Facultad de Ciencias Sociales y de la Comunicación

Diplomatura en Gestión y Administración Pública

Asignatura de:

Redes de datos

Tema II:

Introducción a los Sistemas Operativos

(Transparencias de clase)

DEPARTAMENTO DE LENGUAJES Y SISTEMAS INFORMÁTICOS

Curso: 2009/2010

Profesor: Manuel Fernández Barcell
www.mfbarcell.es

Indice

1OBJETIVOS Y FUNCIONES DE UN SISTEMA OPERATIVO....3
1.1Qué es un Sistema Operativo....3
1.2El Sistema Operativo cómo interfaz usuario/ordenador....3
1.2.1Capas de sistema informático....3
1.2.2Servicios del Sistema Operativo....4
1.2.3Llamadas al sistema....4
1.2.4La interfaz gráfica o de comandos....5
1.3El Sistema Operativo cómo manejador de recursos....5
1.3.1Gestor de recursos....5
1.3.2Elementos de un sistema Operativo....6
1.3.3Estructura por capas....6
1.4El Sistema Operativo como programa....7
2EVOLUCIÓN DE LOS SISTEMAS OPERATIVOS....7
2.1Los primeros Sistemas (proceso en serie)....7
2.2Sistemas batch con Multiprogramación....8
2.2.1Características de los sistemas multiprogramados....8
2.3Sistemas de Tiempo Compartido....9
2.4Sistemas de tiempo real....10
2.5Sistemas monousuarios....10
2.6Sistemas multiusuarios....10
2.7Multitarea....10
2.8Sistemas abiertos....10
2.8.1Tipos de licencias de software....10
2.9Tipos de sistemas Operativos....11
2.10Generaciones de Sistemas Operativos....11
3SISTEMAS MULTIPROCESADORES....12
3.1Sistemas fuertemente acoplados o sistemas paralelos (Tightly coupled) 12
3.1.1Multiproceso simétrico (SMP)....12
3.1.2Multiproceso asimétrico....13
3.2Sistemas débilmente acoplados (Loosely coupled)....13
3.2.1MPP (Massively Parallel Processing)....13
3.2.2Sistemas distribuidos....14
3.2.3Escalas de los sistemas....17
4SISTEMAS OPERATIVOS ACTUALES....17
4.1Sistemas operativos tipo Windows....17
4.2Sistemas Operativos tipo Unix....19
4.3Linux....20
4.4Freebsd....21
4.5Sistemas Operativos Propietarios....21
4.6Sistema Operativo de Red (NOS Net Operating System)....21
5CUESTIONES....23

En este Tema pretendemos aclarar las funciones del Sistema Operativo, como parte del software de un sistema informático. Entender la importancia que tiene la elección de un determinado Sistema Operativo en nuestro sistema. Comprender los distintos tipos de Sistemas Operativos que existen, sus características y aplicaciones principales. Que conozcamos las características de distintos Sistemas Operativos comerciales además de los de Microsoft.

Objetivos

1.Recordar las principales funciones de un sistema operativo (conocimiento).

2. Nombrar las características y tipos de sistemas operativos (conocimiento).

3. Nombrar dos sistemas operativos que no sean de Microsoft (conocimiento)

1 Objetivos y Funciones de un Sistema Operativo

1.1 Qué es un Sistema Operativo

No hay una definición estándar de Sistema Operativo. Si podemos dar una descripción de lo que debe ser un S.O. Un Sistema Operativo es un **programa** (o colección de programas de control) que:

1. Controla la ejecución de otros programas (los programas de aplicación, de servicio...)
2. Actúa como interfaz entre el usuario de un ordenador y su *hardware*
3. Proporciona al usuario un entorno cómodo y eficiente para ejecutar sus programas.

Sus principales objetivos son:

Comodidad: Facilita el uso del Sistema Informático. Un sistema Operativo hace que un ordenador sea más fácil y cómodo de usar.

Eficiencia: El uso **eficiente** del *hardware* del ordenador. Un Sistema Operativo permite que los recursos del ordenador se utilicen de forma eficiente.

En la actualidad el primer objetivo prima en cierta forma sobre el segundo al disminuir el precio de los recursos (memorias, discos, velocidad de los procesadores...) y ser por tanto más abundantes. Ya no es tan crítico.

Tenemos tres visiones de lo que puede ser un Sistema Operativo: El sistema operativo como interfaz entre el usuario y el ordenados, el sistema operativo como gestor de recursos y el sistema operativo como programa. El orden no es significativo.

1.2 El Sistema Operativo cómo interfaz usuario/ordenador

1.2.1 Capas de sistema informático

Los cimientos de un sistema informático están compuestos por el ***hardware*** del ordenador. La arquitectura de cualquier ordenador moderno es compleja. Procesadores, *chipset*, controladas, dispositivos de entrada salida, memoria etc.

Sería muy difícil para el usuario y su aplicación (programa) dominar todos estos elementos en el ámbito de *hardware* y direcciones máquina. Un Sistema Operativo transforma, por ejemplo, una dirección física de un puerto de una impresora en una "máquina ficticia" (prn, lpt) mucho más simple de usar. En el antiguo MS-DOS para mandar imprimir un directorio escribíamos `C:\>dir > prn`. Actualmente en los programas con interfaz gráfica vemos un dibujo (icono) que representa una impresora.

`UN SISTEMA OPERATIVO TRANSFORMA UN CIERTO` *`HARDWARE`* `EN UNA MAQUINA VIRTUAL MÁS FÁCIL DE USAR`

El Sistema Operativo actúa como mediador entre el *hardware* y el usuario, facilitando al programador, a los programas y a los usuarios finales, el acceso y uso de los medios y servicios del sistema.

Establecemos un sistema de ***capas*** de *software* entre el *hardware* puro y duro y el usuario. Estas capas establecen niveles independientes que se encargan cada uno de ellos de alguna función concreta.

Cada capa aisla los problemas específicos a las otras capas, permitiendo un desarrollo independiente. Cada capa sólo se relaciona con las capas adyacentes.

Las áreas en que la máquina virtual difiere de la real son:

- Entradas/salidas.
  - Un sistema operativo evita al usuario el problema de tener que comprender el *hardware* de e/s, aunque manteniendo las mismas posibilidades de e/s.
- Memoria.
- Manejar una memoria **virtual** distinta de la real.
- Sistema de ficheros.
- Protección y tratamiento de errores.
- Compartición de recursos entre varios usuarios.
- Interacción a nivel de programas.
- Redireccionamientos, intercambio de datos etc.
- Control de programas.
- Lenguajes de control (comandos).

Vemos que la capa Sistema Operativo se sitúa entre el *hardware* y el resto del *software.*

El sistema operativo realiza su función de intermediario ofreciendo servicios a los programadores y a los usuarios finales.

- **A los programadores**, ofreciéndoles *las llamadas al sistema* como medio para obtener sus servicios para facilitarles el desarrollo de programas.
- **A los usuarios finales**, mostrándoles una forma amigable (mediante un entorno gráfico o comandos) de realizar operaciones comunes (ejecutar programas, crear carpetas, copiar ficheros, instalar aplicaciones...) y de utilizar los recursos del ordenador.

