

INDICE DE SISTEMAS OPERATIVOS

A

Abstracción · 5

Aplicaciones · 6

B

Bufferring · 9

C

Cargador · Véase Loader

código fuente · 6

código objeto · 6

Compiladores · 6

Conectividad · 10

D

DBMS · 10; 18

E

Ensambladores · 6

F

FIRMWARE · 6

FUNCIONES PRINCIPALES · 5

I

I.O.C.S · 6

Información · 4; 5; 27; 28

Informática · 5

instrucciones privilegiadas · 8

Interpretes · 6

J

JAVA · 10

JCL · 7

L

LA EVOLUCION DE LOS SISTEMAS OPERATIVOS

1ª Generación · 7

2ª Generación · 7

3ª Generación · 7

4ª Generación · 7

limitados por E/S · 9

limitados por la CPU · 9

Loader · 6

M

Máquina Virtual · 10

Memoria Interna · 5

Memoria Secundaria · 5

Modo supervisor · 6

Modo usuario · 6; 8

MULTIPROGRAMACION · 9

N

Núcleo · 6

P

PROCESADOR · 5

procesamiento Off Line · 8

Programas de control · 4; 5; 6

Programas de servicio · 4; 6

Protección de dispositivos · 8

Protección de memoria · 8

R

red · 10; 11; 18; 24

S

Sistema Informático · 5

SISTEMAS OPERATIVOS

Introducción · 3

SO de red · 10

SO distribuidos · 10

SPOOLING · 9

Subsistema físico · 4; 5

Subsistema lógico · 4; 5

Supervisor · 6; 8; 16

T

TECNICAS O SISTEMAS DE TIEMPO REAL · 10

Time-Sharing

Tecnicas de tiempo compartide · 10

TRABAJO · 7

Traductores · 6; 7

U

Utilidades · 6

SISTEMAS OPERATIVOS

Introducción a los sistemas operativos

Gestión de ficheros

Gestión de procesos

Gestión de memoria

Gestión de E/S y Dispositivos

Principios de diseño de un SSOO

TEMA 1 Introducción a los sistemas operativos

Los sistemas operativos se ocupan de la gestión de los recursos de la máquina y gestionar los programas.

Las instrucciones de alto nivel se traducen a instrucciones sencillas.

Los datos de un programa pueden almacenarse en diferentes sitios como la memoria del ordenador, dispositivos secundarios (discos duros, unidades de disco o cinta etc.), registros.

Los dispositivos secundarios son del orden de cien veces más lentos que la memoria.

Los programas exigen estar en la memoria del ordenador para ser ejecutados pero al ser la memoria más pequeña que los dispositivos secundarios, puede darse el caso de que un programa se encuentre en un determinado momento parte en memoria y parte en el dispositivo secundario. Por tanto los programas no se ejecutan en los dispositivos secundarios.

Los registros que son el tercer nivel de almacenamiento contienen únicamente datos para el programa (nunca contienen instrucciones propias del programa). Estos registros son aún más rápidos que la memoria (del orden de milisegundos) por tanto en ellos introduciremos los datos de los programas a los que se acceden más veces.(ejemplo: El contador de instrucciones.)

La caché del ordenador no debe confundirse con los registros ya que esta memoria esta destinada para almacenar cosas leídas y escritas recientemente y en ellas se almacenan también instrucciones.

Otro nivel de memoria puede ser la memoria del procesador que es mucho más rápida (del orden de nanosegundos).

Todo sistema operativo debe proporcionar el conocimiento de la máquina.

Debe de servir como elemento de comunicación entre los programas, el usuario y su Hardware; interactúa con el programa para que el usuario se olvide un poco de como es la máquina.

Debe de actuar como gestor de recursos y hacer un uso óptimo del Hardware, para que un programa funcione lo más rápidamente posible.

Debe tener capacidad de evolución (Para el control de recursos mayores)

Los programadores deben saber además la cantidad de registros de memoria que posee el ordenador así como si disponemos o no de un disco duro, CD-ROM, etc..

Los sistemas operativos deben ser lo suficientemente cómodos para los usuarios finales y flexibles para los programadores.

Deben aparecer servicios para las siguientes áreas.

- Creación de programas (Editores, compiladores)
- Servicio para la ejecución de programas (Permitir a los usuarios la utilización de los recursos del sistema).
- Acceso a los dispositivos de E/S
- Acceso a los archivos (Ficheros) y establecer un control sobre éstos (Estos son tus archivos y estos no)
- Debe controlar y responder a errores
- Debe proporcionar estadísticas.
- Un ordenador es un conjunto de recursos destinado al traslado, procesamiento y almacenamiento de datos.
- El SSOO debe gestionar esos recursos de transporte, debe controlar el funcionamiento de los programas y es suyo mismo (es como un programa pero debe ir más lejos)
- En la mayor parte del tiempo el SSOO funciona como un programa convencional ejecutado en la CPU (instrucciones del SSOO) pero debe ser capaz de ceder la ejecución a otros programas y volver a recuperarla cuando sea necesario.
- ¿Como conseguimos que si un programa se cuelga el SSOO libere los recursos si en este momento el SSOO no está en la CPU?
- El SSOO dispone de una gran parte de programa que funciona de forma convencional y otra pequeña parte de casi Hardware (Firmware) que se encarga de las tareas de control básicas.
- **INFORMATICA**
- **Sistema informático**
- Subsistema físico (ordenador)
- **Procesador**
- **Memoria interna**
- **Memoria secundaria**
- **Periféricos**
- Subsistema lógico (Sistema operativo)
- **Programas de control o de sistema**
- **Programas de servicio**
- **Información**
- Programas de aplicación
- Datos
- **Informática:** Conjunto de conocimientos científicos y técnicos que hacen posible el tratamiento automático de la información.
- **Información:** Conjunto de elementos capaces de expresar un conocimiento determinado.
- **Sistema Informático:** Conjunto de elementos físicos y lógicos capaces de procesar la información que el usuario necesita y controlar de la forma más eficaz posible los recursos disponibles.
- **Subsistema físico:** Ver esquema
- **Subsistema lógico:** Ver esquema
- El subsistema físico se compone de elementos de control (1), de memoria (2), elementos operativos (3) y elementos de entrada salida (4).
- **PROCESADOR**
- Lleva a cavo la ejecución de los programas.
- | |
|----------------------------|
| • Unidad Lógica (1) |
| • ALU (2) |
| • Elementos de memoria (3) |
| • Unidades de E/S (4) |
- **Memoria Interna:** Lugar de almacenamiento de programas y datos. Es volátil (Se borra sin alimentación) RAM (Random Access Memory)

