IES CONSUELO ARANDA
http://www.isca.es

Sistemas Informáticos

UD2. INTRODUCCIÓN A LOS SISTEMAS OPERATIVOS

Ciclo Formativo Superior de Desarrollo de Aplicaciones Multiplataforma

Profesor: Patricia Calatayud Martínez
Curso: 2016/2017

Contenidos:

2.1. El sistema operativo (S.O.).
2.2. Estructura de un S.O.
2.3. Componentes de un S.O.
2.4. Objetivos y funciones de un sistema operativo.
2.5. Tipos de sistemas operativos
2.6. Tipos de aplicaciones
2.7. Licencias
2.8. Tipos de licencias
2.9. Introducción a la virtualización

Contenidos:

2.1.El sistema operativo (S.O.).

2.2.Estructura de un S.O.
2.3.Componentes de un S.O.
2.4.Objetivos y funciones de un sistema operativo.
2.5.Tipos de sistemas operativos
2.6.Tipos de aplicaciones
2.7.Licencias
2.8.Tipos de licencias
2.9. Introducción a la virtualización

¿

?

S.O.

HW <---->SW

acceso al soporte físico de forma sencilla y segura

abstracción de detalles de implementación

gestión de recursos y procesos

Procesos: programas (conjunto de instrucciones) en ejecución
PCB, PID, PPID, UID (linux), GID (linx)

Kernel: núcleo del SO que interactúa con el HW
Administración de la memoria.
Administración del tiempo de procesado de archivos y recursos.
Administración del acceso a los periféricos/elementos.

==> Elementos del SO:

· kernel
· administrador de memoria
· admnistrador de ficheros
·sistema de E/S

==>S.O.:

software de base o programa del sistema
para interactuar el usuario con la máquina **compuesto por:**
compiladores, ensambladores...

Las visiones del SO pueden clasificarse como:

·**usuario:** no conoce diractamente los servicios SO
Interactúa a través de **interprete de órdenes** para trabajar con SO

==> **interpretede órdenes:** programa que traduce las órdenes de los usuarios a interpretables por el SO (llamadas al sistema)

·**programador:** conoce los servicios del SO
Interactúa a través de **llamadas al sistem**a para programar a partir de servicios SO

scripts, aplicaciones independientes del SO, aplicaciones binarias que llaman al SO

==>**llamadas al sistema:** funciones que permiten acceder a los servios del SO
y que pueden clasificarse como traps (interrupciones software modo usuario-supervisor) o intercambio de mensajes por ejemplo;

las **llamadas al sistema** suelen programarse con **API = interfaz de programación de aplicacione**s y pueden ser: win32 para windows, posix para sistemas basados en linux, java API para la JVM (java virtual machine)

Características de un S.O.

· **Conveniencia:** conveniente uso de PC.
· **Eficiencia:** recursos del PC se usen más eficientemente.
· **Habilidad para evolucionar:** desarrollo, prueba o introducción efectiva de nuevas funciones del sistema sin interferir con el servicio.
·**Encargado de administrar el hardware:** cada proceso una parte del procesador.
·**Relacionar dispositivos (gestionar a través del kernel):** comunicar a los dispositivos periféricos, cuando el usuario así lo requiera.
·**Organizar datos para acceso rápido y seguro.**
·**Manejar las comunicaciones en red.**
·**Procesamiento por bytes de flujo** a través del bus de datos aunque la unidad de medida es el bit.
·**Facilitar las entradas y salidas:** fácil acceso y manejo de los dispositivos de E/S para los usuarios.
·**Técnicas de recuperación de errores.**
·**Evita que otros usuarios interfieran:** evitar bloqueos.
·**Generación de estadísticas.**
·**Compartir el hardware y los datos** entre los usuarios.

Contenidos:

Estructura de un S.O.

Estructura monolítica

Estructura jerárquica o en capas

Estructura máquina virtual

Estructura cliente / servidor

Estructura orientada al objeto

Estructura Multiprocesador

Estructura para funciones

Estructura monolítica:

SO compuesto por **un solo programa** que realiza todas las tareas (lo que se tiene que ejecutar) que:
es más **eficiente en ejecución** porque todos los datos y algoritmos disponibles
pero más **difícil de mantener** (si algo cae, cae todo el sistema), construir y modificar.

Programa integrado por un conjunto de rutinas entrelazadas de tal modo que cada una puede llamar cualquier otra.

La estructura consiste en que no hay estructura.

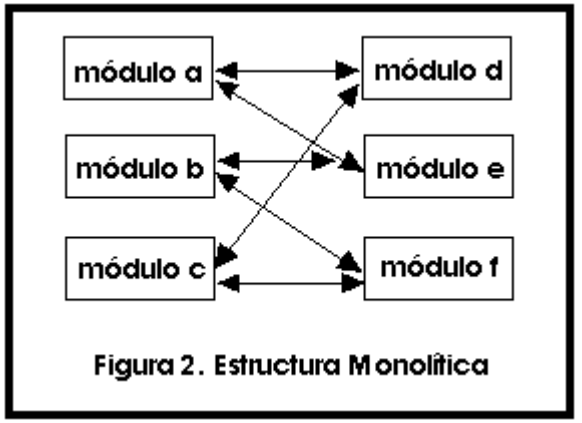


Figura 2. Estructura Monolítica

Las tres principales características de esta estructura son:

1. Es muy común. No existe estructura propiamente o es mínima.
2. El sistema operativo es una colección de procedimientos que se pueden llamar entre sí.
3. Cada procedimiento tiene una interfaz bien definida en términos de parámetros y resultados.

Un ***procedimiento*** es un bloque de acciones que son llamadas desde un mismo programa.

Una ***interfaz*** **es** un conjunto de herramientas que facilitan la comunicación entre los usuarios y el sistema.

Estructura jerárquica/capas:

Mejora de la organización con la división del sistema operativo en pequeñas partes independientes, pero con capacidad de relación con las otras.