1.2.2 Servicios del Sistema Operativo

Un Sistema Operativo proporciona un entorno para la ejecución de programas. Un Sistema Operativo proporciona **servicios** a los programas (programadores) y a los usuarios de dichos programas.

Los servicios que más comúnmente proporcionan los S.O. son:

- **Creación de programas**. Editores, depuradores...
- **Ejecución de programas**. Cargar, ejecutar y terminar programas.
- **Operaciones de e/s.**
- **Gestión de la información.** Gestión de ficheros, almacenamiento.
- **Comunicación entre procesos**. En el mismo ordenador o entre procesos de distintos ordenadores.
- **Detección de errores.**
- Asignación de recursos.
- Contabilidad.
- Protección y seguridad: Interna y externa.

1.2.3 Llamadas al sistema

Las llamadas al sistema proveen una interfaz entre los procesos y los servicios del Sistema Operativo.

Mediante las **llamadas al sistema,** el usuario (programa) solicita los servicios que desea del Sistema Operativo. De no existir las llamadas al sistema, el usuario (programador) tendría que conocer cómo realizar dichos servicios y programarlos en lenguajes de bajo nivel.

Las llamadas al sistema tiene un formato bien definido. El programador incorpora en el código de su programa una llamada (*call procedimiento*) al servicio que solicita. Se les conoce como las interrupciones *software.* A veces se denominan interfaz de programación de aplicaciones (API, *application-programming interfaces*). El procedimiento es:

1. El proceso (programa) solicita un servicio al sistema operativo incorporando en su programa la llamada al sistema que corresponde al servicio deseado.

Nivel	Nombre	Objetos	Ejemplos de Operaciones
13	Shell	Entorno de programación del usuario	Sentencias de un lenguaje de shell
12	Procesos de usuario	Procesos de usuario	Salir, eliminar, suspender, reanudar
11	Directorios	Directorios	Crear, destruir, conectar, desconectar, buscar, listar
10	Dispositivos	Dispositivos externos tales como impresoras, pantallas y teclados	Crear, destruir, abrir, cerrar, leer, escribir
9	Sistema de archivos	Archivos	Crear, destruir, abrir, cerrar, leer, escribir
8	Comunicaciones	Tubos (*pipes)*	Crear, destruir, abrir, cerrar, leer, escribir
7	Memoria Virtual	Segmentos, páginas	Leer, escribir, traer (*fetch*)
6	Almacenamiento secundario local	Bloques de datos, canales de dispositivos	Leer, escribir, asignar, liberar
5	Procesos Primitivos	Procesos primitivos, semáforos, colas de procesos listos	Suspender, reanudar, esperar, señalizar
4	Interrupciones	Programas de tratamiento de interrupciones	Invocar, enmascarar, desenmascarar, reintentar
3	Procedimientos	Procedimientos, pila de llamadas, visualización	Marcar la pila, llamar, retornar
2	Conjunto de instrucciones	Evaluación de la pila, intérprete de microprogra-, mas, vectores de datos y escalares	Cargar, almacenar, sumar, restar, bifurcar
1	Circuitos electrónicos	Registros, puertas, *buses*, etc	Borrar, transferir, activar, complementar

2. El sistema operativo realiza el servicio solicitado.
3. Devuelve el resultado al proceso que lo solicitó.

Pregunta:

¿Porqué una aplicación escrita para un sistema operativo no funciona habitualmente en otro sistema operativo distinto?

> *Como ya hemos comentado, el Sistema Operativo, se sitúa entre los dispositivos físicos (*hardware*) y nuestras aplicaciones. Cuando un proceso necesita realizar alguna función (leer de disco un registro), no lo realiza directamente, sino que se lo solicita al Sistema Operativo del sistema. Para realizar esa petición utiliza las "llamadas del sistema" adecuadas, que son propias de ese Sistema Operativo. Si trasladamos la aplicación a otro Sistema Operativo, este no entenderá las peticiones que le realiza la aplicación.*

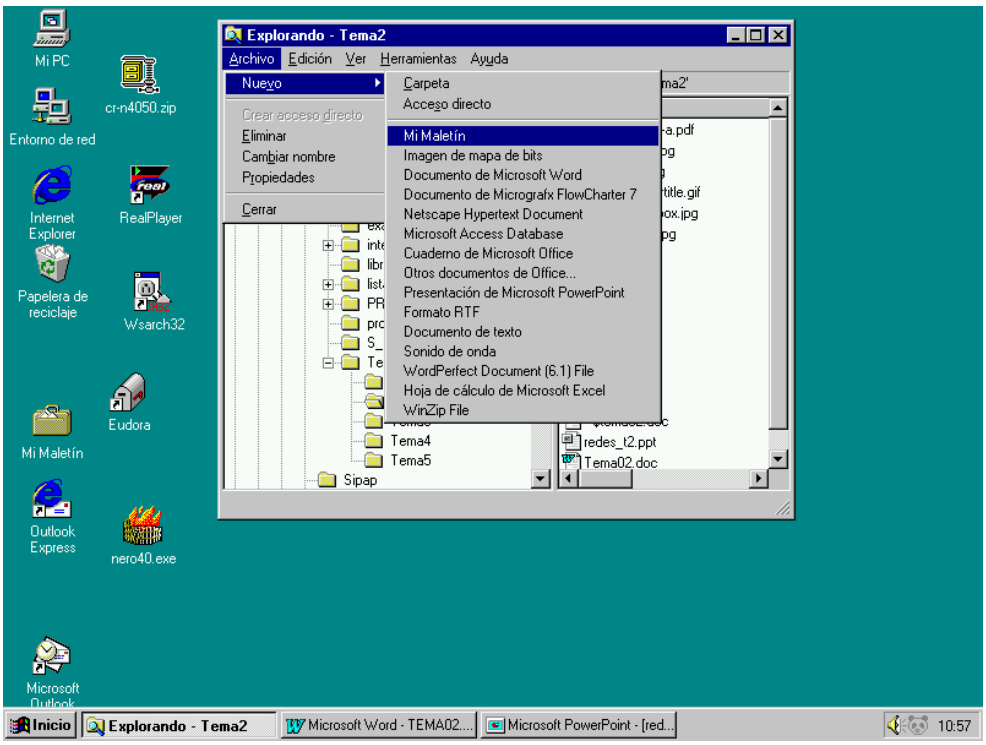
1.2.4 La interfaz gráfica o de comandos

Es la parte del sistema operativo que el usuario final utiliza del sistema operativo. Es la capa más externa del sistema operativo. El usuario final considera normalmente que el sistema operativo es exclusivamente su capa más externa. En el caso de “los Windows” es el escritorio con su `Explorador de Windows` y todos sus menús.

La mayoría de los Sistemas Operativos actuales tienen una interfaz gráfica del tipo "ventanas" y con la utilización de iconos. Unix utiliza la norma `X-windows`.