- **Memoria Secundaria:** Al ser la RAM volátil los programas y los datos se guardan permanentemente en (HDD, cintas magnéticas, discos) no se borran (no volátil)
- Para el proceso de la información y programas los programas y datos se pasan de la memoria secundaria a la memoria interna.
- **Periféricos:** Permiten introducir y sacar resultados (teclado, impresora, pantalla, etc..)
- **¿COMO FUNCIONA UN PROCESADOR?**
- Procesador * Ejecución de las instrucciones indicadas por el usuario como parte de la actividad de procesamiento.
- La unidad de eficiencia se mide en MIPS (millones instrucciones por segundo).
- La unidad de control tiene como función principal ejecutar las distintas instrucciones de un programa. En los registros de indica la instrucción que se está ejecutando en cada momento (Contador) y en otros se decodifican las instrucciones.
- Las instrucciones están formadas por códigos de operación + operandos.
- Instrucciones = Códigos de operación + operandos
- Decodificar: Saber lo que quiere hacer la instrucción con los datos de las direcciones Dir1 y Dir2 (Ej. ADD DIR1 DIR2) (Sumar los números de las direcciones dir1 y dir2)
- La suma la realiza la ALU (Unidad Aritmético-Lógica) Se encarga de este tipo de operaciones y posee un circuito para cada una de ellas (+ - * /).
- También existen operaciones que no son aritméticas (ej. Operaciones de E/S) En el procesador existe una interface para cada una de los periféricos conectados al sistema.
- Las instrucciones que realiza el programa del secuenciador para cada instrucciones de usuario se denominan instrucciones u órdenes.
- **FUNCIONES PRINCIPALES**
- Recoger la información de la memoria interna
- Las ejecuta de una en una.
- **Programas de control:** Gestionan los recursos físicos y (al ser este proceso complejo para el usuario) liberan al usuario del control de una impresora, cintas magnéticas, etc..
- **Abstracción:** Eliminar detalles que complican al usuario. (Liberan)
- **Programas de servicio:** Hacen que el usuario se pueda mover por ese entorno (Traductores, programas de utilidades para manejo y gestión de ficheros de la memoria secundaria)
- **Programas de aplicación:** Las realiza el usuario para resolver sus propias actividades.
- **1) PROGRAMAS DE CONTROL**
- **Núcleo (Kernel):** Conjunto de programas que modelan el comportamiento del sistema operativo
- **Monolítico:** Integra en un solo código binario todas las funciones del SO.
- **Microkernel:** Solo incluye las funciones básicas del SO, y accede dinámicamente a cualquier extensión del SO.
- **Supervisor:** Componente software que se encarga de supervisar la ejecución de aplicaciones
- **Modo supervisor:** Da permiso para activar todas las funciones del SO.
- **Modo usuario:** Solo tiene disponible un conjunto de estas funciones.
- **I.O.C.S. (Input Output Control System):** Sistema de control E/S. Es un bloque del SO que se encarga de gestionar todas las operaciones de E/S
- **Cargador (Loader):** es aquel componente software cuya misión es traer de memoria secundaria las aplicaciones del usuario hacia memoria interna.
- **2)PROGRAMAS DE SERVICIO**
- Tienen por misión proporcionar un entorno de ejecución adecuado para las actividades del usuario.
- **1.Traductores:** Tienen por objeto convertir las aplicaciones de usuario en un lenguaje nativo del procesador (Lenguaje que se denomina código máquina)
- El código máquina es específico de cada tipo de ordenador.
- **Tipos de traductores:**
- **Ensambladores:** El lenguaje ensamblador (assembler) utiliza una serie de códigos mnemotécnicos que se corresponden de forma unívoca a las instrucciones del lenguaje máquina.
- **Compiladores:** Se suele traducir de un lenguaje de alto nivel a código máquina y además la

