

***SIMULACIÓN DE LOS MECANISMOS DE ADMINISTRACIÓN Y CONTROL  
DE UN SISTEMA OPERATIVO PARTE I***

***YENIS ÁLVAREZ JIMÉNEZ***

***YASMÍN MOYA VILLA***

***CORPORACIÓN UNIVERSITARIA TECNOLÓGICA DE BOLÍVAR***

***FACULTAD DE INGENIERÍA DE SISTEMAS***

***CARTAGENA DE INDIAS D.T y C***

***2000***

***SIMULACIÓN DE LOS MECANISMOS DE ADMINISTRACIÓN Y CONTROL  
DE UN SISTEMA OPERATIVO PARTE I***

***YENIS ÁLVAREZ JIMÉNEZ***

***YASMÍN MOYA VILLA***

**Trabajo de Grado presentado como requisito parcial para optar el  
título de Ingenieros de Sistemas**

***Director:  
JUAN MARTÍNEZ LAMBRAÑO  
Magíster en Ciencias Computacionales***

***CORPORACIÓN UNIVERSITARIA TECNOLÓGICA DE BOLÍVAR***

***FACULTAD DE INGENIERÍA DE SISTEMAS***

***CARTAGENA DE INDIAS D.T. y C.***

***2000***

Cartagena de Indias D.T. y C., Abril 24 de 1 2000

Señores

**CORPORACIÓN UNIVERSITARIA TECNOLÓGICA DE BOLIVAR**

Atn: Comité de Evaluación de Proyectos

Facultad de Ingeniería de Sistemas

La Ciudad.

Respetados señores:

Por medio de la presente me permito someter a estudio y aprobación el trabajo de grado titulado "**SIMULACIÓN DE LOS MECANISMOS DE ADMINISTRACIÓN Y CONTROL DE UN SISTEMA OPERATIVO PARTE I**" realizado por los estudiantes Yenis Álvarez Jiménez y Yasmín Moya Villa, quienes lo presentan a ustedes para optar al título de Ingeniero de Sistemas.

Cordialmente,

---

**JUAN MARTÍNEZ LAMBRAÑO**

Director del proyecto

Cartagena de Indias D.T. y C., Abril 24 de 1 2000

Señores

**CORPORACIÓN UNIVERSITARIA TECNOLÓGICA DE BOLIVAR**

Atn: Comité de Evaluación de Proyectos

Facultad de Ingeniería de Sistemas

La Ciudad.

Respetados señores:

Por medio de la presente nos permitimos hacer entrega formal del trabajo de grado titulado **"SIMULACIÓN DE LOS MECANISMOS DE ADMINISTRACIÓN Y CONTROL DE UN SISTEMA OPERATIVO PARTE I"**, como requisito parcial para optar al título de Ingeniero de Sistemas.

Atentamente,

---

**YENIS ÁLVAREZ JIMÉNEZ**

---

**YASMÍN MOYA VILLA**

Cartagena de Indias D. T. Y C., Abril 24 del 2000

Ingeniero

**GONZALO GARZÓN**

Decano de la Facultad de Ingeniería de Sistemas.

Corporación Universitaria Tecnológica de Bolívar.

Respetado Ingeniero:

Por medio de la presente le hacemos entrega formal del proyecto de grado titulado "**SIMULACIÓN DE LOS MECANISMOS DE ADMINISTRACIÓN Y CONTROL DE UN SISTEMA OPERATIVO PARTE I**", para su aprobación.

Cordialmente,

---

**YENIS ÁLVAREZ JIMÉNEZ**

---

**YASMÍN MOYA VILLA**

## **ARTÍCULO 105**

La Corporación Universitaria Tecnológica de Bolívar se reserva el derecho de propiedad intelectual de todos los trabajos de grado aprobados y no pueden ser explotados comercialmente sin su autorización.

**Nota de aceptación**

---

---

---

---

**Presidente del Jurado**

---

**Jurado**

---

**Jurado**

**Ciudad y fecha** (día,mes,año)



## **AGRADECIMIENTOS**

La culminación de este proyecto trae consigo la satisfacción de haber alcanzado los objetivos propuestos, es mirar atrás y saber que los momentos difíciles tenían su razón de ser pues no eran motivo para caer sino retos para seguir adelante con el enriquecimiento del conocimiento y experiencias que ahora permiten o hacen posible la entrega de este trabajo. Por todo esto, expresamos nuestros agradecimientos:

A nuestros padres Edelfa Jiménez, Rafael Álvarez, Nubia Villa y Mario Moya por su fe en nuestras capacidades y por mantener siempre viva la llama de la esperanza en nuestro corazón y pensamiento.