1.3 El Sistema Operativo cómo manejador de recursos

1.3.1 Gestor de recursos



Un ordenador es un conjunto de recursos para transferir, almacenar, y procesar datos. Para controlar esos recursos existe un programa: el sistema operativo. El sistema operativo es el responsable de administrar esos recursos.

Un Sistema Operativo no es más que un *gestionador de recursos*, que asigna los recursos a los programas y usuarios según sus necesidades y disponibilidades. Un Sistema informático tiene recursos como la memoria central, CPU, discos, impresoras y tiene usuarios que quieren utilizar esos recursos. El Sistema Operativo debe optimizar el uso de sus recursos entre los distintos usuarios para conseguir máximo rendimiento.

Un Sistema Operativo debe de lograr que se compartan los recursos del Sistema y los datos entre varios usuarios que trabajen simultáneamente. Debe de aumentar la disponibilidad del ordenador.

1.3.2 Elementos de un sistema Operativo

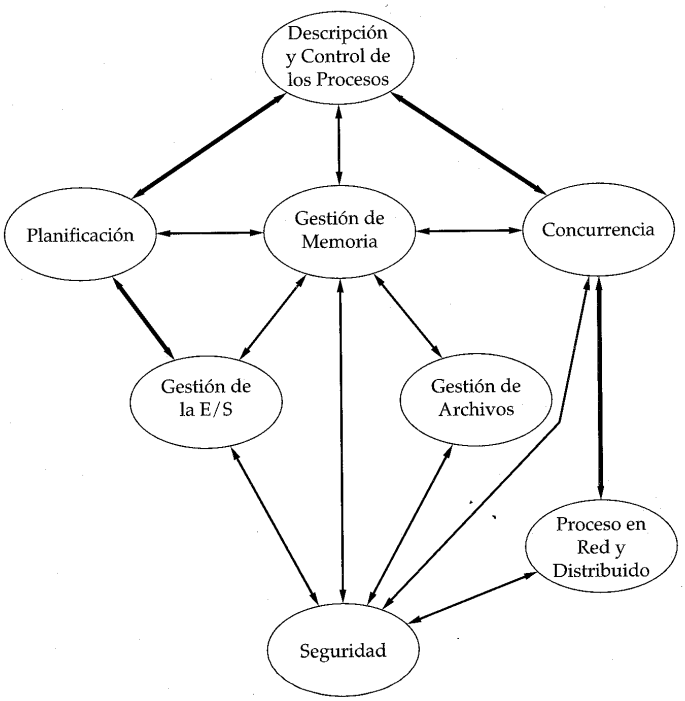
Para administrar los diversos recursos de un sistema informático, hay módulos o parte del sistema operativo que se encargan. Los módulos o componentes típicos de los sistemas operativos son:

- El manejador de procesos
- El gestor de memoria principal
- El gestor del almacenamiento secundario
- El sistema de Entrada Salida
- El sistema de Ficheros
- El sistema de protección
- El sistema de Comunicaciones
- El intérprete de comando (*Shell*)

1.3.3 Estructura por capas

Consiste en organizar el Sistema Operativo como una Jerarquía de estratos o capas, construidos cada uno

arriba del otro.



Cada capa se implementa usando sólo aquellas operaciones que les proporciona el nivel inmediatamente inferior. El Sistema Operativo se divide en pequeñas piezas. Se emplean técnicas modulares y de diseño *Top-down*.

Ventajas:

- Mayor libertad para realizar cambios.
- La depuración y verificación del sistema es mucho más sencillas.

Inconvenientes:

- La mayor dificultad de estos sistemas es definir las capas.
- Menos eficiencia. Las operaciones en vez de realizarse directamente requieren pasar por varias capas.

1.4 El Sistema Operativo como programa

El Sistema Operativo no es nada más que un programa de ordenador. Como otros programas, proporciona instrucciones a la CPU. La única diferencia se encuentra en el objetivo del programa.

El sistema operativo dirige a la CPU en el uso de otros recursos del sistema y en la temporización de la ejecución de otros programas.

El Sistema Operativo se alterna su ejecución con el resto de los programas del sistema.

2 Evolución de los Sistemas Operativos

El diseño de un Sistema Operativo esta fuertemente influido por el tipo de aplicación que se le quiera dar a la maquina. No es lo mismo un Sistema Operativo para gestión que para el de un super ordenador vectorial de cálculo científico.

Para entender lo que es un Sistema Operativo vamos a ver la evolución histórica que han sufrido a lo largo de su historia.

2.1 Los primeros Sistemas (proceso en serie)

En los primeros ordenadores, el programador interactuaba directamente con el *hardware* del Sistema.

Las máquinas se accionaban desde una consola compuesta por luces indicadoras e interruptores.

Estos sistemas planteaban dos problemas fundamentales:

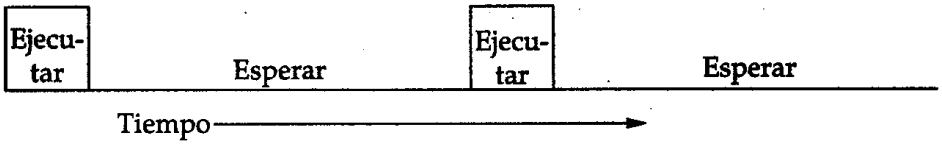
- **Planificación**: se hacía a mano. Con una lista.
- **Tiempo de preparación**: las tareas intermedias, se hacían manualmente con la consiguiente pérdida de tiempo.

Posteriormente, se introduce el **secuenciamiento automático de trabajos**, realizado por un programa residente en memoria llamado **MONITOR RESIDENTE.** (Primer Sistema Operativo rudimentario).

El procesamiento por `lotes` precisa generalmente que el programa, los datos y las órdenes adecuadas, sean remitidos todos juntos al sistema, en forma de "trabajo". De ahí procede el nombre de "`lote`".

Los sistemas operativos por lotes no permiten interacción (o muy poca) entre los usuarios y los programas en ejecución.

Mejora la explotación respecto a los sistemas con procesamiento serie simple.



(a) Monoprogramación

2.2 Sistemas *batch* con Multiprogramación

Justificación

Los procedimientos visto hasta ahora, intentan optimizar el Sistema simultaneando las operaciones de e/s con la utilización de la **CPU** por un único programa.