- traducción es completa antes de ejecutar el programa.
- Fuente Compilador objeto
- **Interpretes:** La traducción se efectúa instrucción a instrucción, con el objetivo de ejecutarlas rápidamente.
- Fuente Interprete Ejecución
- Lenguaje de ALTO NIVEL: El nivel de abstracción ofrecido, permite al usuario expresar la actividad de procesamiento en un lenguaje natural.
- Lenguaje de BAJO NIVEL: Se aproxima al lenguaje máquina.
- {Código máquina – Ensamblador C Pascal – Fortran – Modula}
- Al conjunto ordenado de instrucciones escritas en lenguaje de alto nivel o lenguaje ensamblador se le denomina **código fuente**.
- Al conjunto de instrucciones que se obtienen del proceso de traducción de le denomina **código objeto**.
- **2.Utilidades:** Los programas de utilidad tienen por objeto la manipulación de archivos.
- **3.Aplicaciones:** Son aquellos programas que realizan actividades específicas que el usuario desea encomendar al sistema.
- **3)FIRMWARE**
- Es un conjunto de programas que se almacenan en código máquina en dispositivos de memoria no volátil. ROM En el PC BIOS (Basic Input Output System) Ese conjunto de código de la BIOS activa el POST (Power on self test), que se entiende como el proceso de autodiagnóstico de puesta en marcha. Es lo primero que se efectúa. (proceso físico).
- **Propósitos del Firmware:**
- 1.Carga del SO: Es lo mínimo que se necesita para que pueda tomar el control el SO. IPL (Initial Procesal Loading)
- 2.Aumentar la capacidad (posibilidades) del sistema.
- **DIAGRAMA DE NIVELES DE UN SISTEMA INFORMÁTICO**
- 4. Usuario
- Encontramos 3 módulos que forman un conjunto que tiene como destinatario al usuario.
- Se dice que un determinado nivel utiliza los servicios que le ofrece el nivel situado debajo para ofrecer servicios mejorados al nivel situado encima.
- El usuario nunca accede directamente al SO. Ni las aplicaciones al Hardware.
- **Interface:** Elementos de separación entre subsistemas (es decir, la línea que separa los módulos).
- **TRABAJO:** Conjunto de datos + todos los programas que los procesan
- Un trabajo esta compuesto por un número de pasos que son cada programa que procesa la información. Se realizan de forma secuencial, esto es, los trabajos van llegando de forma secuencial y se forma lo que se llama la cola de trabajo (o cola del sistema) que en principio se ejecutan de forma secuencial según han ido llegando.
- Esta cola se realiza porque en un momento determinado una aplicación utiliza ciertos recursos que no pueden ser utilizados al mismo tiempo por otras aplicaciones. Se utiliza el lenguaje JCL (Job Control Lenguaje) para modelar los trabajos.
- **LA EVOLUCION DE LOS SISTEMAS OPERATIVOS**
- **1ª Generación (1945–1955)**
 - ◆ Todo se hacía en lenguaje máquina (El programador introduce los datos en binario puro)
 - ◆ Todas las funciones tenían que estar implementadas a nivel hardware (Si se quiere sumar debía estar implementada dicha operación en el hardware).
 - ◆ No se conocía ningún lenguaje de programación.
 - ◆ No existían los SSOO (Nadie llevaba los programas a memoria para su ejecución)
 - ◆ Sólo existían aplicaciones de tipo militar.
- **2ª Generación (1955–1965)**
 - ◆ Se introdujo el Transistor (Transfer Resistor). Existencia de transferencia. Permite la comunicación rápida entre elementos Hardware. Dan más estabilidad al sistema.
 - ◆ Aparece por primera vez el lenguaje ensamblador (Código mnemotécnico)
 - ◆ Aparece el JCL que me permite ejecutar de forma automática cada tipo de trabajo.

- ◇ Empiezan a existir los lenguajes de alto nivel (Cobol, Fortran)
- ◇ Aparecen componentes standard (Modelos en los que se conocen las E y S pero no su interior)
- ◇ Aparecen nuevos dispositivos (Tarjetas perforadas, cintas magnéticas), por lo que se crean los "Drivers". Se crea una entrada para los drivers exteriores.
- ◇ Aparece el IOCS (I/O Control System) para controlar todos estos dispositivos.
- ◇ TECNICAS QUE APARECIERON PARA APROVECHAR EL TIEMPO DE LA CPU
- ◇ 1.- Contratar los servicios de un operador profesional. Especialista que se dedica a activar los distintos trabajos que le ofrecen los programadores. Con esta técnica se trata de separar al programador del sistema.
- ◇ 2.- Procesamiento de trabajos agrupados. (Se aprovechan unos procesos para varios trabajos BATCH ¿Por que no lo intentamos hacer de forma automática? aparece para ello el lenguaje JCL que nos permitirá definir el trabajo que hay que realizar, tareas necesarias para ello, programas que se encargan de esas tareas y ficheros de datos que se deben usar en dichos trabajos.
- ◇ **MONITOR:** Primera implementación conocida de un SO. Conjunto de módulos que se componen de:
 - Traductores de lenguaje
 - Interprete de JCL
 - Controlador de dispositivos IOCS
 - Gestor de interrupciones (Básicamente cuando un dispositivo quiere avisar de algo al procesador se utilizan las interrupciones. Con independencia de los procesos que este realizando el procesador, los para y hace caso de las interrupciones (Ej: Se acaba la cinta. El procesador se encarga de avisar al usuario.)
 - Se empiezan a desarrollar mecanismos de protección de sistema.
 - **Protección de dispositivos:** Se trata de impedir que el usuario acceda directamente a los dispositivos.
 - **Protección de memoria:** Se trata de impedir que un usuario modifique los contenidos de las posiciones de memoria ocupadas por el monitor.
 - Para conseguir estos tipos de protección se diseña en hardware dos modos de ejecución del sistema, completamente diferentes.
 - 1.- Modo monitor, Supervisor ó Sistema
 - 2.- Modo usuario
 - Estos modos de funcionamiento se consiguen añadiendo al código máquina un bit a las instrucciones (un 0 si es de monitor, un 1 si es de usuario).
 - Las instrucciones que solo se ejecutan en modo monitor (con el bit 1) se denominan **instrucciones PRIVILEGIADAS**.
 - ¿Que pasa con la memoria? ¿Como la protejo?
 - Básicamente el monitor lleva internamente cuales son las instrucciones inicial y final de cada módulo de programa. Debe saber cuantos programas hay y que rango ocupan. Si una aplicación desea acceder a un dispositivo debe hacerlo mediante las CALL SYSTEM de esta forma el monitor indica al procesador lo que debe hacer.
 - Las CALL SYSTEM son instrucciones privilegiadas y como estas se producen dentro del código de usuario, se produce una interrupción, se cede el control a una parte del monitor y después se cede el control a la instrucción siguiente de la que provocó la interrupción.
 - Mientras un dispositivo de E/S esta funcionando el procesador está