A nuestro director Juan Martínez por su apoyo constante e incondicional.

A los profesores Moisés Quintana y Gonzalo Garzón por toda la colaboración profesional y humana que siempre nos brindaron.

Y a todas aquellas personas que de una u otra forma colaboraron para que la culminación de este trabajo de grado se llevara a cabo.

Gracias.  
*Que el señor los bendiga.*

## CONTENIDO

	Pág.
<b>INTRODUCCIÓN</b>	1
<b>1. MARCO TEÓRICO</b>	3
1.1 ASIGNACIÓN DE MEMORIA	3
1.1.1 Mecanismo de asignación contigua simple	4
1.1.2 Mecanismo de asignación por particiones estáticas	6
1.1.3 Mecanismo de asignación por particiones dinámicas	9
1.1.4 Compactación	13
1.1.5 Mecanismo de asignación por paginamiento puro	14
1.1.6 Mecanismo de asignación por paginamiento con demanda	19
1.1.6.1 Algoritmo de reemplazo FIFO	24
1.1.6.2 Algoritmo de reemplazo LRU	25
1.1.7 Mecanismo de asignación por segmentación	25
1.1.8 Mecanismo de asignación por segmentación y paginamiento con demanda	26
1.2 ALGUNAS GENERALIDADES DE ENTRADA / SALIDA	26
1.2.1 Dispositivos de Entrada / salida	27
1.2.2 Interfaz Procesador-Periférico	28

1.3 SISTEMA DE ARCHIVOS	28
1.3.1 Sistema de Archivos Planos	29
1.3.2 Sistema de Archivos Jerárquicos	30
1.3.3 Implantación de un Sistema de Archivos	31
1.3.4 Administración de archivos en S.O DOS	31
1.3.5 Administración de archivos en S.O UNIX	32
1.4 ADMINISTRACIÓN DEL DISCO	35
1.4.1 Algoritmos para atención de peticiones	36
1.4.1.1 Primero en llegar, primero en acceder	37
1.4.1.2 Primero el de menor tiempo de búsqueda	37
1.4.1.3 Exploración	38
1.4.1.4 Exploración circular	39
<b>2. EL PROBLEMA Y OBJETIVOS DE LA INVESTIGACIÓN</b>	<b>40</b>
2.1 ANTECEDENTES DEL PROBLEMA	40
2.2 IDENTIFICACIÓN DEL PROBLEMA	41
2.3 PLANTEAMIENTO DEL PROBLEMA	41
2.4 ANÁLISIS DEL PROBLEMA	42
2.5 FORMULACIÓN DEL PROBLEMA	42
2.6 OBJETIVOS	43
2.6.1 Objetivo general	43
2.6.2 Objetivos específicos	43
2.7 RESULTADOS DE LA INVESTIGACIÓN	44
<b>3. ENTORNO DEL DISEÑO DEL SOFTWARE</b>	<b>45</b>

3.1 INTRODUCCIÓN	45
3.2 ACERCA DEL LENGUAJE	46
3.3 DESTINATARIOS	48
3.4 PRESENTACIÓN	48
3.5 ÁREAS DEL CONTENIDO	49
3.6 COMPLEJIDAD	49
3.7 CONTROL INSTRUCCIONAL	50
3.8 AYUDAS	51
<b>4. DESCRIPCIÓN DE ASPECTOS DE EJECUCIÓN DEL SOFTWARE</b>	<b>52</b>
4.1 ASPECTOS GENERALES	52
4.2 ASPECTOS PARTICULARES	53
4.2.1 Asignación de memoria	53
4.2.2 Algunas generalidades de Entrada/Salida	54
4.2.3 Sistema de archivos	54
4.2.4 Administración del disco	55
4.3 MENSAJES EN LAS SIMULACIONES	56
<b>5. DESCRIPCIÓN DE ASPECTOS DEL DISEÑO DEL SOFTWARE</b>	<b>58</b>
5.1 ASPECTOS GENERALES	58
5.2 ASPECTOS PARTICULARES	59
5.2.1 Asignación de memoria	59
5.2.1.1 Estructuras de datos relevantes	62
5.2.1.2 Supuestos	79
5.2.2 Algunas generalidades de Entrada / salida	80