Ejemplo de tiempos de utilización:

Leer un registro	*0,0015 ut*
Ejecutar 100 instrucciones	*0,0001 ut.*
Escribir un registro	*0,0015 ut*

TOTAL	*0,0031 ut.*

- Porcentaje de utilización de la CPU 0.0001/0,0032 = 3,2 %
- Porcentaje de ociosidad de la CPU 96,8 %

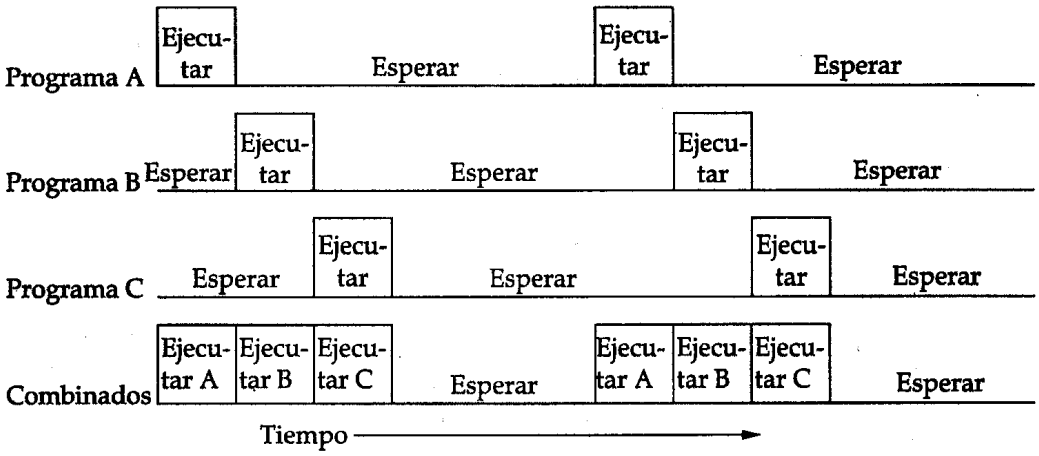
Es muy difícil que un único usuario tenga ocupado completamente a la CPU y a los dispositivos de E/S. Se introducen **sistemas multiprogramados**.

2.2.1 Características de los sistemas multiprogramados

- Hay varios procesos residentes, de modo simultáneo, en la memoria del sistema.
- Los procesos se alternan el uso del procesador/es.
- Los tiempos de E/S de un proceso se solapan con tiempos de proceso de otros.

La idea de la multiprogramación es que la **CPU** atiende a las necesidades de varios programas cargados en memoria simultáneamente.

- Los recursos del sistema informático, son reasignados dinámicamente entre una colección de programas activos en diferentes etapas de ejecución.
- Ejecutan múltiples programas de forma intercalada.



(C) Multiprogramación con tres programas

El *número de programas* que compiten activamente por los recursos de un sistema informático multiprogramado se denomina *grado de multiprogramación*

Los Sistemas multiprogramados son más complejos porque requieren llevar el control de:

- La memoria que ocupa cada programa.
- Tiempo de CPU que asignamos a cada programa (Planificación de la CPU).
- Planificación de los periféricos.
- Control de la CONCURRENCIA.
- Protección.

2.3 Sistemas de Tiempo Compartido

Son Sistemas multiprogramados, multiusuarios, e ***interactivos*** (*hands-on*).

- Los usuarios o programadores pueden interactuar con su programa mientras se ejecuta. Si hay **- n** - usuarios, todos los recursos del sistema son compartidos por todos.
- Los usuarios del Sistema tienen la "ilusión" de tener todos los recursos del Sistema a su disposición.
- Proporcionan un buen tiempo de respuesta de terminal.
- Utilizan planificación de reparto de tiempo.
- Están en contraposición a los Sistemas por **lotes o "*batch*"** (no interactivos)

En modo "**batch**" el usuario desde que pone su trabajo en "cola" hasta que pasado cierto tiempo no recibe los resultados, se desentiende de su programa.

Los conceptos de interactivo, en primer plano, *foreground* son equivalentes, lo mismo que `lote`, *bach*, segundo plano e *background.*

Los Sistemas actuales soportan trabajos por lotes y de tiempo compartido, es decir: Los Sistemas Operativos modernos son ***multiprogramados*** y de ***Tiempo Compartido***.

Comparación: Sistema *batch* multiprogramado versus sistema de tiempo compartido.

	Sistema *batch* multiprogramados	Sistema con tiempo compartido

Objetivo principal	Maximizar el uso del procesador	Minimizar el tiempo de respuesta
Fuente de las instrucciones al Sistema Operativo	El lenguaje de control de trabajos (*Job Control lenguaje*)	Comandos desde el terminal

2.4 Sistemas de tiempo real

El procesamiento en Tiempo Real puede definirse como un tipo de procesamiento en que la exactitud del sistema no depende sólo del resultado lógico de un cálculo sino también del instante en que se produzca el resultado.

Se utilizan en entornos en donde deban de ser aceptados y procesados un gran número de sucesos, la mayoría externa al sistema informático, en breve plazo o dentro de unos límites.

Ejemplos: Control de plantas industriales por medio de ordenador, el control de procesos y robótica, control del tráfico aéreo, mando militar.

Su prioridad es el **tiempo de respuesta corto** por encima de otra consideración.

El modo de planificación debe de permitir responder rápidamente a los eventos externos, que asíncronamente se producen y el sistema debe controlar.

La conveniencia del usuario y la utilización de recursos son una preocupación secundaria para los diseñadores de los sistemas de tiempo real.

Cada proceso tiene asignado una prioridad (planificación apropiativa basada en prioridades).

2.5 Sistemas monousuarios

Todos los recursos del sistema lo utiliza un solo usuario. Ejemplo: un Ordenador personal. La reducción del costo del *hardware,* ha hecho posible la dedicación de un sistema a un solo usuario.

2.6 Sistemas multiusuarios

Los recursos del Sistema son compartidos por más de un usuario que operan simultáneamente.

2.7 Multitarea

Sistemas en los que un usuario puede tener activas varias tareas simultáneamente.

Son posibles, además, sistemas monousuarios-multitareas, multiusuario-monotarea, multiusuario-multitarea...

2.8 Sistemas abiertos

Sistemas operativos capaces de interactuar con otros sistemas. Incorporan un conjunto de normas estandarizadas para la interconexión con otros sistemas. Pueden funcionar sin muchos condicionantes, en equipos de otros fabricantes. Los Sistema abiertos facilitan la migración de un sistema a otro y de un fabricante a otro.

Los sistemas operativos abiertos se contraponen a los sistemas operativos propietarios.

Ejemplos de normas abiertas:

- Normas ISO.
- X/open; POSIX..
- OSF (*Open Software Foundation*).
- X windows.

2.8.1 Tipos de licencias de software

Por el modos de pagos

- Freeware, que es gratuita

- Shareware: Nos ofrece un periodo de prueba gratuito para que probemos el programa, pero si decidimos usarlo de modo permanente, hay que pagar.
- Comercial: de pago
- Demos: versiones parciales en el tiempo o en funciones

Por el modos de distribución y desarrollo

Software libre (free): Es un tipo de licencia que nos permite utilizar y modificar el software como deseemos. GNU es un proyecto pata crear software libre. El tipo de licencia que ampara este software es GPL (*General Public Licencie*). Dentro del software libre hay distintas categorías, por ejmplo un software libre protegido con *copyleft*, se exige que a las modificaciones que se realicen no se le añadan restricciones. Un software libre no tiene por que ser gratuito. Para más información al respecto mirar en la dirección http://www.gnu.org/home.es.html , y la filosofía la podemos encontrar en:

http://www.gnu.org/philosophy/philosophy.es.html

2.9 Tipos de sistemas Operativos

Podemos hacer varias clasificaciones de los sistemas operativos dependiendo de distintos puntos de vista.

