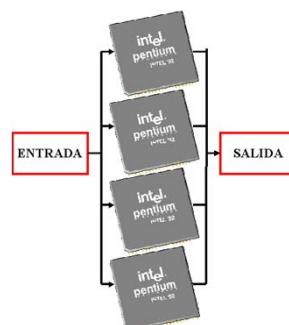
Atendiendo al número de CPUs

- **Monoprocesador**

- El núcleo del sistema sólo soporta un único microprocesador
- Pueden ser monotarea o multitarea

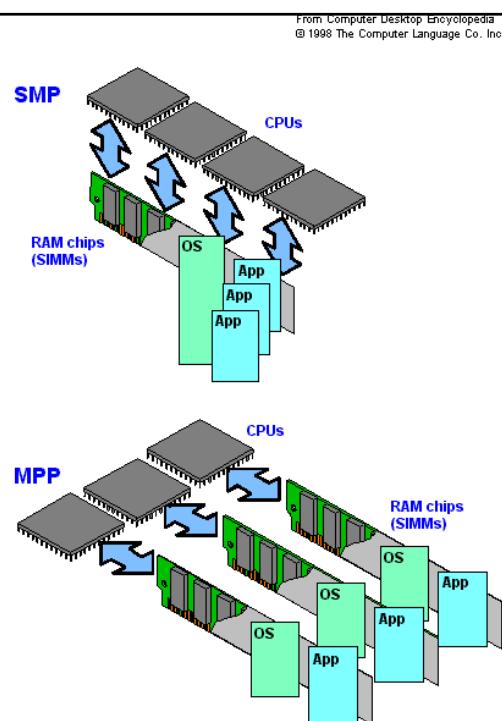
- **Multiprocesador**
(procesamiento paralelo)

- El núcleo soporta varios microprocesadores
- Son siempre multitarea



55

Distintos tipos de proceso paralelo

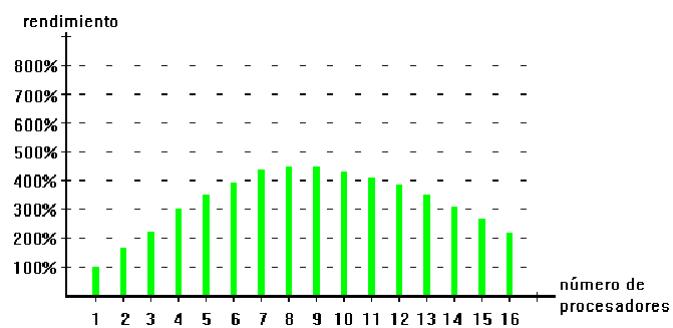
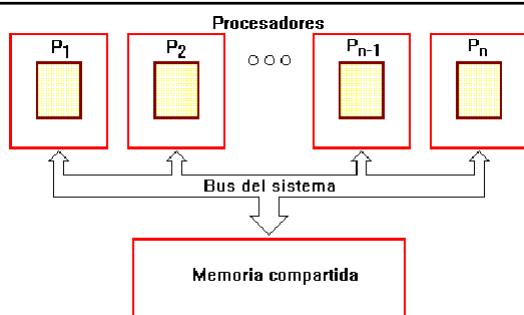


56

Arquitecturas de procesamiento paralelo

- **Multiprocesamiento simétrico (symmetric multiprocessing /SMP)**
 - Múltiples procesadores comparten la memoria RAM y el bus de sistema
 - También se llama «diseño estrechamente acoplado» (tightly coupled) o «compartiendo todo» (shared everything)
 - Cada procesador puede ocuparse de una o más hebras de ejecución
 - A medida que añadimos procesadores el bus compartido se satura (SMP es poco escalable)