5.2.2.1 Estructuras de datos relevantes	81
5.2.2.2 Supuestos	82
5.2.3 Sistema de archivos	83
5.2.3.1 Estructuras de datos relevantes	83
5.2.3.2 Supuestos	87
5.2.4 Administración del disco	87
5.2.4.1 Estructuras de datos relevantes	88
5.2.4.2 Supuestos	88

## **CONCLUSIONES**

## **RECOMENDACIONES**

## **BIBLIOGRAFÍA**

## LISTA DE FIGURAS

	<b>Pág.</b>
<b>Figura 1.</b> Contigua simple	4
<b>Figura 2.</b> Particiones estáticas	6
<b>Figura 3.</b> Particiones Libres adyacentes	13
<b>Figura 4.</b> Paginamiento puro	15
<b>Figura 5.</b> Sistema de archivos Plano	28
<b>Figura 6.</b> Sistema de archivos Jerárquico	28
<b>Figura 7.</b> Representación de un í-nodo	33
<b>Figura 8.</b> Tiempos de acceso a disco	34
<b>Figura 9.</b> Planificación del disco FCFS	36
<b>Figura 10.</b> Planificación del disco SSTF	37
<b>Figura 11.</b> Planificación del disco SCAN	38
<b>Figura 12.</b> Planificación del disco C-SCAN	38
<b>Figura 13.</b> Diseño General del Software	56
<b>Figura 14.</b> Proceso de Entrada / salida	80

## GLOSARIO

**ADMINISTRADOR DE MEMORIA:** Modulo o parte del sistema operativo que determina que parte de la memoria asignar a los procesos y que parte de los mismos debe permanecer en disco o cuales partes deben abandonar temporalmente la memoria para ceder su espacio a otros que lo requieran en un momento determinado.

**BLOQUE:** División física del espacio de direcciones reales.

**DOS:** Sistema operativo monousuario y monotarea también conocido como MS-DOS (Microsoft-Disk Operating System) lanzado por la IBM en 1981

**DIRECCIÓN VIRTUAL:** Dirección en memoria virtual o en disco de un dato.

**DISPOSITIVO DE BLOQUE:** Tipo de dispositivo de entrada/salida que almacena la información en bloques de tamaño fijo, cada uno con su propia dirección.

**DISPOSITIVO DE CARÁCTER:** Tipo de dispositivo de entrada / salida que envía o recibe un flujo de caracteres, sin sujetarse a una estructura de bloque, no puede utilizar direcciones ni tiene una operación de búsqueda.

**DRIVER:** Conjunto de rutinas y tablas que, ejecutan y controlan todas las operaciones de entrada y salida de cualquier periférico conectado a la computadora, siendo particulares para cada dispositivo.

**ENTORNOS BATCH:** Consiste en la automatización del procesamiento de un grupo de trabajos que comparten características comunes.

**FALLO DE PÁGINA:** Ausencia de una página solicitada en memoria.

**FIFO:** (First in, First Out, Primero en llegar, Primero en salir) Algoritmo de reemplazo de página que selecciona para su sustitución la página que lleva más tiempo en memoria

**Í-NODO:** Descripción interna de un archivo y de sus bloques de datos. Contiene la siguiente información de un archivo: el tipo de archivo, el identificador del usuario al cual pertenece el archivo, el identificador del grupo al cual pertenece el archivo, el tamaño del archivo, el tiempo de su último acceso y modificación.

**LOG:** Archivo que recopila todos los acontecimientos ocurridos en un sistema o programa en un tiempo dado.

**LRU:** (Least Recently Used, Menos usada recientemente) Algoritmo de reemplazo de página que selecciona para su sustitución la página que menos se ha referenciado, suponiendo que es la que tiene menos probabilidad de volver a ser referenciada.

**MEMORIA VIRTUAL:** Método que consiste en destinar parte del disco para almacenar aquellas partes de los programas que no se estén utilizando de manera que parezca que se excede la cantidad de memoria física disponible.