- - Según la configuración del *hardware* subyacente:
  - Sistemas Operativos para micro-ordenadores.
  - Sistemas Operativos para mini-ordenadores.
  - Sistemas Operativos para *mainframe*.
- Según el nivel de recursos:
  - Sistemas Operativos Monoprogramados.
  - Sistemas Operativos Multiprogramados.
- Según el tipo de iteración permitida:
  - Sistemas Operativos para procesos por lotes.
  - Sistemas Operativos Interactivos.
  - Sistemas de Tiempo Real.

2.10 Generaciones de Sistemas Operativos

GENERACIÓN CERO (mitad de la década de 1.940).

- No había sistemas operativos.

PRIMERA GENERACIÓN (Década de 1.950).

- Tubos de vacío y tableros enchufables.
- Horward Aiken, John von Neuman, J.Presper.
- Su función era el encadenamiento automático de trabajos.
- Comienzo de los procesamiento por **lotes**.
- Un trabajo tenía el control absoluto de la máquina, mientras se estaba ejecutando.

SEGUNDA GENERACIÓN (Primera mitad de la década 1.960).

- Sistemas compartido con **multiprogramación.**
- Principios del **multiproceso**.
- Tiempo compartido modo **conversacional.**
- Surgen los sistemas de tiempo real **control de procesos**.

TERCERA GENERACIÓN (1/2 década 60 a 1/2 década 70).

- Comienza con los ordenadores de la familia S/360 de IBM.
- Se introduce el concepto de **familia** de ordenadores.
- Son sistemas de uso general.

- Son complejos.
- Soportan trabajos por lotes, multiprogramación, multiproceso, tiempo real y tiempo compartido.
- Tienen complicados lenguajes de control de trabajos.
- Colas de impresión.

CUARTA GENERACIÓN (1/2 década 70 hasta nuestros días).

- Redes de datos y sistemas distribuidos.
- Interfaz gráficas amables al usuario.
- Importancia de las bases de datos.

3 Sistemas multiprocesadores

Son sistemas que tienen más de un Procesador o CPU.

Multiproceso sistemas en los que se ejecutan más de un proceso, realmente de modo simultáneo, cada uno en un procesador diferente.

Estos sistemas se justifican para conseguir:

- **Un aumento en las prestaciones:** Un sistema con N procesadores no aumenta sus prestaciones N veces. El aumento siempre es algo menor.
- **Un aumento de la fiabilidad**: Si falla un procesador, siempre hay otro que retoma sus funciones evitando la caída del sistema.

Ejemplo: Tandem - sistemas tolerantes de fallos.

Requerimientos

- Tarjeta base con más de una ranura para procesadores
- Procesadores que soporten el multiproceso
- Sistema Operativo que soporte el multiproceso

Hay dos esquemas básicos de Sistemas con Multiprocesadores:

3.1 Sistemas fuertemente acoplados o sistemas paralelos (*Tightly coupled*)

Características:

- Poseen dos o más procesadores de uso general similares y de capacidades comparables.
- Constan de un conjunto de procesadores comparten una memoria principal común y reloj.
- Se encuentran bajo el control integrado de un sistema operativo.
- La comunicación se realiza por medio de la memoria compartida.
- Sus principales usos son aplicaciones científicas.

Los sistemas con multiproceso fuertemente acoplados pueden ser:

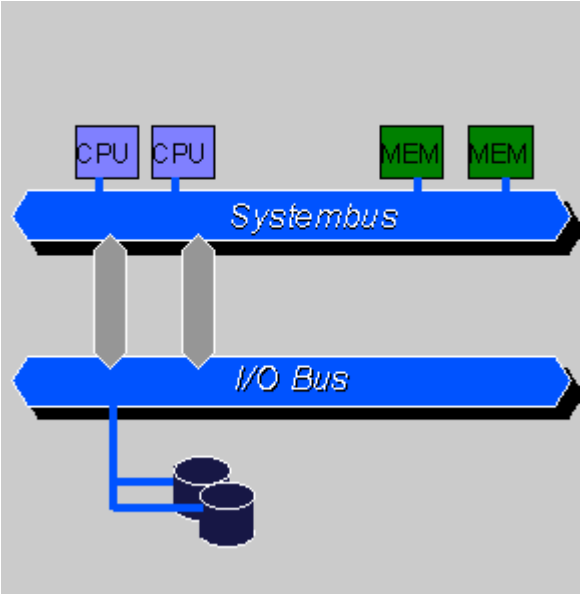
3.1.1 Multiproceso simétrico (SMP)

En estos sistemas todas las CPU´s comparten los recursos disponibles (bus del sistema, memoria, bus e/s). Hay una sola copia del sistema operativo y de las bases de datos. Cualquier procesador puede ser usado por el sistema operativo o por las aplicaciones de usuarios. El control de las operaciones de E/S debe asegurar que cada dato llega al procesador adecuado. Es el más usado.

Ejemplo: Multimax computer, cada procesador funciona con una copia de UNIX.

Dentro del multiproceso simétrico se definen distintos grados de multiproceso.

- **Paralelismo independiente**: estos sistemas tienen una planificación similar que un sistema momoprocesador, pero con más de un procesador. Los procesos de la cola que están en estado de espera, se reparten entre los distintos procesadores que disponga el ordenador.



Funciona de una forma similar a de los cajeros de un supermercado. Si hay una sola caja, todos los clientes tienen que esperar en una única cola. Si hay muchos clientes y/o el cajero/a no es muy rápido, los tiempos de espera de los clientes aumenta. Si disponemos de más de un cajero/a, la cola se reparte entre los distintos cajeros. El tiempo de espera en la cola disminuye.

El número de procesadores que pueden incorporar este tipo de sistemas es relativamente pequeño (hasta 10 o 12 procesadores). Se utilizan en aplicaciones de gestión o de calculo no muy masivo.

- **Paralelismo de grado medio o fino**. El programador debe especificar explícitamente el posible paralelismo de la aplicación. Son los sistemas Vectoriales. Ejemplos son ordenadores como los CRAY´s o los CONVEX. Su aplicación suele ser de tipo científico (previsión del tiempo).

Con un sistema SMP si un proceso requiere 10 horas de cómputo (de procesador) seguirá tardando 10 horas. Mejora porque está menos tiempo en la cola de espera. Con MPP el tiempo de cómputo de un proceso se reparte entre los distintos procesadores. Su tiempo de computo será 100/n, siendo n el número de procesadores. Insisto que sólo estoy hablando de tiempo de cómputo, no del tiempo total que tarda en ejecutarse. El tiempo total de ejecución, es la suma del tiempo que tiene el dominio de la CPU y del tiempo que está a la espera.

Windows 2000 server soporta multiproceso simétrico SMP.

3.1.2 Multiproceso asimétrico

Hay una CPU maestra que controla a las restantes CPU. En una determinada CPU se ejecuta el Sistema Operativo y en el resto de los procesadores sólo las aplicaciones de Usuarios. Es menos eficiente.

