



Sistemas informáticos

Desarrollo de Aplicaciones Multiplataforma/Desarrollo de Aplicaciones
Web (DM1E y DA1D1E)

1º Presencial

Profesor: Juan Ignacio Benítez Palacios





Conceptos básicos sobre sistemas operativos

TEMA 2



Sistema operativo

- Constituye el **SW** más importante del sistema informático.
- Se denomina SW base.
- Sin él, el equipo HW son un conjunto de componentes sin utilidad ninguna.



Sistema operativo

- Se encarga de la gestión y asignación de recursos HW.
 - Recursos: procesador, memoria, periféricos y sistemas de archivos.
- Proporciona interfaz al usuario para que pueda usar el sistema con facilidad.



Componentes de un sistema operativo



- Núcleo
 - Parte del SO que interacciona con el HW.
 - Deja que las aplicaciones accedan a estos de forma segura.
 - Se encarga de la gestión de recursos del sistema: procesador, memoria, etc.



Componentes de un sistema operativo



- Núcleo
 - La gestión de los recursos la realiza en función de los servicios.
 - Si una aplicación, por ejemplo, necesita del uso de memoria llama al sistema para alertar del gestor de memoria.
 - El gestor localiza las zonas ocupadas y el núcleo contacta con el HW para llevar a cabo la acción.



Componentes de un sistema operativo



- Servicios
 - Cuando usuario ejecuta aplicación, ésta necesita de componentes HW.
 - Observamos la existencia de interfaz de comunicación entre aplicaciones y componente = gestor del elemento.
 - Distinguimos diferentes gestores de servicios según su funcionalidad: de memoria, de procesos, de E/S, etc.



Componentes de un sistema operativo



- Servicios
 - Gestor de memoria: se encarga de gestionar memoria principal (partes libres u ocupadas, asignar y liberar procesos, etc)
 - Gestor de procesos: encarga de gestionar procesador. Cuando hay varios en ejecución asigna qué proceso usa el procesador.
 - Gestor de E/S: encarga de gestionar dispositivos de E/S, periféricos.



Componentes de un sistema operativo



- Servicios
 - Gestor de archivos y directorios: encarga de gestionar almacenamiento secundario, gestión de ficheros y directorios.
 - Comunicación y sincronización entre procesos: mecanismos para que los procesos se comuniquen y sincronicen comunicación.
 - Seguridad: Conocer qué puede hacer un usuario que accede al sistema, para indicar al núcleo qué recursos puede usar.



Componentes de un sistema operativo



- Interfaz de llamadas al sistema
 - Interfaz que usan servicios para comunicarse con aplicaciones de nivel superior.
 - Estas aplicaciones pueden realizar peticiones a los gestores de servicios.



Componentes de un sistema operativo



- Interprete de comandos, órdenes o Shell
 - Proporciona al usuario interfaz por la que puede “hablar” con el PC.
 - El usuario dispone de comandos que ejecuta
 - El Shell interpreta dando información o aportando alguna operación concreta.



Estructura de sistema operativo

(Sistemas operativos según su estructura)



- Monolíticos:
 - Lo forma un programa donde se integran los componentes (núcleo, servicio y componentes).
 - Algunos de los sistemas operativos que conocemos comenzaron usando esta estructura.
 - Al adquirir popularidad requirieron nuevas funciones que se agregan al núcleo
 - Ejemplos: MS DOS, UNIX



Estructura de sistema operativo

(Sistemas operativos según su estructura)



- Estructurados
 - Siguen una estructura concreta.
 - Se organizan los componentes según la estructura.
 - Pueden ser: por capas y cliente-servidor



Estructura de sistema operativo

(Sistemas operativos según su estructura)



- SO estructurados por capas
 - Cada capa define una función del sistema
 - Cada capa ofrece servicios a la que está encima de ella mediante un interfaz de servicios.
 - Se dividen sus funciones
 - Cada parte es menos compleja



Estructura de sistema operativo

(Sistemas operativos según su estructura)



- SO estructurados por capas
 - Cada capa conoce las funciones que implementa
 - Y NO la que implementan las demás SÓLO conoce la interfaz, cabeceras de función que puede usar, no su código.