**MONOPROGRAMACIÓN:** Esquema de administración de memoria en el cual solo se tiene un proceso en memoria en cada instante.

**MULTIPROGRAMACIÓN:** Consiste en tener varios procesos en memoria al tiempo de forma que cada uno de ellos recibe un tiempo de procesamiento.

**MAPA DE MEMORIA:** Representación gráfica de la memoria.

**PÁGINA:** División lógica de un proceso en tamaños iguales.

**PROCESO SOLICITANTE:** Trabajo que requiere espacio en memoria para su ejecución.

**SEGMENTO:** Módulo independiente de un proceso. Es un componente lógico de tamaño variable de un proceso.

**SISTEMA OPERATIVO:** Conjunto de programas y funciones que ocultan los detalles de hardware, ofreciendo al usuario una vía sencilla y flexible de acceso al mismo. A la vez que administra los recursos ofrecidos por el hardware para alcanzar un eficaz rendimiento de los mismos.

**SWAPPING:** Trasladar una página de disco a memoria.

**UNIX:** Sistema operativo multiusuario desarrollado en sus inicios por las empresas Bell Telephone Laboratories y General Electric Company del que hoy existen varias versiones producidas por diferentes fabricantes.

**VISOR DE LOG:** Herramienta que permite ver el contenido de un log.

**V-TOC:** Etiqueta del disco que se encuentra localizada en el primer sector del mismo y que contiene la tabla de particiones del disco.

## RESUMEN

El aprovechamiento de los recursos de una máquina depende en gran parte del Sistema operativo, siendo este el pilar de todos los programas y hardware del sistema, controla las operaciones de la computadora, a la vez que proporciona a los usuarios una interfaz que hace transparente las operaciones de bajo nivel.

Este proyecto tiene como fin la simulación de los mecanismos de administración y control de un sistema operativo, de los cuales hacen parte: asignación de memoria, algunas generalidades de entrada/salida, sistema de archivos y administración del disco. El rendimiento del sistema está sujeto a la forma en que el sistema operativo manipula dichos mecanismos.

Siendo la memoria uno de los recursos más importantes, el Sistema Operativo debe ser cuidadoso en su administración, dicha tarea se lleva a cabo a través del **administrador de memoria** que es quien determina que parte de la memoria asignar a los procesos y que parte de los mismos debe permanecer en disco (en caso de que el mecanismo empleado lo permita). Los mecanismos estudiados son: 1) **Mecanismo de asignación contigua simple**. Divide la memoria en dos áreas contiguas. Una de ellas está asignada permanentemente a la parte residente del sistema operativo. La memoria restante se asigna a los procesos de usuario los cuales son alojados de tal forma que hasta que no termine uno su trabajo no cederá el control al siguiente. 2) **Mecanismo de asignación por particiones estáticas**. Este modo de asignación consiste en particionar la memoria en tamaños fijos y contiguos, en los cuales se asignan los procesos. Los tamaños de las particiones son definidos al inicializar el sistema y permanecerán estáticos hasta que se apague el equipo. 3) **Mecanismo de asignación por particiones dinámicas**. Asigna dinámicamente la memoria a los trabajos de acuerdo con su tamaño. 4) **Compactación**. Técnica que permite recolocar las zonas de memoria utilizadas para lograr un único espacio de memoria disponible. 5) **Mecanismo de asignación por paginamiento puro**. La paginación es una técnica de gestión que permite asignar la memoria de forma discontinua para un mejor aprovechamiento de la misma. Con este fin se divide el proceso en Páginas

y la memoria en partes de tamaño fijo llamados bloques, los cuales son de igual tamaño para alojar las páginas de los procesos. 6) **Mecanismo de asignación por paginamiento con demanda.** Esta técnica permite asignar memoria en forma dinámica; a medida que llegan los procesos, sus páginas son alojadas en disco y son cargadas en la memoria en el momento que son referenciadas, en caso de que la página referenciada no se encuentre en memoria es necesario remover alguna,

Dos de las políticas empleadas para elegir la página a remover son Algoritmo de reemplazo FIFO (Primero en entrar primero en salir) y Algoritmo de reemplazo LRU (La de menor uso reciente). 7) **Mecanismo de asignación por segmentación.** Un segmento de memoria constituye un espacio independiente de direcciones, estos pueden crecer o reducirse sin afectar a los demás. Un segmento puede contener un procedimiento, un arreglo una pila, o bien un conjunto de variables escalares. 8) **Mecanismo de asignación por segmentación y paginamiento con demanda.** Con el paginamiento y la segmentación en forma independiente se tienen ventajas e inconvenientes, es decir, si los segmentos son muy grandes podría ser inconveniente e incluso imposible mantenerlos en la memoria en su totalidad, esto hace que surja la idea de paginarlos de forma que sólo permanezcan en memoria las páginas que sean necesarias.