Cada **parte está definida** y con una **interfaz** con la resto de los elementos.
La capa n utiliza los servicios de la capa n−1
y ofrece los servicios a la capa n+1.
Las ***funciones*** de un nivel superior pueden invocar a otros de los nivele
no las de los niveles superiores.

La estructura jerárquica también se puede representar en forma de anillas (capas circulares).

La estructura jerárquica los sistemas operativos se estructuran en las capas siguientes:

- **Planificación del procesador:** gestiona el procesador y el modo como pueden acceder al procesador los diferentes programas.

- **Gestión de memoria:** gestiona la memoria y la utilización que se puede hacer por parte de los diferentes programas.

- **Gestión de entrada / salida:** gestiona cada uno de los periféricos de el ordenador.

- **Sistema de archivos:** gestiona la información de los usuarios y fija mecanismos de protección necesarios para lograr un sistema de seguridad aceptable.

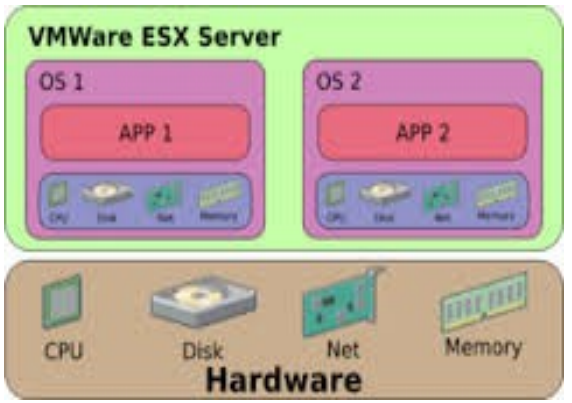
- **Programas de usuario:** aplicaciones especificas de utilización por parte del usuario.

Estructura máquina virtual:

Sistemas que se ejecutan en una **máquina que parece idéntica** a la maquina real, llamada máquina virtual.

La máquina virtual es una **interfaz** que mantiene una máquina mediante la cual nos da comunicaciones con los dispositivos del ordenador. De esta manera, se trabaja en nivel superior y se elimina complejidad.

Así podemos mantener **diferentes sistemas operativos** en funcionamiento sobre una misma máquina.



El **núcleo** de este sistema operativo denomina **monitor virtual.**
Tiene como misión hacer la **multiprogramación** de manera que presenta los niveles superiores tantas máquinas virtuales como se soliciten.

Multiprogramación: varios procesos en ejecución a la vez aprovechando eficientemente los servicios del SO.

Cuando un programa ejecuta una llamada, la llamada se coge y se envía al sistema operativo de su propia máquina virtual para que la gestione.

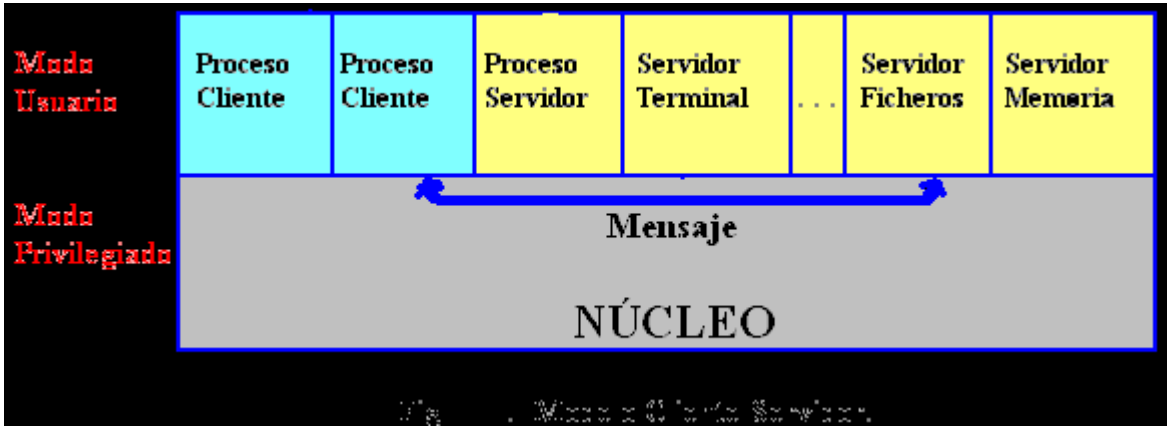
Las 5 **características principales del monitor virtua**l son:

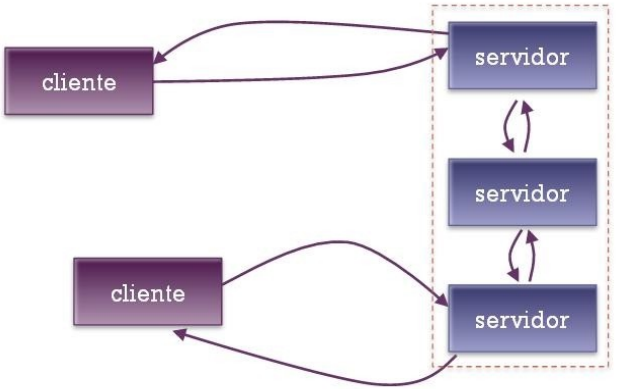
1. Proporciona varias maquinas virtuales en la capa superior.
2. Las máquinas virtuales instrumentan copias ″exactas″ del HW simple.
3. Pueden ejecutar cualquier SO de forma directa sobre el HW.
4. Las diferentes máquinas virtuales pueden ejecutar diferentes Sos.
5. Soportan periféricos virtuales.

Estructura cliente/servidor:

Traslada las funciones del sistema operativo en la **capa de los procesos del usuario** (todo el código posible a las capas superiores) y conseguir un **kernel mínimo** (liberar tanto como posible las capas inferiores) y de suministrar mecanismos para la gestión de accesos a la memoria y la comunicación entre programas.

Los **procesos** pueden ser tanto **servidores como clientes.**





Un programa de aplicación normal es un cliente que llama al servidor correspondiente para acceder a un fichero o para hacer una operación de E / S sobre un dispositivo. A la vez, un **programa cliente** puede actuar como servidor de otro.