inactivo un periodo que depende de la velocidad del dispositivo.

- **TECNICAS QUE SE UTILIZARON PARA ELIMINAR EL TIEMPO PERDIDO**

- Las técnicas utilizadas para la eliminación del tiempo perdido de la CPU mientras se utilizan los dispositivos son:
- LECTOR CPU Impresora
- **A) Se diseñó el procesamiento Off Line.** En esta técnica se pasa la información de dispositivos lentos a otros más rápidos
- LECTOR Disp. Rápido CPU Disp. Rápido Impresora
- Este tipo de proceso se realiza de forma autónoma (Procesos satélites) Se realizan con independencia de la CPU. Se realizan a nivel físico o incluso se utiliza otra CPU.
- El SO trata como siempre de optimizar la máquina con independencia de cambiar la máquina y todas las optimizaciones de realizan de forma invisible.
- El procesamiento Off line no permite hacer operaciones simultáneas del procesador y de E/S. Si deseamos lograr la superposición de ambas cosas es necesario incluir elementos hardware que lo permitan (Ej. Otro procesador satélite.)
- La actividad de procesamiento del usuario se puede hacer de dos modos:
 - – Dispositivos especiales: Se introduce una pequeña CPU entre el dispositivo lento y el otro más rápido.
 - – Procesamiento satélite: Toda la operación de E/S se realiza a través del modo IOCS. En esta situación no es necesario modificar el software creado en un sistema.
- Se puede incrementar la utilización de la CPU utilizando varias actividades de procesamiento Off Line. Un trabajo específico tarda más en ejecutarse.
- Existen otros tipos de optimización:
- **B) Buffering** Se basa en la utilización de buffers (elementos de almacenamiento intermedio). Esta técnica consiste en situar en un buffer la información leída de un dispositivo de entrada, información que podrá ser procesada mientras el dispositivo comienza a leer la siguiente información.
- La CPU puede leer un dato del disco pero se leen 20 y se llena el buffer
 - ♦ Existen los trabajos **limitados por E/S** La velocidad de ejecución esta limitada por la velocidad de los dispositivos.
 - ♦ Los **limitados por la CPU** Los buffers de entrada están siempre llenos y los dispositivos de salida siempre vacíos. (La CPU no es capaz de mantener el ritmo de trabajo de los dispositivos de E/S).
 - ♦ La gestión de los buffers es tarea del SO, concretamente del subsistema de E/S del SO.
 - ♦ En el peor de los casos volvemos a la situación anterior de los procesos Off line.
 - ♦ 3ª Generación (1965–1980)
 - ♦ Los fabricantes de ordenadores de principios de los 60 tenían dos tipos de sistemas informáticos, los que dirigían al comercio (de propósito general) y los de tipo científico. Tenían líneas incompatibles. Empiezan a aparecer el nuevo

término de escalabilidad (es decir son nuevos ordenadores con nuevas aplicaciones). IBM creó una familia llamada IBM 360 que se caracterizaban por tener una arquitectura compatible, y por ser multipropósito (tanto comercial como científico) y se adaptan a los nuevos cambios técnicos que aparecen (son capaces de evolucionar). En definitiva se estableció el concepto de Compatibilidad. Esta evolución pretendía que los SSOO fueran más complejos.