En algunos mecanismos de asignación de memoria parte de los procesos permanece en disco, pero no solamente los procesos hacen uso de este fundamental recurso; pues, también es accedido por los usuarios con mucha frecuencia; razón por la cual es necesario una eficiente **Administración del disco**, la cual tiene entre sus objetivos minimizar los tiempos de acceso al disco. Se han propuesto varios algoritmos para atender a las peticiones de disco. 1) **Primero en llegar, primero en acceder (FCFS):** Según este criterio se atienden los accesos en su orden de llegada. 2) **Primero el de menor tiempo de búsqueda:** Su método es atender primero la petición más cercana a la última servida. 3) **Exploración (SCAN):** La exploración establece un recorrido predeterminado del brazo del disco que va del primer al último cilindro y vuelta. En este recorrido irá atendiendo las peticiones más cercanas según el sentido del desplazamiento. 4) **Exploración circular (C-SCAN):** Este algoritmo es una variación del anterior, en este caso el movimiento del brazo sigue siendo el mismo, pero solo atiende peticiones en uno de los dos sentidos. Al llegar al último cilindro regresará al principio sin atender petición alguna.

El disco uno de los dispositivos de entrada/salida más importante. Los dispositivos de **Entrada/Salida** se pueden clasificar según el tipo de información que manejen en dos grandes grupos, *dispositivos de bloque* que almacenan la información en bloques de tamaño fijo, cada uno con su propia dirección, dentro de este grupo se encuentran los discos. Y dispositivos de carácter este dispositivo envía o recibe un flujo de

caracteres entre estos dispositivos tenemos las terminales, impresoras de línea y teclados. El software de control de entrada/salida se denomina Driver y está formado por un conjunto de rutinas y tablas que, ejecutan y controlan todas las operaciones de entrada y salida de cualquier periférico conectado a la computadora, siendo particulares para cada dispositivo.

El **sistema de archivos** es otro aspecto fundamental de un sistema operativo. Los directorios de archivos en sistemas informáticos pertenecen generalmente a una de dos categorías: de nivel único o jerárquico.

Un directorio de nivel único o plano, contiene todos los archivos del sistema o de un volumen dado. El directorio jerárquico cataloga en una jerarquía de directorios cuya estructura es similar a la de un árbol con raíz.

Es interesante como se asignan los archivos, esto también depende del sistema operativo. Los sistemas operativos UNIX y DOS tienen como tales su manera particular de asignar los archivos; a través de DOS se ilustra la asignación contigua de archivos en disco y con UNIX se ilustra una forma de asignación no contigua basado en los í-nodos.

Como plataforma computacional se utilizó WINDOWS y como lenguaje de programación Delphi versión 3.0

## **INTRODUCCIÓN**

Representar las ideas e imágenes que se forma de los hechos y de las cosas es una constante en el ser humano. Es comprensible entonces como las personas piensan en un proceso en memoria y lo asocian a una forma geométrica o como esquematizan el orden en que el disco atiende las peticiones que tiene en un instante de tiempo.

Estos y muchos otros aspectos del área de los sistemas operativos es la causa que lleva a la realización de este proyecto, en el cual se simulan algunos mecanismos de administración y control de un sistema Operativo, como son: Asignación de memoria, algunas generalidades de Entrada/salida, sistema de archivos y administración del disco, dichas simulaciones muestran todo tipo de situaciones puesto que las condiciones de los procesos, periféricos, memoria, disco, operaciones sobre los archivos son totalmente aleatorias.