El **núcleo** tiene como misión establecer la comunicación entre los clientes y los servidores.

SO en partes ===> cada parte controla una faceta del sistema:
servicio a ficheros, a procesos, a terminales o a memoria;

Además, todos los **servidores** se ejecutan como **procesos en modo usuario, y no en modo núcleo, no tienen acceso directo al hardware.**

En consecuencia, si hay un **error en el servidor** de archivos, éste puede fallar, pero esto **no afectara** el funcionamiento general de toda la máquina.

Estructura orientada al objeto:

El sistema operativo es una **colección de objetos.**

El **núcleo del sistema operativo** será el responsable del mantenimiento de las definiciones del tipo de objetos soportados y del control de los privilegios de acceso al mismo.

Cuando un programa quiere hacer alguna **operación sobre un objeto** determinado, deberá de **ejecutar una llamada al sistema operativo** indicando qué derechos tiene para poderlo utilizar y qué operación interna trata de hacer.

Estructura multiprocesador:

Combinación de procesadores para conseguir sistemas más rápidos que soporten un volumen de trabajo mayor, o por abordar problemas más costosos en tiempo de cálculo.

Esta opción es la más **económica** de conseguirlo.

Podemos clasificar las arquitecturas multiprocesador según la relación que existe entre procesadores y memoria:

- **Multiprocesadores acoplados fuertemente**. Este sistema también se llama sistemas de memoria compartida. En este caso, cada procesador ve y, por tanto, puede acceder directamente a la totalidad de la memoria.

- **Multiprocesadores acoplados débilmente.** También se denominan sistemas de memoria distribuida. Cada procesador tiene acceso sólo a una memoria privada. Los procesadores se comunican entre ellos a través de mecanismos de mensajes.

Estructura por funciones:

Sistema con código organizado para sus funciones. Esta estructura da una organización vertical.

Hay agrupaciones se hacen según el tipo de servicio que se quiere dar, sin tener en cuenta la proximidad o la distancia del hardware, como en el caso de la estructura en capas.

Estas agrupaciones se pueden hacer a partir de servicios de E / S, la gestión de la memoria, etc.

Contenidos:

Componentes de un sistema operativo

Núcleo o kernel.

Administrador de memoria.

Sistema de entrada / salida.

Administrador de archivos.

Sistema de protección.

Interfaz de usuario.

Núcleo o Kernel

Módulo o elemento **más bajo** del SO,
descansa directamente **sobre el hardware** del ordenado
y controla el resto de los módulos y **sincroniza su la ejecucion**.

Está formado por:

- **Un planificador o dispatcher:** asignar el tiempo de procesador a los programas, de acuerdo con una cierta política, normalmente jerarquía de prioridades que muy común en los sistemas operativos multiprogramados y multiacceso.

- **Submódulos para el control de interrupciones** (de programa, por fallo HW, de E/S y de reloj del sistema). Este submódulo esta vinculado al planificador, ya que se utilizan interrupciones para modificar la secuencialización darles respuesta.

- **Comunicador de procesos** (semáforos, mecanismos de waiting / signal): encargado de evitar los bloqueos entre procesos, y ayuda a volver a poner en marcha los procesos, tarea muy importante en el control de concurrencia en sistemas operativos multiprogramados y de procesos distribuidos.

El núcleo del sistema operativo generalmente realiza las **funciones** siguientes:

- Manipulación de interrupciones.
- Creación y destrucción de procesos.
- Cambio de estados de procesos.
- Despacho (dispatcher).
- Suspensión y reanudación de procesos.
- Sincronización de procesos.
- Comunicación entre procesos.
- Manipulación de bloques de control de proceso.
- Soporte de actividades de E / S.
- Soporte de la asignación y desasignación de almacenamiento.
- Soporte del sistema de archivos.
- Soporte de mecanismos de llamada / retorno al procedimiento.
- Soporte de ciertas funciones estadísticas del sistema.

Se trata de dar servicio a **varios usuarios y múltiples tareas** en forma concurrente

Administrador de memoria

Módulo elemento que se encarga de asignar **ciertas porciones de la memoria** principal (RAM) en los diferentes **programas o partes de los programas** que la necesitan para **trabajar.**

Se dijo que el **administrador de memoria** es lo que:

- Ubica, reemplazar, carga y descarga los procesos en la memoria principal.
- Protege la memoria de los accesos no deseados (accidentales o intencionados).
- Permite compartir zonas de memoria (indispensables para la cooperación de procesos).

Memoria virtual: memoria del ordenador aparece más grande de lo que es; gestión de memoria en la que no es necesario tener en un mismo momento todo el programa en la memoria principal para poderse ejecutar.

Sistema de entrada / salida (E / S)

Este componente presenta al usuario los **datos** como una cuestión **independiente del dispositivo**; y el sistema operativo el responsable de atender las particularidades de cada uno.

Funciones:

1. Garantizar el acceso a los dispositivos.
2. Ofrecer un servicio a los procesos, sin conocer el dispositivo de E / S.
3. Tratar las interrupciones, señales recibidas por el procesador, indicando que debe interrumpir el curso de la ejecución actual y pasar a ejecutar un código específico para tratar esta situación, generada por los dispositivos.
4. Planificar los accesos de los dispositivos.
5. Mantener la eficiencia del sistema evitando cuellos de botella.

Las **técnicas más utilizadas** por los SOs para gestionar las E/S son:

- Gestión de colas o spooling: cola en un dispositivo de almacenamiento
- Buffering: espacios de memoria principal para el almacenamiento intermedio

Administrador de archivos

Esta parte del sistema operativo se encarga de **mantener la estructura de los datos y los programas del sistema** correspondientes a los diferentes usuarios y asegura el **uso efectivo** de los medios de almacenamiento masivo.