- ◆ **SPOOLING:** SPOOL (Simultaneous Peripheral Operation On Line) Aparece esta operación cuando se crearon los discos, van a sustituir a las cintas magnéticas ya que podrán leer información y enviar esta a la CPU simultáneamente. En el proceso de salida o escritura, si el CPU quiere escribir una línea en la impresora, primero va hacia un buffer y luego pasa al disco y de este a la impresora. (SPOOLING es idéntico al buffering pero si se supone al disco con buffer de almacenamiento ilimitado, y se utiliza para leer de los dispositivos de entrada tanto como sea posible y para escribir la información que se envía a los dispositivos de salida hasta que estos puedan procesarla.). Con el spooling es más beneficioso, ya que el CPU puede solapar la entrada salida de unos trabajos con otros. El almacenamiento de trabajos en el disco se llama **POOL** o depósito de trabajos.
- ◆ La **planificación de Trabajo** es el conjunto de técnicas que utiliza el SO para escoger un determinado trabajo entre varios trabajos candidatos que están esperando en un Pool.
- ◆ Otra técnica que se desarrolla con el Spooling es
- ◆ **La MULTIPROGRAMACION:** Es una técnica que apareció casi de forma natural, ya que con la planificación del trabajo se vio necesario la multiprogramación. Todos los procesos o técnicas de Off Line, buffering y spooling. Los usuarios no pueden tener ocupado al CPU y la salida simultáneamente. La multiprogramación incrementa la utilización de los trabajos de modo que la CPU tenga siempre algo que ejecutar, pero este proceso hace que los sistemas operativos sean más complicados. Todos los trabajos que han quedado suspendidos en el proceso de E/S se almacenan en memoria. Por lo que necesitarán una gestión de memoria.
- ◆ **TECNICAS DE TIEMPO COMPARTIDO (Time-Sharing)**
- ◆ Es lo que hoy se conoce como Multitarea. El tiempo compartido es una extensión lógica de la multiprogramación, en este caso el CPU ejecuta varios trabajos conmutando entre ellos. Esta conmutación es tan rápida que permite al usuario interactuar con un programa mientras se esta ejecutando. Los sistemas de Tiempo Compartido proporcionan al usuario la posibilidad de operar con el sistema en modo interactivo utilizando terminales externos (monitor + teclado). Si hay varios terminales conectados al CPU y solo unos cuantos están ejecutando algún programa, el tiempo asignado a cada terminal se les asigna exclusivamente a los que están

ejecutando algo.

- ◆ E/S Usuario 1
- ◆ CPU E/S Usuario 2
- ◆ E/S Usuario 3
- ◆ Ej. Cajeros automáticos, Cajas registradoras etc..
- ◆ **TECNICAS O SISTEMAS DE TIEMPO REAL**
- ◆ Los sistemas de tiempo real, diseñados para el control de procesos industriales, en estos sistemas se garantizan unos tiempos de respuesta máximos para cada una de las operaciones.
- ◆ 4ª Generación: (1980–1990)
- ◆ Está caracterizada por la aparición de los microprocesadores, gracias al diseño de circuitos VLSI (Very Large Scale Integration). Aparecen ordenadores de dos tipos: Personales y Estaciones de trabajo.
- ◆ En los ordenadores personales se utiliza como SO el MS–DOS y en las estaciones de trabajo el UNIX.
- ◆ **1.– Conectividad:** (Integración de redes) Conexiones de redes.
- ◆ La creación de conexiones entre ordenadores creando redes. Pero para el intercambio de información se utiliza la criptografía o codificación de la información para que no sea violada por otra sistema ajeno.
- ◆ **Los SO de red:** Los usuarios son conscientes de la existencia de otros ordenadores, y para obtener la información de estos, el usuario tiene que llamar directamente a la información de esos ordenadores.
- ◆ **Los SO distribuidos:** El usuario cree estar utilizando un SO tradicional que usa un solo procesador pero en realidad esta usando de forma transparente todos los recursos disponibles en la red. Es el propio sistema operativo el que se encarga de buscar la información pedida en la red, sin la interacción del usuario.
- ◆ Los SO de red necesitan unos protocolos establecidos para establecer la comunicación entre los ordenadores, además de unos módulos de bajo nivel.
- ◆ **2.– Máquina Virtual:** Lenguaje JAVA es un lenguaje que trabaja con máquinas virtuales. Son ordenadores conectados a red que utilizan recursos de la red como si fuesen del propio ordenador.
- ◆ **3.– DBMS:** Utilización de sistemas de gestión de bases de datos. Se distribuyen los datos en varias máquinas. Esto permite una segunda técnica llamada Replicación de datos y consiste en la duplicación de los datos, con lo que se una maquina se estropea, otra la puede sustituir. Esto se hace por seguridad y para poder repartir el trabajo.
- ◆ **PROTECCION DE LA INFORMACION Y SEGURIDAD**
- ◆ Existen cuatro niveles o mecanismos de protección y los presentamos en orden creciente de seguridad.
- ◆ **5.– No compartición:** Todos los procesos están totalmente aislados, no existe la posibilidad de que interfieran, esto se

puede conseguir con unos mecanismos denominados Memoria Virtual, hace que cada programa va a tener una memoria total virtual. Gestiona de forma exclusiva de todos los recursos. Si dos programas acceden al mismo archivo, para solucionar esto se copia el archivo a su parte de memoria, llamada memoria virtual que trabaja como si fuese única..