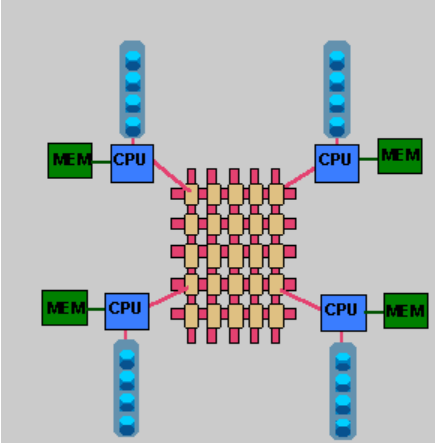
3.2 Sistemas débilmente acoplados (*Loosely coupled)*

Características:

- Los procesadores no comparten ni memoria ni reloj.
- Cada procesador tiene su memoria local.
- Las comunicaciones entre los procesadores se realizan por medio de líneas de comunicaciones (buses o líneas de comunicación).
- Los procesadores pueden ser desde pequeños microprocesadores hasta grandes sistemas. Se les conoce también como sistemas multicomputadores.

3.2.1 MPP (*Massively Parallel Processing*)

Cada nodo de la CPU tiene sus propios recursos (memoria, procesador..). Cada procesador ejecuta una copia del sistema operativo. El número de procesadores de estos sistemas es alto (pueden llegar a varios centenares)



3.2.2 Sistemas distribuidos

Se entiende por **Sistema Distribuido** aquel en que la **inteligencia** del Sistema Informático esta repartida por distintos procesadores. Ejemplo: una red de ordenadores.

Ventajas de los sistemas distribuidos

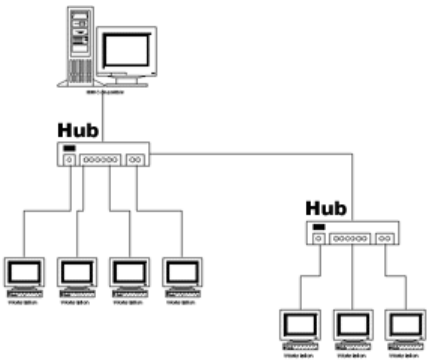
- Compartición de recursos.
  - Se puede utilizar otro dispositivo que esté conectado a otro nodo de la red.
- Aumenta la velocidad de computación.
  - Se distribuye el trabajo entre los distintos procesadores, o nodos.
  - Fiabilidad.
  - Si falla un ordenador otro puede seguir con la tarea.
- Comunicación.

Ejemplos:

Red de ordenadores

Una red de ordenadores es un conjunto de ordenadores unidos por un "medio físico" con el fin de intercambiar información y compartir recursos.

Podemos poner como ejemplo una red de ordenadores en que el trabajo se distribuye entre los distintos ordenadores. Cada ordenador (servidor) se puede dedicar a una tarea especializada.



Clustering

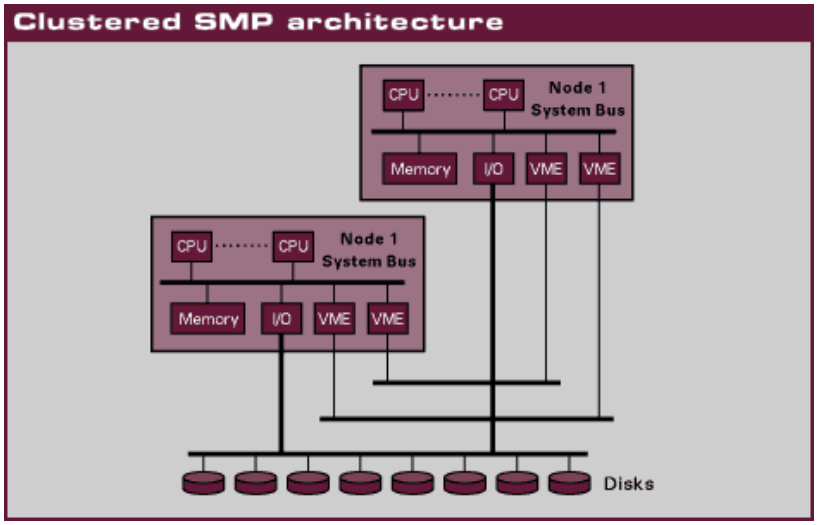
Un cluster es un grupo de múltiples ordenadores unidos, mediante una red de alta velocidad, de tal forma que el conjunto es visto como un único ordenador más potente por los usuarios y las aplicaciones. Se espera de un cluster que presente combinaciones de los siguientes servicios:

1. Alto rendimiento (High Performance)
2. Alta disponibilidad (High Availability)
3. Equlibrio de carga (Load Balancing)
4. Escalabilidad (Scalability)

Los ordenadores pueden tener todos la misma configuración de *Hardware* y Sistema Operativo (Cluster Homogéneo), diferente rendimiento pero con arquitecturas y sistemas operativos similares (cluster semi-homogéneo) o tener diferente hardware y sistema operativo (cluster heterogéneo), lo que hace más fácil y económica su construcción.

Para que un cluster funcione como tal, no basta solo con conectar entre si los ordenadores, sino que es necesario proveer un sistema de manejo del cluster, el cual se encargue de interactuar con el usuario y los procesos que corren en él para optimizar el funcionamiento.

El *clustering* se emplea para procesamiento paralelo, balanceo de carga de trabajo y tolerancia a fallos. Son ordenadores que se distribuyen la carga de trabajo entre todos.

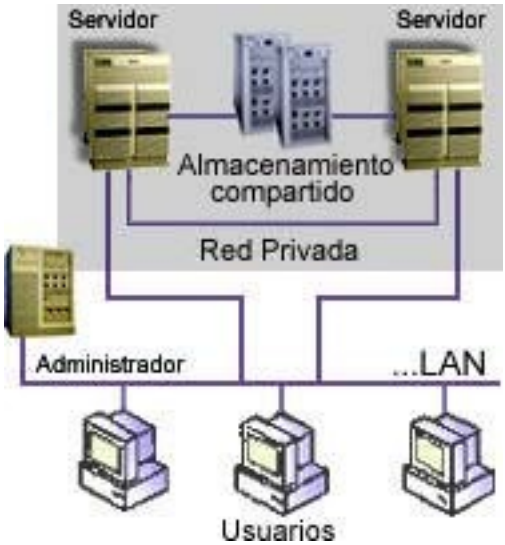


Características

- Múltiples servidores independientes, funcionando como un sistema servidor único.
- Se maneja como un sistema único. Los servidores tienen un nombre común.
- Los servidores son disponibles a todas las máquinas conectadas a la red.
- Pueden tolerar fallas de componentes.

Se pueden agregar componentes, sin interrumpir a los usuarios (*hot swap*)

ejemplo de producto comercial es Orion de Novel, permite el uso de hasta 16 servidores de multiprocesador, basados en la arquitectura Intel, para trabajar en conjunto como un solo sistema y así ofrecer una administración de red más sencilla y un acceso a los recursos más rápido.



¿Por qué Clustering?

Alta disponibilidad de Recursos: Las aplicaciones Cliente/Servidor recaen en la disponibilidad de los servicios de la red. Estos servicios son proporcionados por los recursos. Si los recursos no están disponibles debido a fallos en aplicaciones o fallos del *hardware*, el trabajo del usuario es interrumpido. *Clustering* incrementa la disponibilidad de estos recursos del servidor.