El administrador de archivos también **supervisa** la creación, actualización y eliminación los archivos, **manteniendo un directorio** con todos los archivos que hay en el sistema en cada momento, **y coopera con el módulo de administración de memoria** durante las transferencias de datos.

Los archivos almacenados en los dispositivos de almacenamiento masivo tienen **diferentes propósitos**.

Por tanto, cada archivo está dotado de un conjunto de **privilegios de acceso**, que indican la extensión con la que se puede compartir la información contenida en el archivo. El sistema operativo comprueba que estos privilegios no sean violados (administración de seguridad).

Hay **diferentes sistemas de archivo**, es decir, diferentes maneras de organizar la información como, por ejemplo: FAT, FAT32, EXT3, NTFS, XFS, etc.

Sistema de seguridad

Módulo o elemento que gestiona los **mecanismos que controlan el acceso a los programas o los usuarios** para poder llegar a los recursos del sistema.

Así pues, este sistema se encarga de las siguientes **funciones:**

- Distinguir entre el uso autorizado y no autorizado
- Especificar los controles de seguridad a hacer
- forzar el uso de mecanismos de protección

Hay desarrollados **diferentes modelos genéricos de protección** de recursos para los sistemas operativos, para controlar el acceso de los usuarios a los recursos

Interfaces de usuario de los sistemas operativos

Una interfaz de usuario es el conjunto de elementos con que los **usuarios** se **comunican o interaccionan** con los **ordenadores u otros maquinas.**

Los sistemas operativos ofrecen dos tipos diferentes de interfaces de usuario:

- **Interfaces de usuario alfanuméricas o de línea de comandos.**
- **Interfaces gráficas de usuario.**

Los sistemas operativos actuales, como las interfaces de las aplicaciones, ofrecen interfaces gráficas de usuario muy evolucionadas, basadas en criterios de usabilidad y que potencian los modelos mentales.

Contenidos:

Objetivos:

- Incrementar la productividad de los usuarios (facilitando el uso).
- Proporcionar un entorno cómodo y una abstracción del hardware al usuario.
- Optimizar la utilización de los componentes o recursos del hardware.
- Gestionar los recursos del hardware y del software
- Decidir quién, cuándo, cómo y durante cuánto tiempo se utiliza un recurso.
- Resolver conflictos entre peticiones de recursos, preservando la integridad del sistema.
- Maximizar el rendimiento del sistema informático.
- Permitir la concurrencia de procesos.
- Posibilitar la ejecución de cualquier proceso en el momento que se solicite, siempre que haya suficientes recursos libres para él.
- Ser eficiente en cuanto a reducir el tiempo que ocupa cada trabajo, el tiempo que no se utiliza la CPU, el tiempo de respuesta en sistemas multiacceso y el plazo entre dos asignaciones de CPU en un mismo programa.
- Ser eficiente en cuanto a aumentar la utilización de recursos en general, tales como la memoria, los procesadores, los dispositivos de E / S, etc.
- Ser fiable, es decir, un sistema operativo no debe tener errores y debe prever todas las posibles situaciones.
- Posibilitar y facilitar en lo posible el diálogo entre el hardware y usuario.
- Permitir compartir entre varios usuarios los recursos de hardware que tiene un PC.
- Permitir a los usuarios compartir datos entre ellos, en caso necesario.
- Facilitar las operaciones de E / S de los diferentes dispositivos conectados a un PC.

Funciones:

1. Facilitar la gestión del HW con la constitución de una máquina virtual o extendida. poner al servicio del usuario una **máquina virtual** que tiene unas características que son diferentes (y **más fáciles de abordar**) que las de la **máquina real subyacente** lo que se refleja en distintos aspectos:

- **Entrada / salida (E / S).**
  H**ardware básico** puede que sea **extremadamente complejo y programas sofisticados**
  ==> SO **evita** al usuario de tener que **comprender el funcionamiento.**

- **Memoria:** maquina virtual con memoria que difiere en tamaño a la de la máquina real.
  ==>SO usa memoria secundaria (discos magnéticos, etc.) dando la imagen de una memoria principal **mucho más extensa** de la que se dispone en la realidad.
  ==>SO repartir la memoria principal entre varios usuarios, de forma que cada uno de tiene la imagen de una máquina virtual en que la memoria es **más pequeña.**

- **Sistema de archivos.**
  ==>El SO permite al usuario acceder a la información almacenada a través de **nombres simbólicos** en lugar de hacerlo a través de su posición física en el medio de almacenamiento.

· **Protección y tratamiento de errores.**
==> SO constituye máquinas virtuales que evitan que aunque los Pcs sean **compatidos** por distintos **usuarios** no sean afectados por errores o malas intenciones de los otros.

· **Interacción a nivel de programa.**
==> SO posibilita la **interacción** entre los diferentes programas de los usuarios de forma que, por ejemplo, la **salida** de uno de ellos se utilice como **entrada** de otro.

2. Facilitan la utilización compartida de recursos.

==> SO logra que se **compartan los recursos** de un ordenador entre **usuarios** que trabajan de **forma simultánea**.
Se **incrementa la disponibilidad del ordenador respecto a los usuarios** y, al mismo tiempo, maximizar la utilización de los recursos como procesadores, memorias y dispositivos E/S.

Contenidos:

Clasificación habitual = funcionalidad:

- La utilización de recursos, es decir, procesos concurrentes.
- La interactividad con el usuario.
- El número de usuarios que pueden acceder.
- El tipo de aplicaciones.
- El número de procesadores de que dispone el ordenador.

Sin embargo podemos encontrar **otro tipo de clasificación** con criterios diferenciados:
- El tipo de tecnología.
- La propiedad y licencia de uso.
- El estado de desarrollo (históricos o activos).

Hay que tener presente que esta ultima clasificación puede ser transversal.

No hay sistemas operativos mejores ni peores, sino que es el **usuario** quien tiene que determinar cuál es el **sistema operativo más adecuado** a sus necesidades.

Según la **utilización de recursos:** número de programas a ejecutar simultáneamente.