- ◆ **6.– Compartición de originales:** Con este caso se pueden compartir tanto programas como datos. Para la compartición de programas se ha diseñado un conjunto de códigos que hacen que un programa sea utilizado por varios procesos. En el caso de datos, se deja la copia original y se protege con un mecanismo que hace que con un espacio de memoria virtual parezca que lo tiene grabado, pero en realidad se llama directamente al original. En el momento en el que un proceso tenga que modificar la copia original, se deberá copiar el original a su espacio de dirección en la memoria virtual. Esto se llama (Copy on Write)
- ◆ **7.– Subsistemas confinados o sin memoria:** Se hace lo mismo que antes pero ahora el usuario es consciente del proceso. Se diseñan módulos independientes que necesitan servicios y este se lo va a ofrecer un módulo servidor. Este programa servidor solo se encarga de ofrecer los servicios ofrecidos por el cliente (es el modelo ofrecido en las WEB de Internet). Además un servidor puede ofrecer o atender simultáneamente los servicios de diversos clientes, por lo tanto el servidor utiliza un código de tipo entrante (Un programa que se ejecuta en distintos procesos)
- ◆ **8.– Diseminación controlada de la información.** Se utiliza un conjunto de técnicas que hace que un usuario de una aplicación se le asigne un código de acceso o credenciales que le permite la entrada a distintos servicios. Los datos y recursos están agrupados a niveles de clasificación haciendo que el usuario deba tener unas credenciales con el suficiente nivel para acceder y utilizar los datos o recursos.
- ◆ **ESTRUCTURA DE LOS SSOO.** (Libro)
- ◆ Una cola es una lista de procesos que está a la espera de que se le asigne un recurso, habrá tantas colas como tipos de recursos que se puedan compartir. Pero el – central son las **colas de corto plazo** son procesos que están ya en la memoria principal para su ejecución. La cola de corto plazo la controla un elemento llamado Dispatcher (es un módulo que toma la decisión de elegir un proceso:
- ◆ 1.– Utiliza una técnica llamada Round_Rolin (Turnos rotatorios) le asigna un tiempo corto del procesador a cada proceso y se van ejecutando rotatoriamente.
- ◆ 2.– La otra técnica es según los niveles de prioridad antes de entrar el proceso en la cola, el SO le asigna al proceso un nivel de prioridad (según su urgencia de ejecución o por lo que le dice el usuario). Además las dos técnicas se pueden combinar o utilizar a la vez. También, un primer nivel de prioridad puede aumentar por la finalización de los procesos

de un nivel superior a este.

- ♦ Las **Colas de largo plazo** son listas de trabajos que esperan poder usar el sistema (es similar a pool), esta cola normalmente van a estar en disco o en memoria virtual y estarán allí hasta que puedan ser ejecutados y pasen a la cola de corto plazo, este traslado lo realiza el SO. Para que sean trasladados hay que asignarle un espacio de memoria y un tiempo de proceso.
- ♦ Para que un programa pase de la cola de largo a corto plazo el SO evaluará el espacio de memoria y el tiempo de proceso que necesita el programa.
- ♦ Existe tantas colas de E/S según los dispositivos que haya de E/S. Va a utilizar las técnicas del dispatcher.
- ♦ El gestor de servicios es el módulo que se utiliza cuando algún programa o proceso necesita un servicio del SO.
- ♦ El gestor de interrupciones, gestiona las interrupciones cuando se solicita un proceso de E/S.
- ♦ **GESTION DE INTERRUPCIONES.**
- ♦ Una interrupción es un mecanismo que interrumpe el flujo de ejecución normal de un programa. Normalmente se trata de un mecanismo hardware. El SO hace consultas periódicas durante la ejecución de programas, se dirigen a los dispositivos E/S para ver si están ocupados. Pero se cambió por las interrupciones asíncronas ya que pueden llegar en cualquier momento, las de consultas son de tipo síncronas. Generalizando encontramos dos tipos de Interrupciones:
 - ♦ – Hardware: Indican algún tipo de evento o circunstancia físico. (Asíncronas).
 - ♦ Ej: La impresora se queda sin papel
 - ♦ – Software: Generados explícitamente por un programa. Los genera cuando quiero. (Síncronas). Son previsibles.
 - ♦ *Tipos de interrupciones:*
 - ♦ **1.– De programas:** (derivadas de las ejecuciones de programas) Algunos programas pueden estar mal programados (ej. La división por cero, desbordamiento numérico, instrucciones ilegales, violaciones del espacio de memoria (cuando un programa intenta ocupar un espacio de memoria de otro programa)).
 - ♦ **2.– De E/S:** Las generan los controladores de dispositivos. Avisan al SO si un dispositivo de E/S ha quedado libre.
 - ♦ **3.– De reloj o temporizador:** Todos los sistemas tienen un reloj interno que emite secuencialmente unos impulsos, utilizándose en operaciones de sincronización, principalmente se utiliza en la transmisión de datos, en la planificación de procesos.
 - ♦ **4.– Fallos a nivel hardware del sistema:** Por ejemplo en un error de paridad en la memoria, en un cambio brusco de tensión o de electricidad estática que modifica los datos, se usan unos datos llamados parity (paridad) que son la suma de los datos codificados y estos, si no coincide se interrumpe. Los programas que se encargan de revisar los fallos de hardware se guardan en la ROM ya que allí son inborrables.
 - ♦

♦ **Ejemplo de flujo de control.**

♦ Rutina de Servicio de interrupciones RSI y en el momento en el que haya una interrupción de E/S en la ejecución, se realiza dicho proceso de E/S y continúa la ejecución, en el momento en el que haya otra interrupción se activa la RSI que se encarga de identificar la causa de la interrupción y se ejecuta el código asociado a dicha causa. Si el tiempo necesario para la operación de E/S a nivel físico es menor que el tiempo necesario para procesar el siguiente bloque de información se dice que estamos en un escenario de espera corta y podemos hacer simultáneamente operaciones de E/S y el procesamiento del siguiente bloque. Si el tiempo de la operación de E/S es superior al tiempo físico de proceso del siguiente bloque, estamos en un escenario de espera larga o limitada por el tiempo de la operación de E/S.

♦ **Diagramas de tiempo.**

♦ 1.- ESPERA CORTA

♦ sin interrupciones

♦ con interrupciones

♦ 2.- ESPERA LARGA

♦ sin interrupciones

♦ con interrupciones

♦ **SECUENCIA DE INTERRUPCIONES**

♦ 1.- El dispositivo E/S produce una interrupción.