57

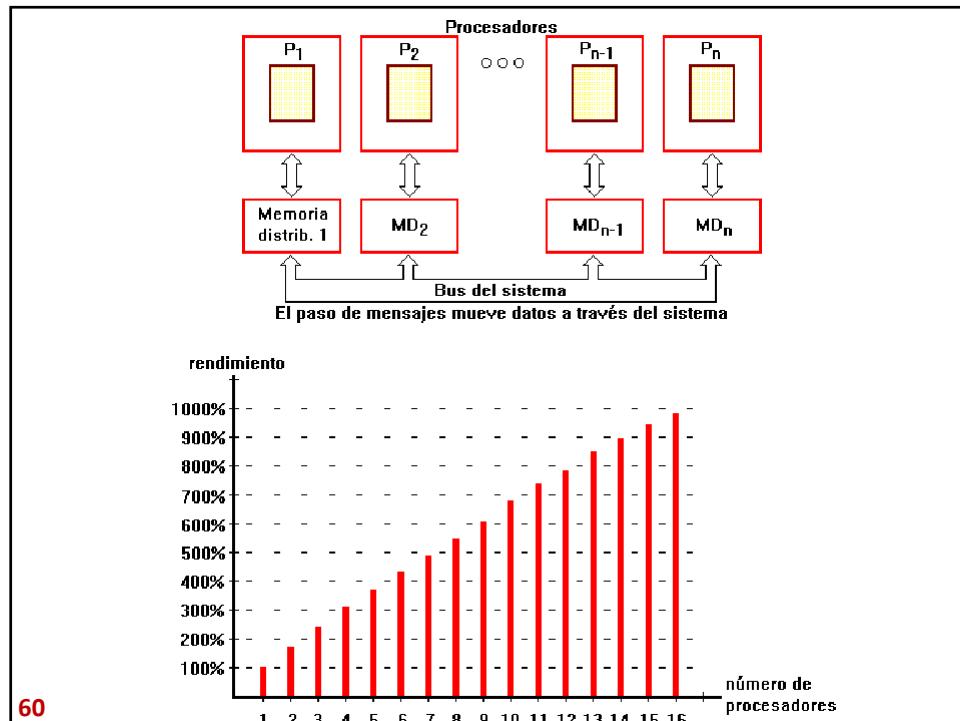


58

Arquitecturas de procesamiento paralelo

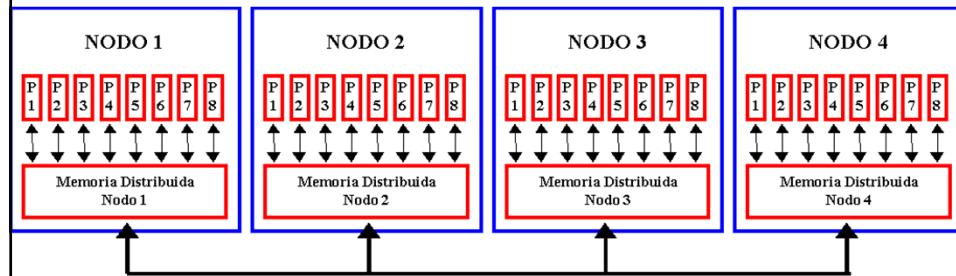
- **Procesamiento masivamente paralelo**
(Massively parallel processing /MPP)
 - No comparte memoria, sino que la reparte entre los procesadores asemejándose a una red de proceso distribuido
 - También se le conoce como «diseño de acoplamiento disperso» (loosely coupled) o «compartiendo nada» (shared nothing)
 - Hacen la programación más complicada