- **Sistemas monoprogramados:** solo admiten un programa en el sistema y no permiten hacer uso de las técnicas de multiprogramación.
  El programa es cargado en memoria y está ahí hasta que acaba de ser ejecutado. Durante este periodo de tiempo no se puede ejecutar ningún otro programa.
- **Sistemas multiprogramados o multitarea:** utilizan técnicas de multiprogramación y pueden admitir uno o más programas de uno o varios usuarios simultáneamente.

Según la **interactividad:**el tipo de trabajo a que son destinados los sistemas.

- **Sistemas de procesamiento por lotes (batch):** cada trabajo efectúa una serie de pasos secuenciales relacionados y todos los paquetes (bloques de información) de un mismo trabajo se juntan para formar un único lote que pueden utilizar la multiprogramación para ejecutar varios programas a la vez.
- **Sistemas de tiempo compartido (times sharing):** aceptan que varios programas compitan por los recursos del sistema por lo que la CPU es asignada durante un periodo de tiempo limitado a cada uno, llamado quantum.
- **Sistemas de tiempo real (real time):** multiprogramados y interactivos mas exigentes, basados en una respuesta rápida con control del sistema a partir de las informaciones recibidas.
  Los sistemas que no necesitan una respuesta rápida denominan de **tiempo diferido.**

Según el **número de usuarios:**

- **Sistemas monousuario:** solamente permiten en un determinado momento la conexión de un único usuario a la vez en el sistema.
- **Sistemas multiusuario:** ofrecen la posibilidad de que varios usuarios accedan a la vez al sistema y utilizan técnicas de multiprogramación.

Según el **tipo de aplicación:** tipo de aplicaciones informáticas que ejecutara el sistema.

- **Sistemas de propósito general: s**e caracterizan por la capacidad de poder ejecutar cualquier tipo de aplicación informática.
- **Sistemas de propósito especial:** han sido diseñados específicamente para dar servicio a determinadas aplicaciones informáticas.

Según el **número de procesadores:**

- **Sistemas monoprocesadores:** una única CPU y por lo tanto todos los programas se ejecutarán en la misma CPU.
- **Sistemas multiprocesadores o multiproceso:** varias CPU que permiten que un mismo trabajo o diferentes trabajos ejecuten en diferentes CPU en función de los objetivos de rendimiento que tengan fijados.

Según la **distribución de las tareas del sistema,** según como se reparte el trabajo entre varios procesadores conectados en red.

- **Sistemas centralizados:** una misma máquina realiza todas las tareas del S.O.
- **Sistemas distribuidos:** un sistema operativo distribuido es un sistema operativo que engloba y gestiona un entorno distribuido de manera transparente para el usuario. Podríamos definir el entorno distribuido como varios sistemas interconectados con una red que son capaces de cooperar y comunicarse gracias a esta red y el software que la gestiona.
  Externamente los sistemas operativos distribuidos pueden ofrecer los mismos servicios que un sistema de propósito general.

Tendencias actuales y futuras de los sistemas operativos

- Paralelismo:
  - Incremento de multiprocesadores.
  - Extensión de lenguajes paralelos.
- Conmutación distribuida: incremento de las redes de ordenadores conectadas.
- Sistemas tolerantes a fallos.
- Interfaces de usuarios mas amigables:
- Sistemas abiertos: estandarización de sistemas para compatibilizar los diferentes fabricantes a nivel de:
  - Comunicaciones de red.
  - Interfaces de usuario ofertas.

Principales sistemas operativos

- Dentro de los sistemas de tipo **multiusuario**:
  - **DOS.** Sistema operativo monotarea y monousuario muy utilizado desde la aparición del primer PC. Hoy en día, la tendencia se que desaparezca por las limitaciones que tiene. Hay diferentes fabricantes que comercializan el DOS con diferentes nombres. Las marcas más conocidas son las versiones MS-DOS (Microsoft), y PC-DOS (IBM).
  - **Unix.** Es un sistema multiusuario, creado a partir del lenguaje C. Es modular y admite programas de diferentes fabricantes. El núcleo se interactivo, el shell se puede convertirse en un lenguaje de programación, tiene muchas utilidades y herramientas de desarrollo. En el mercado existen varias versiones con cierta compatibilidad entre sí: SCO Unix, Linux, BSD, AIX, Solaris, etc.
  - **Windows-x:** sistema monousuario y multitarea con una arquitectura de 32 bits, que permite ejecutar múltiples aplicaciones simultáneamente y de manera completa. Destaca por tener un entorno gráfico muy potente. Hay diferentes versiones.
  - **Theos:** sistema multiusuario de gestión de pequeñas y medianas empresas
  - **Xenix:** sistema multiusuario variante de Unix.
  - **MVS:** conjunto de máquinas virtuales; el usuario puede ejecutar cualquier S.O.)
  - **OS/2:** sistema monousuario multitarea diseñado para ordenadores personales.

Cada sistema operativo tiene sus **propias limitaciones,** normalmente debido de su filosofía de funcionamiento o de la disponibilidad de memoria y de recursos físicos.

Evolución de los S.O.

Hoy en día: **microordenadores de 64 bits**, con capacidades más potentes de hardware, con futuro prometedor, especialmente la version Linux por su gratuidad y su evolución.

La aparición de las **redes de área local (LAN)** ha resuelto algunas limitaciones: posibilidad de la integración entre sistemas que hace pocos años era impensable (Unix / NetWare, Unix / Windowsx, Unix / DOS, etc.).

Entornos de sistemas **clientes / servidores:**
S.O.s: Unix, OS / 2 y Windows NT, Windows 2000 Server, etc.

La mayoría de los **sistemas operativos actuales**; diseñados para que sean fáciles de usar.
Los **sistemas operativos del futuro** diseñados para un uso y desarrollo fáciles.

Sistemas propietarios y sistemas abiertos
Los sistemas abiertos permiten su utilización de manera libre.
Los sistemas propietarios presentan limitaciones para utilizarlos como estar registrado.