Estructura de sistema operativo

(Sistemas operativos según su estructura)



Estructura de sistema operativo

(Sistemas operativos según su estructura)



- SO cliente-servidor
 - Las funciones que tiene que implementar el SO se desarrollen mediante procesos de usuario.
 - Tiene micronúcleo y servidores (procesos que desarrollan la funcionalidad comentada antes)
 - Ejemplos: Windows XP. Vista, 7, etc.



Estructura de sistema operativo (Sistemas operativos según su estructura)



- SO cliente-servidor



Funciones de un sistema operativo



- Gestión del procesador
- Gestión de memoria
- Gestión de sistema de archivos
- Gestión de E/S
- Interfaz de usuario
- Interfaz para uso en aplicaciones



Funciones de un sistema operativo

- Gestión del procesador
 - Creación y finalización de procesos
 - Control de recursos o secciones críticas
 - Asignación y liberación de recursos críticos
 - Solucionar bloqueos

Funciones de un sistema operativo

- Gestión de memoria
 - Reserva y liberación de memoria
 - Conversión de direcciones virtuales
 - Comprobación del uso de memoria
 - Uso de memoria virtual



Funciones de un sistema operativo

- Gestión de sistema de archivos
 - Creación y eliminación de archivos y directorios
 - Modificación de ficheros y directorios
 - Asignación y manejo de permisos de archivos



Funciones de un sistema operativo

- Gestión de E/S
 - Coordinación de procesos a dispositivos de E/S
 - Manejo de memoria para acceso directo a dispositivos.
 - Proporcionar interfaz entre sistema y dispositivo
 - Proporcionar interfaz entre usuario y dispositivo

Funciones de un sistema operativo

- Interfaz de usuario
 - Proporcionar entorno para que el usuario se comuniquen con el SO.
 - Establecer comandos para realizar la comunicación.
 - Proporcionar interfaces gráficas de usuario intuitivas y fáciles de usar

Funciones de un sistema operativo



- Interfaz para uso de aplicaciones
 - Interfaz con funciones para ser usada en desarrollo de aplicaciones SW y que pueda hacer uso de recursos



Tipos de sistemas operativos

- 1ª etapa
 - El programador desarrolla el programa a mano y en código máquina.
 - Usa las tarjetas perforadas para introducirlo en el ordenador
 - Los resultados son impresos o grabados en cintas perforadas para ser usados
 - Si hay errores el programador realiza depuración, observando el estado de la memoria, registros, etc.



Tipos de sistemas operativos

- 1ª etapa
 - Los trabajos se ejecutan en serie: se introduce uno, se espera salida y después comenzaba el siguiente.



Tipos de sistemas operativos

- 2ª etapa
 - Aparecen sistemas operativos por Lotes.
 - El funcionamiento es similar al que hace el programador en la 1ª etapa.
 - La secuencia de trabajo es en serie
 - Se agrupan las tareas del mismo tipo formando lotes (batch)

Tipos de sistemas operativos

- 2ª etapa
 - El programador deja las tarjetas perforadas sobre la bandeja de entrada.
 - El SO las procesa en la salida de forma secuencial.

Tipos de sistemas operativos



- 3ª etapa:
 - Tipos de sistemas operativos: multiprogramación, tiempo compartido, tiempo real y de propósito general o multimodo.



Tipos de sistemas operativos

- 3ª etapa. Multiprogramación
 - Permitían mantener varios programas simultáneamente en memoria.
 - Se busca mejorar la productividad del SO
 - Se hace un mejor uso de los recursos



Tipos de sistemas operativos

- 3ª etapa. Tiempo compartido
 - Comparte sus recursos con todos los usuarios
 - Asigna a cada uno tiempos de memoria, de CPU, etc.
 - Forma de trabajar transparente al usuario: cree que usa los recursos sin compartirlos con nadie
 - Manera de trabajar habitual



Tipos de sistemas operativos

- 3ª etapa. SO de tiempo real
 - Son contruidos para tareas específicas
 - Se instalan en sistemas que procesan muchos eventos.
 - Tratan de proporcionar rápidos tiempos de respuesta ante sucesos exteriores
 - Uso: control de tráfico aéreo, de trenes, etc.

Tipos de sistemas operativos

- 3ª etapa. SO de propósito general o multimodo
 - SO capaces de procesar por lotes, en multiprogramación, tiempo real o tiempo compartido.
 - SW más complejo
 - Programadores o usuarios del sistema deban aprender lenguaje de control complicado para preparar sus trabajos.