Escalabilidad: Recursos de aplicación, de entrada/salida y CPU pueden ser añadidos, para expandir eficientemente la capacidad del sistema sin interrupción del servicio al usuario. Esto se traduce en un acceso confiable a recursos del sistema e información, así como protección de la inversión de los recursos de *hardware* y *software*.

Administración centralizada: En un ambiente de servidores comunes, se utilizan varias herramientas administrativas para identificar los servidores en la red, monitorear sus contenidos y actividades. Sin embargo en un ambiente de *cluster*, la administración de aplicaciones y servicios puede ser centralizada, a través del uso de una herramienta de administración y monitoreo de redes como *HP Open View*.

Componentes de un Cluster

Un cluster necesita de varios componentes de software y hardware para poder funcionar. A saber:

- Nodos (los ordenadores)
- Sistemas Operativos
- Conexiones de Red
- Middleware (capa de abstracción entre el usuario y los sistemas operativos)
- Protocolos de Comunicación y servicios.
- Aplicaciones (pueden ser paralelas o no)

Nodos: Pueden ser simples ordenadores, sistemas multi procesador o estaciones de trabajo (workstations).

Sistema Operativo: Debe ser de fácil uso y acceso. Debe permitir múltiples procesos y usuarios. Ejemplos

- GNU/Linux
- Solaris
- Windows NT
- Cluster OS etc

Conexiones de Red: Pueden conectarse mediante una simple red Ethernet con placas comunes, o utilizarse tecnologías especiales de alta velocidad como Fast Ethernet, Gigabit Ethernet, Mirynet, SCI, etc.

Middleware: Es un *software* que generalmente actúa entre el Sistema Operativo y las aplicaciones con la función de proveer los siguientes requerimientos de un cluster:

- Una interfaz única de acceso al sistema, denominada SSI (Single System Image), la cual genera al usuario la sensación de que utiliza un único ordenador muy potente
- Herramientas para la optimización y mantenimiento del sistema: migración de procesos, checkpoint-restart (congelar uno o varios procesos, mudarlos de servidor y continuar su funcionamiento en el nuevo host), balanceo de carga, tolerancia a fallos, etc.
- Escalabilidad: debe poder detectar automáticamente nuevos servidores conectados al cluster y proceder a su utilización.

Existen diversos tipos, como por ejemplo: MOSIX, OpenMOSIX, Condor, OpenSSI, etc.

El middleware recibe los trabajos entrantes al cluster y los redistribuye de manera que el proceso se ejecute más rápido y el sistema no sufra sobrecargas en un servidor. Esto lo realiza mediante políticas definidas en el sistema (automáticamente o por un administrador) que le indican donde y cómo debe distribuir los procesos y por un sistema de monitorización del cluster, el cual controla la carga de cada CPU y la cantidad de procesos en él. El middleware también debe poder migrar procesos entre servidores, con el fin de:

- Balancear la carga: si un servidor está muy cargado de procesos y otro está ocioso, pueden pasarse procesos a este último para liberar al primero y optimizar el funcionamiento
- Mantenimiento de servidores: si hay procesos corriendo en un servidor que necesita mantenimiento o una actualización, es posible migrar los procesos a otro servidor y proceder a desconectar del cluster al primero.
- Priorización de trabajos: en caso de tener varios procesos corriendo en el cluster, pero uno de ellos de mayor importancia que los demás, puede migrarse este proceso a los servidores que posean más y/o mejores recursos para acelerar su procesamiento.

Computación distribuida (GRID computing) De Wikipedia

La **computación distribuida**, informática en rejilla (GRID), es un nuevo modelo para resolver problemas de computación masiva utilizando un gran número de ordenadores organizados en racimo incrustados en una infraestructura de telecomunicaciones distribuida.

La computación en rejilla ha sido diseñada para resolver problemas demasiado grandes por cualquier simple superordenador, mientras mantiene la flexibilidad de trabajar en multiples problemas más pequeños. Por tanto, la computación en rejilla es un entorno multi-usuario.
Debido a esta razón, las técnicas de autorización segura son esenciales para permitir que los recursos informáticos sean controlados por usuarios remotos (distantes).

La informática en rejilla consiste en compartir recursos heterogéneos (basadas en distintas plataformas, arquitecturas de equipos y programas, lenguajes de programación), situados en distintos lugares pertenecientes a diferentes dominios de administración sobre una red que utiliza estándares abiertos. Dicho brevemente, consiste en virtualizar los recursos informáticos.
En términos de funcionalidad, las Rejillas se clasifican en Rejillas computacionales (incluyendo las rejillas In terms of functionality, Grids are classified into Computational Grids (incluyendo rejillas de barrido de la CPU) y en Rejillas de Datos.

La rejilla ofrece una forma de resolver los problemas de Gran Reto como el plegamiento de las proteinas y descubrimiento de medicamentos, modelización financiera, simulación de terremotos, inundaciones y otras catástrofes naturales, modelización del clima/tiempo, etc. Ofrecen un camino para utilizar los recursos de las tecnologías de la información de forma óptima en una organización.

- http://www.gridcomputing.com/
- http://irisgrid.rediris.es/

3.2.3 Escalas de los sistemas

1. Ordenador aislado
2. Multiproceso Simétrico (SMP)
3. Red de ordenadores
4. *Clustering*

5. *GRID*
6. Multiproceso masivo (MPP)

4 Sistemas Operativos Actuales

4.1 Sistemas operativos tipo Windows

Sistemas para estaciones de trabajo o clientes

- Windows 95
- Windows 98
- Windows Mileniun
- Windows NT 4.0 workstation
- Windows XP

- OS/2 (IBM)
- Mac OS 9 (macintosh)
-

- Windows 2000 profesional.
- Windows XP profesional
- Linux

Sistemas para ordenadores de mano (*Palm Desktop*)

Ordenadores de mano. Tiene la misma interfaz que el resto de los sistemas Windows.

- Windows CE
-
- PalmOS (3M)

-

Sistemas para servidores

Adaptados para equipos servidores. Requieren mayor grado de seguridad y de sistemas de compartición de recursos.

- Windows NT server 4.0
- Windows 2000 server
- Windows XP
- OS/2 server
- Mac OS X Server
- AS/400
- BeOS5 (para gestión de grandes ficheros multimedia)

4.2 Sistemas Operativos tipo Unix

Sistema muy estable. Muy utilizado en servidores. Existen varias versiones de distintos fabricantes con pequeñas variaciones. Hay variantes para tiempo real, multiproceso etc., especialmente adaptado para comunicaciones.

- Digital UNIX
- HP-UX
- AIX (IBM)
- SCO Unixware
- Sun Solaris (de SUN).

Es la implementación de Sun del Unix System V Release 4.