Contenidos:

Software del sistema: conjunto de programas que controlan el funcionamiento del ordenador junto con los recursos y la resto de programas, proporcionando al usuario una interfaz cómoda en la comunicación con el ordenador; está orientado a facilitar y mejorar el rendimiento de los procesos en el ordenador y podemos distinguir:

- **Sistema operativo:** programas que controlan el funcionamiento del ordenador; parte más importante del software del sistema.
- **Controladores de dispositivos (drivers):** programas que permiten al S.O. interactuar con el periférico.

Software de servicios: conjunto de programas o utilidades que permiten la construcción de programas incluyendo herramientas como:

- **Compiladores:** traductor de programas con lenguaje fuente en programas que pueden ser interpretados por los objetos.
- **Editores de textos:** programas para escribir texto y guardarlo en la memoria.
- **Depuradores:** programas que permiten depurar o limpiar los errores de otro.

IDE (entornos integrados de desarrollo): herramientas especialmente diseñadas para el desarrollo de software base con una buena interfaz gráfica para el usuario en que se agrupan todas las herramientas anteriores;
==> programador no necesite introducir órdenes para interpretar, compilar y depurar.

Software aplicación: software que permite al usuario llevar a cabo una o varias tareas específicas; podemos diferenciar entre:

- **Software de aplicación horizontal:**
  - **Procesadores de textos:** programas para creación de documentos de texto. Ejemplos: Lotus Word Pro, Microsoft Word, Corel WordPerfect, OpenOffice.org Writer.
  - **Hojas de cálculo:** programas orientados a la utilización de información en que se requieren cálculos matemáticos. Ejemplos: Quattro Pro, Lotus 1-2-3, OpenOffice.org Calc, Microsoft Excel.
  - **Bases de datos:** programas que permiten manipular grandes cantidades de datos relacionado que son los mismos sistemas gestores de bases de datos que hacen que el usuario interaccione con la misma BD. Ejemplos: MySQL, Microsoft Access, dBase, etc.
  - **Comunicación de datos:** programas que nos permiten navegar por Internet. Ejemplos: Safari, Mozilla Firefox, MSN Explorer, Internet Explorer, Netscape Navigator, Kazaa, MSN Messenger Yahoo! Messenger, Opera, etc.
- **Software de aplicacion vertical:** software hecho a medida para las necesidades del usuario o entidad que compra este tipo de software; por ejemplo:
  - **Aplicaciones para gestionar las multas de tráfico** específicas.
  - **Aplicaciones para gestionar las nóminas** de los trabajadores de una empresa.

Contenidos:

Licencia: contrato entre el programador de un software sometido a la propiedad intelectual sobre el derechos de autor y el usuario, en que se definen con precisión los derechos y deberes de cada parte; el programador, o la persona a quien le haya dado permiso sobre los derechos de explotación, es quien elige el tipo de licencia según se quiera distribuir el software.

Copyright o derechos de autor: forma de protección proporcionada por las leyes vigentes en la mayoría de países, **para los autores de software** tanto para el publicado como para el pendiente de publicar.
Copyleft: antítesis del copyright, mediante la legislación propia de los derechos de autor, para asegurar que **la persona que recibe una copia** u obra derivada del software pueda utilizar, modificar y también distribuir tanto el trabajo como sus versiones.

Diferentes **tipos de software en función del propietario,** la disponibilidad de **utilización y distribución** posterior:

- **Software libre:** se puede modificar para hacer mejoras y redistribuirlo al público, copiado y utilizado para cualquier propósito; por tanto, debe ir acompañado del código fuente para poder hacer efectivas estas libertades que lo caracterizan.
- **Software de dominio público:** no requiere licencia; los derechos de explotación son para toda la humanidad, porque pertenece a todos por igual; cualquiera lo puede utilizar siempre dentro de la legalidad y haciendo referencia al autor original; puede venir de un autor que la ha dado a la humanidad o los derechos de autor del cual han expirado.

- **Software semilibre:** mantiene las características del software libre para los usuarios individuales o entidades educativas sin ningún lucro, pero que prohíbe estas libertades para una utilización comercial.
- **Software gratuito (freeware):** se puede redistribuir libremente pero no se puede modificar porque el código fuente no está disponible.
Así pues, un software gratuito no es un software libre.
- **Software de prueba o shareware:** permite la redistribución, pero no incluye el código fuente y, por tanto, no se puede modifica; además, pasado un periodo de tiempo, normalmente es necesario pagar una licencia para continuarlo utilizando.
- **Software descatalogado o abandonware:** los derechos de autor no son reconocidos o en que la compañía que los creó ya no lo vende; así, pues, aparece el término abandono, que viene de ser abandonado.
- **Software pirateado o warez:** se distribuye violando el copyright del autor; así pues, esta fuera de la ley.
- **Software de propiedad:** para copiarlo, modificarlo, redistribuirlo o utilizarlo se debe solicitar permiso al propietario o pagar. También se denomina **software no libre,** software privado, software privativo, software con propietario y software propietario.
- **Software comercial:** creado por una empresa que quiere sacar beneficios de su utilización.

Contenidos:

Desde el punto de vista del **software de propiedad:**

- **Licencia CLUF (contrato de licencia para el usuario final) o EULA (End user license agreement)**.:la utilización del producto sólo está permitida a un único usuario, el comprador; en formato papel en el mismo producto o en formato electrónico.
  Ejemplos: Word o Excel de Microsoft.

- **Código abierto u open source:** visión práctica ya que la distribución del código fuente hace que el software sea de calidad mejor, más seguro, creativo, evolucione más rápidamente y se oriente a las necesidades del usuario;
  Ejemplo: código de una empresa con generaciones de beneficios.

Hay dos tipos de licencias, **la de código abierto y la de software libre.**
Se suelen mezclar pero más estricto sería considerarlas por separado.
Software libre cumple todos los requisitos para ser software de código abierto;
Ejemplo: licencia de software libre GNU / GPL se podría considerar una licencia de código abierto.