Tipos de sistemas operativos

- 4ª etapa.
 - Se centra más en el uso del sistema informático que en el rendimiento de la máquina.
 - Conseguir SO fáciles de usar con menor rendimiento
 - Surge el concepto red de ordenadores
 - También los sistemas virtualizados o el concepto de máquina virtual.

Tipos de sistemas operativos

- 4ª etapa. SO orientados a usuarios finales
 - SO que sin dejar de realizar su función son accesibles por todos.
 - No es necesario el estudio de comandos complejos para su uso.
 - Ejemplo: Windows 10 o Ubuntu

Tipos de sistemas operativos

- 4ª etapa. SO distribuidos
 - Común a varios equipos
 - Mantienen comunicación a través de la red
 - Cuando un usuario accede no sabe donde están almacenados los ficheros o donde se procesan los datos.
 - Ejemplo: forma la red de sucursales de un banco.

Tipos de sistemas operativos

- 4ª etapa. SO middleware.
 - Evolución de sistemas operativos distribuidos
 - Middleware: capa de software que se ejecuta sobre equipo que tiene SO propio encargada de gestionar sistema distribuido
 - En SO middleware los equipos disponen de su propio HW y su SO, agregando SW que hace las veces de SO distribuido.

Tipos de sistemas operativos

- 4ª etapa. SO middleware.
 - En los últimos años ha surgido concepto middleware de virtualización.
 - Con la virtualización tenemos en nuestro PC varios SO huésped.
 - El SW de virtualización gestiona recursos en todas las máquinas virtuales configuradas.

Tipos de sistemas operativos



- Según su uso.
 - El número de usuarios que utilicen el sistema → MONOUSUARIO o MULTIUSUARIO.
 - El número de tareas que se puedan realizar al mismo tiempo → MONOTAREA o MULTITAREA.



Sistemas operativos comerciales

- Windows 7
- Windows 8.
- Windows 10



UNIX y GNU/LINUX



- UNIX es un SO de multiproceso, multiprogramación y multiusuario.
- Desarrollado en 1969 por Thompson, Ritchie y McIlroy
- En 1991 Linus Torvalds estudiante desarrolla el SO Linux basado en Minix. Parecido a UNIX de código abierto.



SO propietarios y libres

- SO = una aplicación que se puede catalogar como propietario o libre.
- Las características que distingue a un SW libre son:
 - Libertad 0: el programa se ejecuta siempre que quiera el usuario y con los propósitos que quiera.
 - Libertad 1: Acceso al código fuente
 - Libertad 2: Libertad de distribuir copias del SW
 - Libertad 3: Posibilidad de mejorar el SW y compartir mejoras.

SO propietarios y libres



- Tipos de licencias del SW
 - Licencia: Contrato desarrollador SW y usuario que establecen los derechos y deberes de ambos.
 - Patente: Derechos garantizados por un gobierno al inventor de un nuevo producto.
 - Derecho de autor o copyright: forma en que las leyes protegen las obras originales.



SO propietarios y libres



- Licencia de SW libre: GNU General Public License
 - Licencia creada por Free SW Foundation en 1989.
 - Está orientada a la protección de las 4 libertades del SW libre
 - Licencia GPL debe preservar que SW sea libre protegiendo de intentos de apropiación que hagan perder libertades



SO propietarios y libres



- Licencia de SW propietario
 - Al adquirir un SW propietario hay que estar protegidos por derechos de autor y quien lo compre debe registrarse a ellos.
 - La única forma de asegurar que se cumplen los derechos de autor es con el uso de licencias



SO propietarios y libres



- Licencias:
 - CLUF: Licencia de código cerrado en las que el autor del SW es quien decide qué se puede hacer con él.
 - OEM: Licencias al adquirir equipos con una versión del sistema operativo o SW preinstalado.
 - Licencia de producto empaquetado: Adquisición del producto original en caja donde aparece el contrato de licencia.



SO propietarios y libres



- Licencias:
 - Por volúmenes: Licencias para empresas con un número de PC's a las que se facilitan la compra de licencias con descuento.
 - Licencias educativas y para otros sectores: Según el SW se pueden configurar licencias ajustadas a las necesidades.