Las situaciones son tratadas de acuerdo al mecanismo que se este simulando y las condiciones aleatorias que se presenten, tal como lo hace el sistema operativo, con que el objetivo de que el usuario perciba una situación lo más parecida a la realidad y pueda entender el mecanismo, establecer de manera clara sus ventajas y desventajas, y basándose en ello sea capaz de proponer variantes.

## 1. MARCO TEÓRICO

### 1.1 ASIGNACIÓN DE MEMORIA

Al ser la memoria un recurso tan influyente en el rendimiento general de una computadora, debe ser manejado de la forma más eficiente posible. Por este motivo, el sistema operativo dedica gran parte de su software al **administrador de la memoria**. Su labor consiste en llevar un registro de las partes de memoria que se estén utilizando y aquellas que no, con el fin de asignar espacio en memoria a los procesos cuando estos la necesiten y liberarlo cuando termine, así como administrar el intercambio entre la memoria principal y el disco, en los casos en que la memoria principal no pueda albergar todos los procesos.

A continuación se tratan aspectos básicos de la forma como el sistema operativo gestiona la memoria principal en un sistema con características de monoprogramación y/o multiprogramación. Se estudian métodos de asignación de memoria, estratificados en aquellos sistemas que asignan espacios de memoria contigua o espacios de memoria no contigua. Así como también aquellos métodos que “dividen” el proceso en porciones y/o

porciones lógicas (Funciones, Subrutinas, Subprogramas, Datos). Todos los métodos que se tratan tienen sus ventajas y sus desventajas y calificarlos de mejor o peor, depende básicamente del tipo de arquitectura computacional, como también del tipo de aplicaciones que se estén manejando en el sistema operativo.

**1.1.1 Mecanismo de asignación contigua simple.** Es el esquema más simple de administración de memoria, ya que es un solo proceso de usuario quien reclama un espacio de memoria determinada, es decir, el grado de multiprogramación es nulo. En este esquema la memoria tiene la semblanza que se muestra en la figura 1.

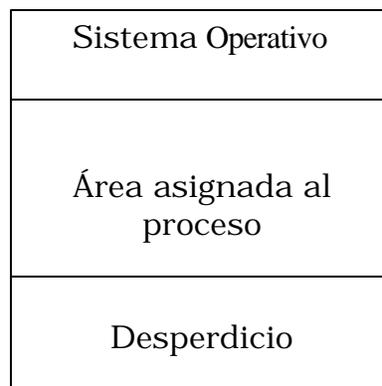


Figura 1 Contigua Simple

El sistema operativo es el primer elemento lógico que se aloja en la memoria. Cuando un proceso llega, solicita un espacio para su

alojamiento. El Administrador de Memoria verifica si hay espacio suficiente en memoria para cargar ese proceso, si no es así, se reporta un error de memoria insuficiente. Si hay espacio se le asigna. Una vez asignado el espacio de memoria correspondiente, se espera que el proceso termine, en ese momento el sistema operativo puede alojar otro proceso para su ejecución. Este modelo simple es usado en sistemas operativos monousuario como DOS.

Es necesario tener un mecanismo de protección del espacio de direcciones del sistema operativo, con el objeto de evitar que el proceso cargado en memoria no referencie y/o actualice alguna dirección ocupada por el sistema operativo. Generalmente estos son mecanismos que requieren alguna asistencia de hardware. Uno de ellos es la Implementación de un registro vigía utilizado para delimitar o separar los linderos del sistema operativo y del proceso cargado en memoria. Cuando este proceso referencie una dirección de memoria menor que el registro vigía, no le es permitido el acceso a dicha dirección, ya que esto denota que está intentando accesar una localidad utilizada por el sistema operativo. Otra forma puede ser asociar un bit con cada dirección de memoria, si este bit está encendido, denota que está ocupada la dirección.