Desde el punto de vista del **software libre** consideramos las licencias:

- **Licencia GPL (general public license of GNU):** licencia que utiliza el copyleft.
El usuario tiene derecho a utilizar el programa con licencia GNU GPL, modificarlo y distribuir las versiones modificadas y obligar a que estas versiones modificadas estén bajo la licencia GNU GPL.

- **Licencia AGPL (licencia pública general de Affero):** ampliación de la GPL diseñada específicamente para asegurar la cooperación con la red si el software se encuentra en servidores de red; es decir, incluye la obligación de distribuir el software si éste se ejecuta para ofrecer servicios por medio de la red.

- **Llicencia LGPL (lesser general public license of GNU):**
licencia que tienen las bibliotecas de software libre.

- **Licencia FDL (free documentation license):**
licencia que tienen el manuales y la documentación en general del software libre.

Contenidos:

Introducción a las máquinas virtuales

La **virtualización** permite **aprovechar los recursos de máquina** en todo tipo de situaciones, incluso a tener **varios sistemas operativos** dentro de la **misma máquina y trabajando a la vez.**

En muchos **momentos** ==> capacidades de las **computadoras no se utilizan al máximo**,
==> margen para el **aprovechamiento de estos recursos**
==> **virtualización:** aprovechamiento sin instalar ni ejecutar más SW

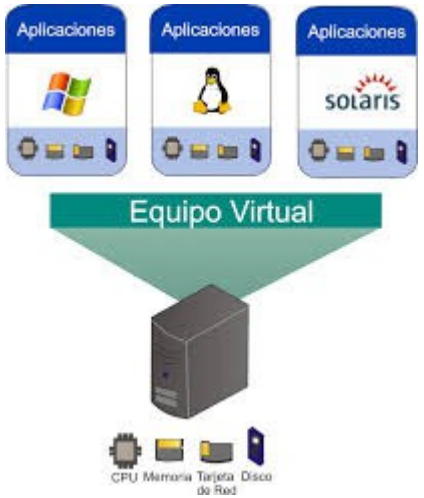
La virtualización consiste en la **abstracción de los recursos** de la máquina para poder **utilizar los que sobran** y crear maquinas virtuales que utilizan este hardware (hardware) como **si estuviera perfectamente libre.**

Compartir una máquina física: CPU, memoria, disco, conexión de red
que de otra manera estarían **libres esperando un pico de trabajo ==>**
==> varias máquinas virtuales con procesos limitados por los recursos y abstracciones.

La virtualización permite **utilizar máquinas virtuales** con **independencia del hardware** lo que se consigue **ocultando los detalles técnicos**.

Esta virtualización se produce mediante esta capa de software conocida como **máquina virtual** que se conoce como **software anfitrión** que simula un entorno de hardware capaz de alojar un **software guest o huésped.**

La **virtualización** ==> es el método para crear una **versión virtual de un dispositivo** o recurso tanto si es todo un servidor como simplemente un disco duro, una red o un **sistema operativo completo** (que se ejecutaría como si estuviera aislado en una plataforma de hardware autónoma).



Tipos de máquinas virtuales

El **apantallamiento del hardware** de un ordenador como si fuera una máquina totalmente autónoma a partir del **encapsulamiento/abstracción** proporcionado por la capa de **software llamado máquina virtual** se puede producir básicamente de **dos maneras.**

Esta clasificación se produce según la **funcionalidad de la máquina virtual** y según su **grado de equivalencia en una verdadera máquina autónoma**:

máquinas virtuales de proceso ==> virtualizar sólo determinados procesos

máquinas virtuales de sistema ==> alojar un sistema operativo huésped

Máquinas virtuales de proceso

La **máquina virtual de proceso o de aplicación**, se ejecuta como un proceso normal dentro de un sistema operativo y **soporta un solo proceso.**

La máquina **se inicia automáticamente** cuando se lanza el proceso que se quiere ejecutar **y se detiene** cuando este proceso ha finalizado.

El **objetivo** es proporcionar un **entorno de ejecución independiente** del **HW y del SO**, por lo que esconde los detalles de la plataforma subyacente y permite que un programa se ejecute siempre de la misma manera sobre cualquier tipo de plataforma.

Ejemplos más conocidos: JVM (Java Virtual Machine), .Net de Microsof
Otros ejemplos: Macromedia Flash Player, SWF, Perl virtual machine, Perl.

La finalidad siempre se **apantallar el hardware** del proceso a ejecutar; de esta manera, se puede programar el proceso sin tener en cuenta en qué tipo de máquina se ejecutase.

Desventajas: añaden una gran complejidad al sistema en tiempo de ejecución y son más lentos que los lenguajes completamente compilados debido a la sobrecarga que genera tener una capa de SW intermedia entre la aplicación y el hardware de la computadora.

JVM

Una de las máquinas virtuales de proceso más utilizadas es la máquina virtual de Java debido a la cantidad de aplicaciones que utilizan este lenguaje de programación.

Diferencia principal Java (que se ejecuta en la JVM) <===> lenguaje común: **eficiencia.**
Para ejecutar un **programa escrito en un lenguaje completamente interpretado,** el intérprete debe hacer el análisis léxica y sintáctica en el momento de ejecutar el programa, lo que provoca una sobrecarga considerable en la ejecución de este programa.
En cambio, los **lenguajes basados en una máquina virtual** suelen ser más rápidos, ya que dividen la tarea de ejecutar un programa en **dos partes**:
en la primera se hace el análisis léxica y sintáctica del programa fuente, para generar el programa en instrucciones del procesador virtual (código intermedio);
en la segunda parte se hace una iteración sobre el código intermedio para obtener la ejecución final del programa.
JVM ==> núcleo del lenguaje de programación ==> procesador virtual (SW intermedio)

Entre las **propiedades/características** del lenguaje Java está:
la **portabilidad** (es posible ejecutar el mismo archivo de clase .class en una amplia variedad de arquitecturas de hardware y de software), el **dinamismo** (ya que las clases son cargadas en el mismo momento de utilizarlas), la **eficiencia** y la **seguridad**.