Algoritmos de planificación del procesador



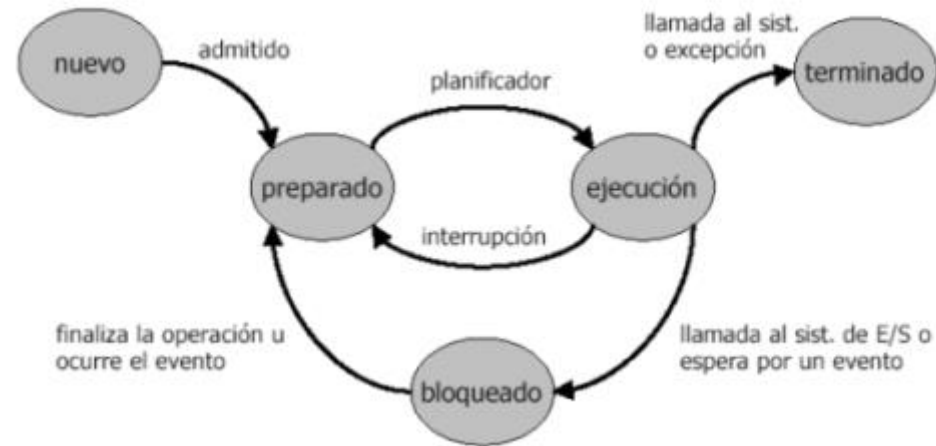
- Se encargan de asegurar que un proceso no monopoliza el procesador.
- Un proceso es un programa que está en ejecución.
- Este proceso puede estar en 3 estados distintos “Listo” “Bloqueado” y “En Ejecución”.



Algoritmos de planificación del procesador



- Los procesos son almacenados en una lista junto con la información que indica en qué estado está el proceso, el tiempo que ha usado el CPU, etc.



Algoritmos de planificación del procesador. FIFO.



- El procesador ejecuta cada proceso hasta que termina.
- Los procesos que en cola de procesos preparados permanecerán encolados en el orden que lleguen hasta que les toque su ejecución.
- Este método se conoce también como FIFO (First input, First output, primero en llegar primero en salir).



Algoritmos de planificación del procesador. FIFO.



- Política muy simple y sencilla de llevar a la práctica, pero muy pobre en cuanto a su comportamiento.
- La cantidad de tiempo de espera de cada proceso depende de:
 - Número de procesos que se encuentren en la cola en el momento de su petición de ejecución
 - Y del tiempo que cada uno de ellos tenga en uso al procesador
 - Y es independiente de las necesidades del propio proceso.



Algoritmos de planificación del procesador. FIFO.



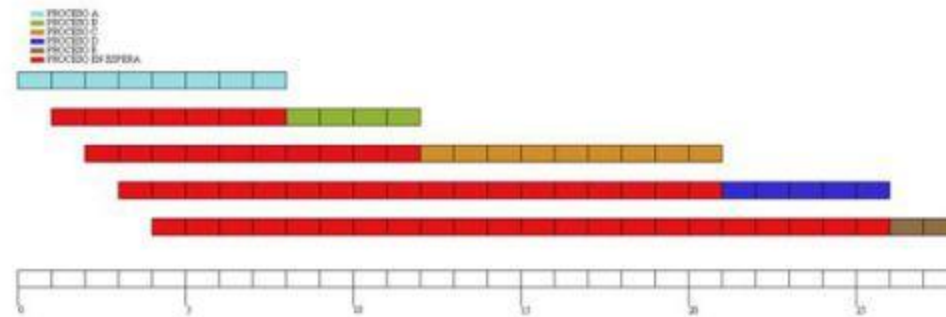
- Sus características son:
 - No apropiativa.
 - Es justa, aunque los procesos largos hacen esperar mucho a los cortos.
 - Predecible.
 - El tiempo medio de servicio es muy variable en función del número de procesos y su duración.



Algoritmos de planificación del procesador. FIFO.



Proceso	Tiempo de ejecución	Tiempo de llegada	Tiempo de comienzo	Tiempo de finalización	Tiempo de retorno	Tiempo de espera
A	8	0	0	8	8	0
B	4	1	9	12	$12 - 1 = 11$	$11 - 4 = 7$
C	9	2	13	21	$21 - 2 = 19$	$19 - 9 = 10$
D	5	3	21	26	$26 - 3 = 23$	$23 - 5 = 18$
E	2	4	26	28	$28 - 4 = 24$	$24 - 2 = 22$



Algoritmos de planificación del procesador. FIFO.