♦ 2.- Cuando el CPU recibe una señal de interrupción la CPU corta la instrucción en curso.

♦ 3.- Asentimiento, el CPU va hacer unas indicaciones para anular la señal de interrupción. Y se identifica la interrupción con el RSI (de que tipo y de donde viene)

♦ 4.- Prepara la transferencia a la RSI. Se guarda una estructura llamada PWS (Program State Word), que permite conocer el estado del procesador, almacenándose en una pila. Otra estructura importante es el PC (Program Counter) que me dice cual es la posición de memoria de la siguiente instrucción que hay que ejecutar. El PC se almacena en el PSW, en la pila.

♦ 5.- El PSW envia la dirección de memoria de la siguiente instrucción y se inicia el RSI.

♦ 6.- Se inicia la ejecución del RSI, y almacenan los valores de los distintos registros del procesador en la pila. Los registros contienen valores de programa que se está ejecutando antes de la interrupción.

♦ 7.- Se inicia el verdadero procesamiento de RSI.

♦ 8.- Se restauran de la pila todos los registros almacenados, si inicia la ejecución de la instrucción siguiente a la que quedó interrumpida en un principio.

♦ **SERVICIOS DEL SO**

♦ Servicios que tienen los Sistemas Operativos.

♦ 1.- Creación de programas, es decir, debe ser capaz de permitir la creación de programas utilizando editores, depuradores y utilidades que soportan el proceso de creación de programas.

- ◆ 2.- Ejecución de aplicaciones; debe gestionar las instrucciones y datos para poder llevarlas a la memoria para que sean ejecutadas. Debe gestionar todos los recursos de la memoria para poder almacenar las instrucciones y los datos.
- ◆ 3.- Acceso a dispositivos de entrada/salida: Suministrar unos módulos específicos que permiten gestionar cualquier operación de E/S
- ◆ 4.- Acceso controlado a dispositivos E/S; Se va hacer mediante una gestión física y otra lógica (como se almacena la información en los dispositivos físicos.)
- ◆ 5.- Acceso físico al sistema; se controla el acceso de los recursos por usuarios ajenos y se produce un control de entrada al sistema informático.
- ◆ 6.- Detección de fallos: Debe detectar cualquier fallo a nivel hardware o en la ejecución de una aplicación, es decir, a nivel de software. Por lo que debe identificar la causa.

◆ TEMA 2 Gestión de Archivos

◆ 1.- Introducción

- ◆ ◇ **Campo:** Un campo es el elemento de datos básico que maneja el usuario con un mismo tipo de datos. Se caracteriza por su longitud y el tipo longitud puede ser:
 - ◇ - fija
 - ◇ varios subcampos (nombre del campo, longitud del campo que viene a continuación y datos
 - ◇ - variable
 - ◇ inclusión de separadores especiales (indican que se ha terminado un dato) normalmente los SO no soportan los campos de longitud variable
 - ◇
 - **Registro:** Conjunto de campos relacionados que son considerados como una unidad por parte de los programas de aplicación (Campos de un determinado alumno)
 - fija (todos sus campos son de longitud fija)
 - Longitud
 - Al menos un campo es variable
 - Variable
 - Existe un número variable de campos (Con esta posibilidad los SO asignan un nombre
 - - **Bases de datos:** Es un conjunto de datos que están relacionados gracias a la utilización de varios archivos.
 - Los SO deben reconocer o identificar lo que son archivos de texto y los binarios (para evitar que los usuarios impriman un archivo binario)
 - - ◆ **Archivo:** Colección de registros similares (todas los registros son del mismo tipo)
 - ◆ Programa binario ejecutable

- ◆ Archivo
- ◆ Texto (Conjunto de caracteres)
- ◆ Datos
- ◆ (Usuario)
- ◆ Binarios (NO ejecutables)
- ◆ – Imagen (Datos de los puntos de la imagen)
- ◆ – Sonido (Muestras de la amplitud de un det. tiempo)
- ◆ Los archivos poseen un nombre y unas propiedades (tipo, fecha, dueño, longitud, ...) Existen tantos tipos de archivos como contenidos posibles. Los usuarios y aplicaciones accederán a estos archivos mediante el nombre del archivo. El SO hace referencia de otra forma mediante un identificador (MS DOS mediante un nº entero).
- ◆ Para dar permisos a los accesos en principio siempre se hace a nivel de archivo, los SO actuales ya lo hacen a nivel de registros.
- ◆ Los archivos se almacenan en términos de bloques, unidad de almacenamiento secundaria. Podemos entender un bloque como un lugar de almacenamiento físico. Normalmente todos los bloques son del mismo tamaño, y por ello entonces un bloque o registro físico contendrá varios registros lógicos.
- ◆ El numero de registros lógicos en un bloque; depende de:
 - ◆ – El tamaño del registro lógico.
 - ◆ – El tamaño del bloque físico.
 - ◆ – La Técnica de empaquetamiento.
- ◆ Un archivo para el SO es una secuencia de bloques

físicos. (Por ello, siempre se va a desperdiciar la última parte del último bloque) Es la denominada

Fragmentación Interna, sufrida por todos los SO.

- ◆ En cambio, la **Fragmentación Externa**, es cuando el archivo esta compuesto por bloques físicos no consecutivos, provocando con ello una perdida de tiempo.

- ◆ Elementos Fundamentales para organizar y localizar Archivos.