59



60

Sistemas mixtos SMP y MPP

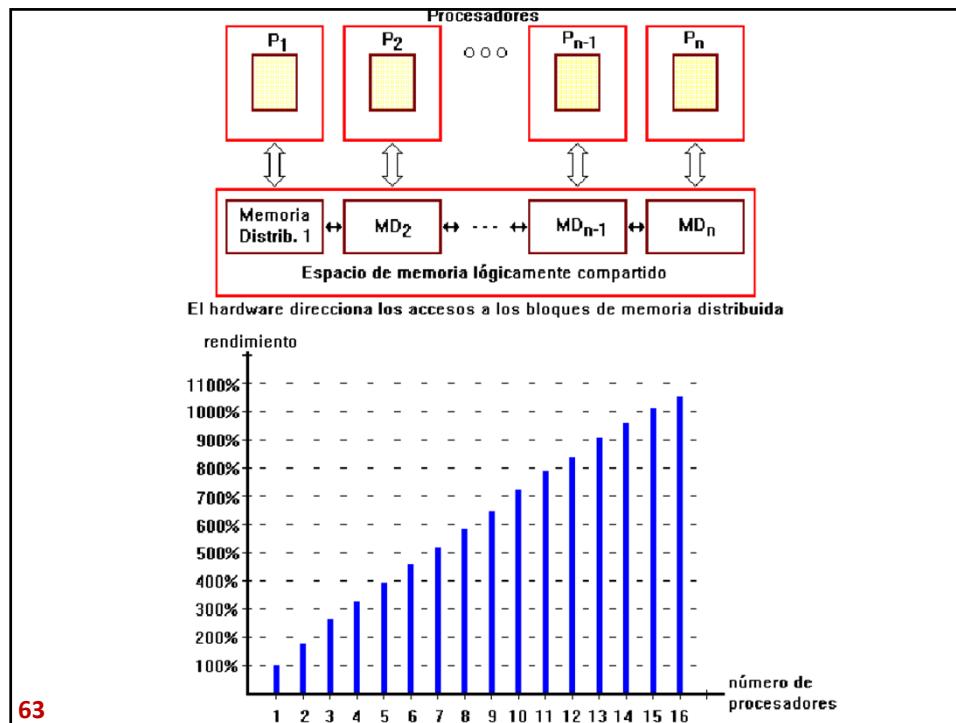


61

Arquitecturas de procesamiento paralelo

- **Procesamiento paralelo escalable (Scalable parallel processing /SPP)**
 - Es un híbrido entre SMP y MPP
 - Utiliza una memoria jerárquica de dos niveles para alcanzar la escalabilidad

62



Atendiendo a su instalación

- Sistemas operativos **instalables**
 - El sistema se instala en un dispositivo no volátil
- Sistemas operativos **autoarrancables**
 - Se arrancan desde CD/DVD o por red, pero no requieren instalación
 - Sistemas Live-CD
- Instalables con gestión de configuración separada
 - Se instalan, pero se separan instalación y configuración

64

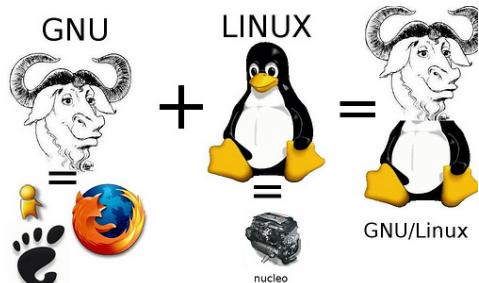


SISTEMAS OPERATIVOS LIBRES

65

Sistemas operativos libres

- Se tiene acceso al código fuente del sistema operativo
- Generalmente son gratuitos, aunque no necesariamente
- Ejemplo: GNU/Linux