***Instalación**: Página oficial de Sun ==> escoger hardware y S.O. ==> descargas + ayuda*

Máquinas virtuales de sistema

Las **máquinas virtuales de sistema o de hardware**, permiten **multiplexar** la máquina física subyacente en **varias maquinas virtuales**, cada una ejecutando el **propio S.O**.

La capa de software que permite la virtualización se denomina **monitor de maquina virtual, MMV o tambien hipervisor** y **gestiona los cuatro recursos principales** de un PC:
CPU, Memoria, Red, Almacenamiento ==> **virtualización completa**.
Por ejemplo: PC o portátil con distintos Sos = servidor y clientes ==> simulación de red.

Entre las aplicaciones de las máquinas virtuales de sistema sobresalen:
- **Arquitectura de instrucciones (ISA):** proporcionar una arquitectura de instrucciones ISA ligeramente diferente de la verdadera máquina; es decir, se puede simular hardware.
- **Coexistencia de sistemas operativos:** varios sistemas operativos pueden coexistir sobre la misma máquina, sólidamente aislados unos de otros, por ejemplo, para probar un sistema operativo nuevo sin necesidad de instalarlo directamente.
- **Optimización de recursos:** la virtualización es una opción muy buena para el aprovechamiento de los equipos, ya que en la mayoría de las máquinas actuales están siendo “infrautilizadas”.

Un **MMV** admite varios tipos de arquitectura:

1. Clásica o hipervisor tipo I: el monitor **se ejecuta directamente sobre el HW** y los SO huéspedes corren sobre la capa de SW.

No carga excesivamente el sistema, permite una buena optimización de los recursos de la máquina, pero requiere que los SOs en la máquina virtual sean compatibles con el SW

Ejemplo: la version ESX de VmWare.

2. Indirecta o hipervisor II: el monitor **se instala encima del SO** en vez de hacerlo sobre el HW y los SOs virtuales se ejecutan encima del MMV y los diferentes **sistemas operativos** son instancias de la máquina virtual.

Entre los inconvenientes están la menor eficiencia que tipo I, y no poder instalar un SO que no sea compatible con la arquitectura del procesador de la maquina.

Ejemplo: VitualBoxOSE de Oracle (libre), VMWare, VirtualBox, Microsoft Virtual PC, KVM

3. Paravirtualización: solución intermedia para virtualizar cualquier SO en cualquier HW y hacerlo prácticamente sin penalización en el rendimiento del SO (2% –8%).

Se instala **directamente sobre el HW,** haciendo una **instalación de bajo nivel,** muy cerca del mismo hardware.
El inconveniente es que esta técnica requiere **modificar los SOs** huéspedes para ser instalados en la máquina virtual, **y reduce los posibles SOs huéspedes** aunque ofrecen las mismas prestaciones, y esto pasa desapercibido para el usuario.

Ejemplo: Xen. Xen.

4. Virtualización completa: maquinas virtuales que apantallan el HW y además lo emulan para poder instalar todo tipo de SOs, cualquiera sea la arquitectura.

Se **instala directamente sobre el HW**, los SOs virtuales se ejecutan sobre la máquina virtual, y esta emula el HW cuando no es compatible con el SO.

Como ejemplos: actualmente los fabricantes de procesadores están haciendo modelos de procesadores con la máxima compatibilidad con las máquinas virtuales (AMD–V, Intel VT–x) para facilitar la virtualización total o completa y otros son: Mac–on–Linux, ParallelsWorkstation, algunas versiones de VmWare, como la VmWare GSX Server, VirtualBox y algunas otras.

VmWARE:

Máquinas virtuales de pago, aunque también algunas distribuciones libres.
Dentro de las versiones gratuitas están las VMware Player, VMware Server y VMware ESXi, y la versión comercial VMware ESX Server.

Puede funcionar en Windows, Linux y Mac OS X que utilice procesadores Intel.

La **máquina virtual libre** tanto para Windows como para Linux es: **VMware Player.**
Permite instalar **sistemas operativos huéspedes** tanto del entorno Windows como Linux y Mac OS (para versiones de procesador Intel), tanto de **32 bits como de 64 bits.**
Si se va a instala un **sistema operativo virtual de 64 bits,** primero habrá que asegurarse de que el procesador es de 64 bits, y sobre todo que se de tipo AMD64-VT o Intel64-VT, es decir, un procesador con **soporte específico para la virtualización.**

VirtualBox:

Máquinas virtuales de sistema con arquitectura de hipervisor tipo II.
Pertenece a la empresa Oracle, pero continúa distribuyendo versiones de código libre VirtualBoxOSE.

Virtualización en el S.O.

Virtualización por procesos (JVM) + **máquinas virtuales de sistema** (HW):
==> **3er nivel de virtualización:** virtualizar el mismo S.O., el mismo kernel del S.O.
(no es una capa de software que apantalla el hardware)

Kernel ==> varias **instancias de espacios de usuario**, conocidas como contenedores totalmente aisladas y que permiten instalar un S.O. en cada una:
entornos virtuales

Ejemplo de máquina virtual trabajando sobre el kernel del sistema operativo:
la KVM (kernel-based virtual machine) que permite implementar virtualización completa (full virtualization) con un sistema operativo Linux y un procesador x86
Requisitos: procesador x86 con soporte virtualization thecnology(AMD-V, Intel VT-x)
Servicios: S.O.s como, por ejemplo, GNU / Linux (32 bits y 64 bits) y Windows (32 bits).

Contenidos:

2.1.El sistema operativo (S.O.).
2.2.Estructura de un S.O.
2.3.Componentes de un S.O.
2.4.Objetivos y funciones de un sistema operativo.
2.5.Tipos de sistemas operativos
2.6.Tipos de aplicaciones
2.7.Licencias
2.8.Tipos de licencias
2.9. Introducción a la virtualización