- ◆ 1.– **Directorio de**

Dispositivos: Es específico de cada unidad o dispositivo de almacenamiento. Se almacenan todas las propiedades físicas de los archivos.

- ◆ 2.– **Directorio de**

Archivos: Organización lógica de archivos, que se extiende a todos los dispositivos.

- ◆ **Operación con archivos**

- ◆ 1.– Creación: – Creación de espacio.

- ◆ – Entradas de directorio.

- ◆ 2.– Escritura: – Llamada al sistema (nombre u operación).

- ◆ – Localización en directorios.

- ◆ 3.– Lectura: – Llamada al sistema (Archivo o Posición de memoria).

- ◆ – Posición de lectura.

- ◆ 4.– Reinicialización – Poner la posición (de lectura) a 0.

- ◆ 5.– Eliminación – Liberar espacios.

- ◆ – Eliminar las entradas de directorio.

- ◆ **Hay dos métodos de acceso a archivos:**

- ◆ – Secuencial: Se procesa la información en orden. Se va leyendo un registro lógico

tras otro según su orden.

- ◆ – Directa: Un archivo es una secuencia numerada de bloques y no hay ninguna limitación en cuanto al orden de lectura o escritura en bloque. Es el que más se utiliza. Normalmente se asigna un número relativo a cada bloque.

- ◆
 - ◇ No todos los SO soportan los dos métodos. Pero teniendo uno de los métodos, se puede emular el otro.

- ◇ Directo Secuencial

- ◇ Secuencial Directo (muy ineficiente)

- ◇ **Soluciones de UNIX**

- ◇ Solo considera un tipo de archivo secuencia de bytes a la que no le asigna ningún tipo de interpretación (más flexibilidad, más problemas en la gestión de archivos)

- ◇ En todos los SO la gestión de archivos se realiza a nivel de bloques.

- ◇ Archivo siempre dividido en bloques, todos de igual longitud.

- ◇ En raras ocasiones coincide el tamaño del bloque con el tamaño de los registros lógicos. El SO compondrá los bloques para tratar la información.

- ◇ Bloque 1 Bloque 2 Bloque 3 (Seis registros)

- ◇ **SISTEMA DE GESTION DE ARCHIVOS**

◇ Son el conjunto de programas que van a modelar la utilización de archivos. Ofrecen a los usuarios y a sus aplicaciones una serie de servicios para la gestión de archivos. El SO no distingue a quien ofrece los servicios de la gestión de archivos. Se utiliza un interfaz (API) para los programas de aplicación, son un conjunto de funciones que van a servir para la gestión de archivos. El usuario accede a unas aplicaciones del SO y estas a la vez acceden al API para la gestión de archivos

◇ **Función de un sistema de ficheros**

- El primer objetivo es satisfacer las necesidades del usuario, se pretende almacenar información y poder realizar una gestión u operaciones sobre la información almacenada.
- La primera vez que se accede a un archivo, el SO opera unas estructuras

para
modificar el
archivo y
por tanto se
dice que el
archivo está
abierto y
cuando se
sale se dice
que el
archivo está
cerrado.

- Acceso controlado a los archivos, se establecen una serie de permisos para el acceso a los archivos.
- Intercambio de información entre archivos.
- Copias de seguridad. Se deben ofrecer mecanismos de copias de seguridad, almacenamiento en otros soportes físicos y recuperarlos luego en su estado original.
- El usuario debe ser capaz de acceder a un archivo de forma simbólica, es decir, con un identificador

del archivo
como el
nombre del
mismo.
Todos los
archivos
que están
abiertos
tienen un
código de
archivo
único, es
decir,
utilizan un
número
para
identificar
al archivo,
pero solo es
utilizado
por el SO.

· **Otras
funciones
secundarias**

- Además
un
sistema
de
archivos,
debe
de
ser
capaz
de
gestionar
todo
aquello
que
potencie
y
haga
mas
eficiente
la
gestión
de
archivos.
Tanto
para
el
usuario
como

- para el SO
- Debe ofrecer soporte de E/S, en distintos dispositivos.
- Soporte de múltiples usuarios, cuando varios usuarios intentan acceder al mismo archivo es SO debe elegir cual de los usuarios puede acceder al archivo.
- Todo Sistema de archivos debe minimizar la pérdida de información que se produzca y recuperar la mayor

parte
de
información
que
se
haya
perdido.

- **ARQUITECTURA DE UN SISTEMA DE ARCHIVO**
- **Arquitectura de Software de un sistema de archivos**

•
• 1.- Métodos de Acceso
• 2.-E/S Lógica
• 3.- Supervisor básico de E/S
• 4.- E/S Física (Sist. Arch. Basicos)
• 5.- DRIVERS

♦ **Drivers**
En el nivel inferior de esta forma por control de dispositivos (van a permitir operaciones)

de
E/S
espe
de
cada
disp
cada
uno
de
los
bloq
de
los
cont
de
disp
debe
ser
capa
de
inici
el
proc
de
E/S
y
proc
la
term
de
la
E/S)
Con
a
la
infor
com
una
sent
bina
♦ **Nive**
E/S
Físic
Inter
prim
con
el
exte
del
siste
infor
Aqu

se
emp
a
trata
la
infor
com
bloq
Deb
de
ser
capa
de
gesti
toda
la
mem
inter
que
se
está
utiliz
y
la
asig
de
la
mem
secu
de
los
bloq
de
los
arch
y
se
enca
de
gesti
buff
en
mem
♦ Las
dos
parte
suele
ser
parte
del
SO