66



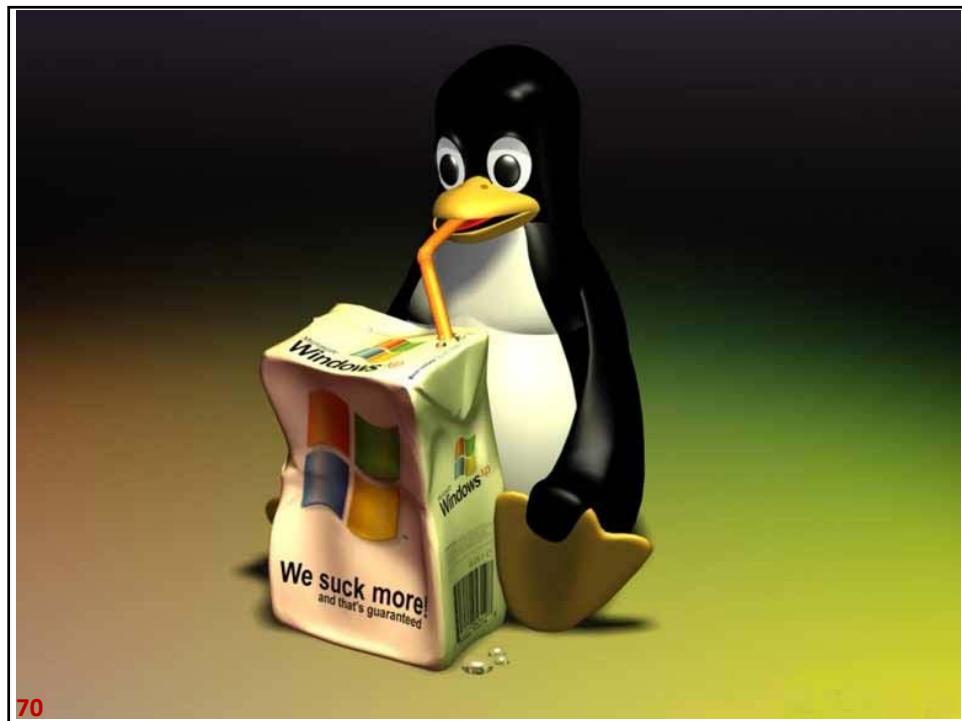
67



68



69



70



SISTEMAS OPERATIVOS PROPIETARIOS

71

Sistemas operativos propietarios

- También llamados propietarios o privativos
- No se tiene acceso al código fuente
- Normalmente hay que pagar por ellos, aunque no necesariamente
- Ejemplo:
 - Cualquier versión de Windows



Microsoft®

72



El escenario de la movilidad

- Apple iOS
 - Versión 12.3 (13-may-2019); actual: v. 13
- Android
 - Android 1.5, Cupcake: April 27, 2009
 - Android 1.6, Donut: September 15, 2009
 - Android 2.0-2.1, Eclair: October 26, 2009 (initial release)
 - Android 2.2-2.2.3, Froyo: May 20, 2010 (initial release)
 - Android 2.3-2.3.7, Gingerbread: December 6, 2010 (initial release)
 - Android 3.0-3.2.6, Honeycomb: February 22, 2011 (initial release)
 - Android 4.0-4.0.4, Ice Cream Sandwich: October 18, 2011 (initial release)
 - Android 4.1-4.3.1, Jelly Bean: July 9, 2012 (initial release)
 - Android 4.4-4.4.4, KitKat: October 31, 2013 (initial release)
 - Android 5.0-5.1.1, Lollipop: November 12, 2014 (initial release)
 - Android 6.0-6.0.1, Marshmallow: October 5, 2015 (initial release)
 - Android 7.0-7.1.2, Nougat: August 22, 2016 (initial release)
 - Android 8.0-8.1, Oreo: August 21, 2017 (initial release)
 - Android 9, Pie: August 6, 2018
 - Android Q: en pruebas (3-abr-2019)
- BlackBerry OS de RIM (Research In Motion)
 - Versión 7.2
- Microsoft Windows 8 y 10
- Otras alternativas en desarrollo:
 - Hongmeng (Huawei), Tizen, WebOS, Firefox OS, Chrome OS, etc.

74

	Google	Alternativas
Navegador		
Buscador		
Nube		
Correo		
Ofimática		
Notas		
Vídeo		
Comunicación		
Fotografía		
Mapas		
Calendario		
Idiomas		
Sistema operativo		
Tienda de aplicaciones		

Estudiar el siguiente documento:
[2019_Cómo sería un Huawei sin Android de Google, y por qué no es tan sencillo escapar de este sistema operativo.pdf](#)

CÓMO SERÍA UN HUAWEI SIN ANDROID DE GOOGLE, Y POR QUÉ NO ES TAN SENCILLO ESCAPAR DE ESTE SISTEMA OPERATIVO

75