Sus características:

- Escalabilidad, multiproceso y *multithreading*, compatibilidad binaria a lo largo de toda la gama
- Disponible para la arquitectura SPARC, y arquitectura INTEL
- La versión actual es la 2.7
- Desarrollo: lenguajes, entornos y tecnologías. Seguridad: control de acceso, *firewalls*, encriptación,...
- Gestión de Red: administración centralizada, gestión de clientes y de almacenamiento.
- Conectividad: Redes de PCs, WANs y otros sistemas.
- Para más información mire en www.sun.es

StarOffice

StarOffice es un potente y completo paquete de ofimática que contiene:

- `StarOffice Writer` (Procesador de Textos)
- `StarOffice Calc` (Hoja de Cálculo)
- `StarOffice Impress` (Para sus presentaciones)
- `StarOffice Draw` (Para la creación de todo tipo de gráficos)
- `StarOffice Image` (Para el tratamiento de Imágenes)
- `StarOffice Schedule` (Gestión de Agendas y listas de tareas)
- `StarOffice Mail` (Gestión del Correo Electrónico)
- `StarOffice Base` (Bases de Datos)
- `StarOffice Discussion` (Noticias en Internet)
- `StarOffice Math` (Manejo de Fórmulas Matemáticas Complejas)

`StarOffice` está disponible para las plataformas Solaris SPARC/Intel, Linux, OS/2, Windows 95/98/NT

Contiene filtros para la importación de ficheros de `Microsoft Office` y de otros formatos. De esta forma, los usuarios de cualquier plataforma pueden acceder, modificar e incluso exportar ficheros fácilmente.

La versión libre de este software es Open office , que podemos encontrar en http://es.openoffice.org/index.html o www.openoffice.org

4.3 Linux

Sistema operativo multiusuario, multitarea.(www.linux.org). Entornos gráficos. Gratuito y con infinidad de *software* gratuito de todo tipo, Ofimático (*OpenOffice*), compiladores, aplicaciones etc.; Es bueno, bonito y gratuito. Para más información mire en la dirección: http://lucas.hispalinux.es/

Hay distintas distribuciones de Linux:

distribuciones	dirección
Red Hat	www.redhat.com
Suse	www.suse.de/es
Debian	www.debian.org
Slackware	www.slackware.com
MandrakeLinux	www.mandrakelinux.com
Knoppix	www.knoooix.org
Madrid_linux	www.educa.madrid.org
Linex	www.linex.org
Guadalinex	www.guadalinex.org
Ubuntu	http://www.ubuntu-es.org/

4.4 Freebsd

Otro UNIX abierto es FeeBSD. a diferencia de Linux, FreeBSD deriva directamente de una versión UNIX estándar (BSD UNIX)desarrollado por la universidad de California de Berkeley. Es un sistema abierto distribuido bajo copyright FreeBSD. Es un producto muy estable. (www.freebsd.org). y http://www.freebsd.org/es/index.html

4.5 Sistemas Operativos Propietarios

Como su propio nombre indica, son sistemas propietarios y específicos de un determinado fabricante. Funcionan exclusivamente en equipos del fabricante. Hacen difícil la migración a equipos de otro fabricante. Son sistemas para *mainfranes* (grandes sistemas centrales) o equipos muy específicos.

- Open VMS (Dígital)
- MVS (IBM)
- S/390 (IBM)

4.6 Sistema Operativo de Red (NOS *Net Operating System*)

El concepto de sistema operativo de red es difuso. Hay muchos sistemas operativos, los que nosotros hemos definido como sistemas operativos de servidores, que se entienden que son sistemas operativos que permiten trabajar en red. Desde ese punto de vista son sistemas operativos de red claramente. Estos que mencionamos ahora son sistemas operativos diseñados exclusivamente para trabajo en red, o dotar a otros sistemas operativos de una capa de *software* que les permita trabajar en red y mejorar sus servicios y prestaciones. Los más conocidos son:

- Banyan VINES
- Netware (Novel)
  - Novell Open Enterprise Server

Estos sistemas están obsoletos.

5 Cuestiones

Pregunta:

1. ¿Las siguientes mejoras en el rendimiento incrementan la productividad, hacen disminuir el tiempo de ejecución, o ambas cosas?
   a) Ciclo de reloj más rápido
   b) Múltiples procesadores para tareas separadas
   c) Procesamiento paralelo de problemas científicos?
      - Productividad de un sistema: número de tareas ejecutadas por unidad de tiempo
      - Tiempo de ejecución: tiempo que tarda en realizar un proceso
2. ¿Cuáles son las funciones principales de los sistemas operativos?
3. Haz un esquema de las "capas" de un sistema informático. ¿Dónde se sitúa el Sistema operativo?
4. Concepto de llamadas al sistema
5. ¿Todos los sistemas operativos tienen las mismas llamadas al sistema?
6. Nombra un sistema operativo (antiguo) que tenga solo una interfaz con el usuario por medio de comandos. Nombra otro sistema que tenga una interfaz gráfica.
7. Nombra varios módulos o componentes típicos de un sistema operativo.
8. ¿Cuál es la principal característica del modo de explotación por lotes?
9. Características de un sistema multiprogramado.
10. La afirmación: Los sistemas operativos actuales combinan el modo de explotación batch e interactivo es cierta o falsa.
11. Características de un sistema operativo de tiempo compartido.
12. ¿Los sistemas de tiempo compartido son multiprogramados?
13. ¿Todos los sistemas multiprogramados son de tiempo compartido?
14. Pon un ejemplo de un sistema operativo monousuario y monotarea. Explica brevemente estos conceptos.
15. Pon un ejemplo de un sistema operativo multitarea, multiusuario. Explica brevemente estos conceptos.
16. ¿Qué diferencia hay entre sistemas multiprogramados y sistemas multiprocesadores?
17. Características de los sistemas fuertemente acoplados
18. Diferencia entre multiproceso simétrico y asimétrico.
19. Características de los sistemas SMP. ¿Qué significan estas siglas?
20. Características de los sistemas MPP ¿Qué significan estas siglas?
21. Compara los sistemas SMP con los MPP.
22. Características de los sistemas débilmente acoplados.
23. Concepto de *Clustering* de ordenadores
24. Concepto de computación GRID
25. Diferencia entre una red de ordenadores y un *cluster de ordenadores.*
26. *Diferencia entre un cluster de ordenadores y un sistema GRID.*
27. Escala de los sistemas (atendiendo al número de procesadores)
28. Características de los sistemas de tiempo real
29. ¿Puede ser un sistema monousuario y multitarea?
30. El Windows XP es monotarea o multitarea
31. Nombra tres sistemas operativos que no sean de Microsoft
32. ¿Qué diferencia hay entre sistemas abiertos y sistemas propietarios?
33. ¿Puede funcionar una aplicación en cualquier sistema operativo?

34. ¿El Windows XP admite multiproceso simétrico?
35. ¿Qué es un sistema tolerante a fallos?
36. Concepto de nivel de multiprogramación
37. Asocias Sistemas fuertemente acoplados, y Sistemas débilmente acoplados con
    - Sistemas que no comparten ni memoria ni reloj
    - Sistemas que comparten memoria y reloj

Actividades

- Busca en webopedia.com los conceptos de clustering, MPP, SMP, GRID
- Busca en las revistas de informática de biblioteca las características del Windows XP.
- Busca información sobre distribuciones LINUX