- En el caso de que los procesos de mayor tiempo de duración llegasen los primeros, el tiempo medio de espera sería mucho mayor.
- Podemos llegar a la conclusión de que este no es un algoritmo eficiente.



Algoritmos de planificación del procesador. SJF



- En este algoritmo, da bastante prioridad a los procesos más cortos a la hora de ejecución y los coloca en la cola.
- Ejemplo:
 - Una cola de personas en Mercadona delante de la caja , la persona que menos compra lleva esa pasa primero.



Algoritmos de planificación del procesador. SJF



- Selecciona al proceso con el próximo tiempo ejecución más corto.
- El proceso corto saltará a la cabeza de la cola.
- Ejecución de un proceso consiste en ciclos de ejecución de CP y ciclos de espera por E/S.



Algoritmos de planificación del procesador. SJF



- El algoritmo selecciona aquel proceso cuyo próximo ciclo de ejecución de CP sea menor.
- El problema está en conocer dichos valores, pero podemos predecirlos usando la información de los ciclos anteriores ejecutados.

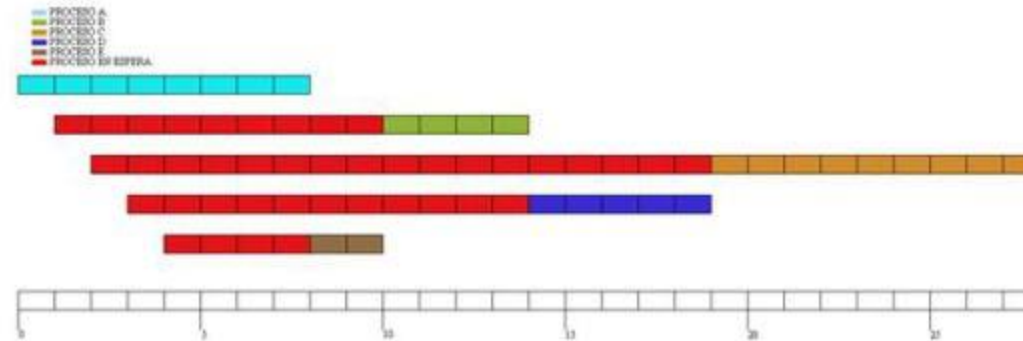


Algoritmos de planificación del procesador. SJF



SJF

Proceso	Tiempo de ejecución	Tiempo de llegada	Tiempo de comienzo	Tiempo de finalización	Tiempo de retorno	Tiempo de espera
A	8	0	0	8	8	0
B	4	1	10	14	$14 - 1 = 13$	$13 - 4 = 9$
C	9	2	19	28	$28 - 2 = 26$	$26 - 9 = 17$
D	5	3	14	19	$19 - 3 = 16$	$16 - 5 = 11$
E	2	4	8	10	$10 - 4 = 6$	$6 - 2 = 4$



Algoritmos de planificación del procesador. SRTF



- Con la diferencia de que si un nuevo proceso pasa a listo se activa para ver si es más corto que lo que queda por ejecutar del proceso en ejecución.
- Si es así, el proceso en ejecución pasa a listo y su tiempo de estimación se decrementa con el tiempo que ha estado ejecutándose.



Algoritmos de planificación del procesador. SRTF



- Los procesos llegan a la cola y solicitan un intervalo de CPU.
- Si dicho intervalo es inferior al que le falta al proceso en ejecución para abandonar la CPU, el nuevo proceso pasa a la CPU y el que se ejecutaba a la cola de preparados.

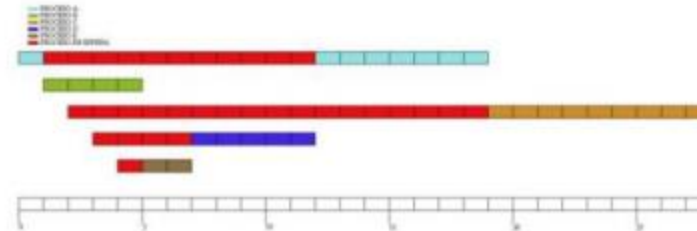


Algoritmos de planificación del procesador. SRTF



SRTF

Proceso	Tiempo de ejecución	Tiempo de llegada	Tiempo de comienzo	Tiempo de finalización	Tiempo de retorno	Tiempo de espera
A	8	0	0-12	1-19	19	$19 - 8 = 11$
B	4	1	1	5	$5 - 1 = 4$	$4 - 4 = 0$
C	9	2	19	28	$28 - 2 = 26$	$26 - 9 = 17$
D	5	3	7	12	$12 - 3 = 9$	$9 - 5 = 4$
E	2	4	5	7	$7 - 4 = 3$	$3 - 2 = 1$



Algoritmos de planificación del procesador. SRTF



- El intervalo de CPU es difícil de predecir
- Posibilidad de inanición: los trabajos largos no se ejecutarán mientras hayan trabajos cortos.



Algoritmos de planificación del procesador. Round Robin



- Es un método para seleccionar todos los elementos en un grupo de manera equitativa y en un orden racional
- Normalmente comenzando por el primer elemento de la lista hasta llegar al último y empezando de nuevo desde el primer elemento.



Algoritmos de planificación del procesador. Round Robin



- Round Robin es uno de los algoritmos de planificación de procesos más complejos y difíciles
- Asigna a cada proceso una porción de tiempo equitativa y ordenada, tratando a todos los procesos con la misma prioridad.
- Se define un intervalo de tiempo denominado cuanto, cuya duración varía según el sistema.



Algoritmos de planificación del procesador. Round Robin



- La cola de procesos se estructura como una cola circular.
- El planificador la recorre asignando un cuanto de tiempo a cada proceso.
- La organización de la cola es FIFO.

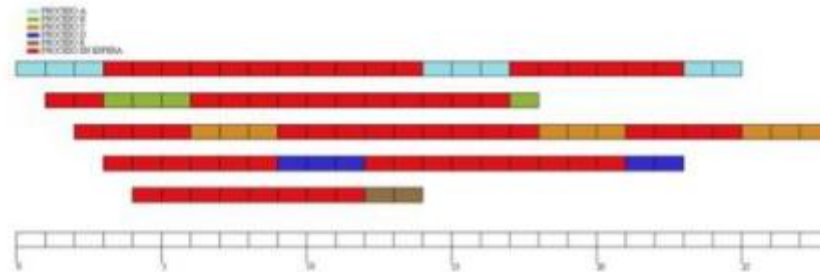


Algoritmos de planificación del procesador. Round Robin



Robin

Proceso	Tiempo de ejecución	Tiempo de llegada	Tiempo de comienzo	Tiempo de finalización	Tiempo de retorno	Tiempo de espera
A	8	0	0-14 -23	3 -17 -25	25	$25 - 8 = 17$
B	4	1	3 -17	6 -18	$18 - 1 = 17$	$17 - 4 = 13$
C	9	2	6 -18 -25	9 -21 -28	$28 - 2 = 26$	$26 - 9 = 17$
D	5	3	9 -21	12 -23	$23 - 3 = 20$	$20 - 5 = 15$
E	2	4	12	14	$14 - 4 = 4$	$10 - 2 = 2$



Algoritmos de planificación del procesador. Round Robin



- Cada proceso tiene asignado un intervalo de tiempo de ejecución, llamado quantum o cuanto.
- Si el proceso agota su quantum de tiempo, se elige a otro proceso para ocupar la CPU.
- Si el proceso se bloquea o termina antes de agotar su quantum también se alterna el uso de la CPU.



Algoritmos de planificación del procesador. Round Robin



- Round Robín es muy fácil de implementar.
- Todo lo que necesita el planificado es mantener una lista de los procesos listos.
- Es el mas fiable ya que cada proceso se le asigna un tiempo de ejecución si se agota elige otro proceso



Otros algoritmos. Planificación de colas múltiples. MQS



- La cola de procesos que se encuentran en estado de listos es dividida en un número determinado de colas más pequeñas.
- Los procesos son clasificados mediante criterio para determinar en qué cola será colocado cada uno cuando quede en estado de listo.
- Cada cola puede manejar un algoritmo de planificación diferente a las demás.