**1.1.2 Mecanismo de asignación por particiones estáticas.** Una forma de introducir algún grado de multiprogramación es dividiendo la memoria disponible en particiones de tamaño determinado y que será fijo mientras el sistema esté activo. Cuando un proceso solicita memoria, el administrador lo aloja en una de esas particiones desocupadas teniendo en cuenta que se puede asignar esa partición, es decir, el tamaño de la memoria solicitada es menor o igual que el tamaño de la partición. El número y el tamaño de las particiones se basa en datos tomados de estadística, teniendo en cuenta el grado de multiprogramación requerido, tamaños típicos de los procesos y la memoria disponible. La figura 2 muestra la memoria dividida en 5 particiones, en una de las cuales se

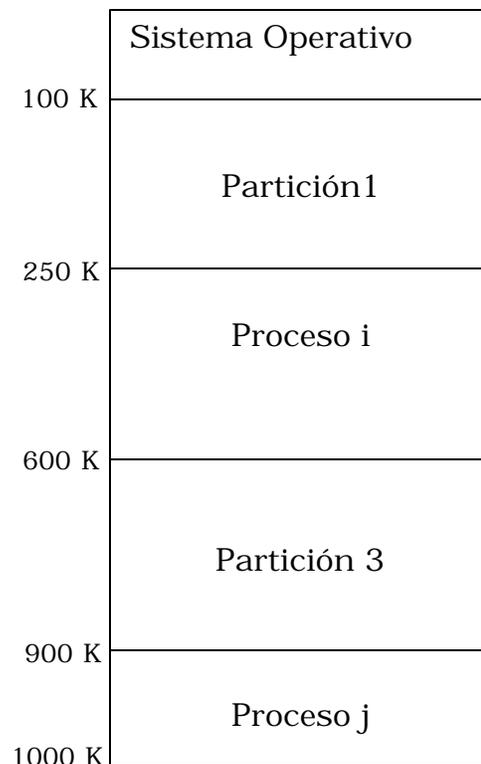


Figura 2 Particiones Estáticas

aloja el sistema operativo. En la partición 2 y la partición 4 se alojan los procesos i y j respectivamente. La partición 1 y 3 están libres y pueden ser asignadas.

Para gestionar la memoria basado en este esquema de asignación, el sistema operativo utiliza una estructura de datos llamada **TDP** (Tabla de descripción de particiones), la cual contiene los campos Identificador de partición, Tamaño de la partición, Dirección Base y Estado de la partición.

Para el caso ilustrado en la figura 2, la TDP contiene la siguiente información:

Id. Partición	Tamaño	Dir. Base	Estado
0	100K	0	Ocupada
1	150K	100	Libre
2	350K	250	Ocupada
3	300K	600	Libre
4	100K	900	Ocupada

Cuando un proceso reclama memoria, esto es cuando se activa o se crea en el sistema, el Administrador de memoria revisa la **TDP**. Si hay un espacio desocupado y además el proceso tiene un tamaño menor o igual

que el tamaño de la partición, se aloja el proceso en esa partición. Cuando se ubica el proceso, el bloque Descriptor del proceso es actualizado con los datos de la partición y la entrada en la **TDP** se actualiza como estado de Ocupada.

La asignación de una partición se puede hacer seleccionando la primera que aparezca en la tabla como Desocupada y en la cual el proceso quepa, o bien, seleccionar aquella partición en la que mejor quepa el proceso. Mediante esta segunda estrategia se garantiza un menor desperdicio de memoria, aunque lleva consigo un mayor tiempo de respuesta para la asignación.

Es posible que cuando un proceso reclame memoria no se le pueda asignar, debido a que todas las particiones están ocupadas, o bien, que entre las desocupadas no haya alguna lo suficientemente grande para alojar el proceso. En el segundo caso, el sistema operativo no tiene otra opción que reportar un mensaje de error de la imposibilidad de asignación.

Para el caso en que todas las particiones estén ocupadas, es posible que el sistema operativo desasigne memoria en algún proceso mediante algunas políticas, procesos sin actividad de CPU, o bien, procesos de baja prioridad. Los procesos desasignados son llevados a memoria auxiliar

(Disco) en espera de una pronta reasignación. Este concepto se denomina **Swapping**, el cual se discutirá más adelante.

**1.1.3 Mecanismo de asignación por particiones dinámicas.** Contrario al método anterior, en el cual las particiones se crean y son estáticas durante toda la administración, la asignación de memoria mediante este método crea las particiones dinámicamente, haciendo ver la memoria como un objeto que se ensancha o se comprime de acuerdo a las políticas de asignación de espacios libres y de reagrupación de espacios cuando se desasigna memoria a un proceso.

En principio, ni el tamaño ni el número de particiones necesitan estar definidas en tiempo de generación del sistema. En general el administrador de memoria puede seguir creando y asignando particiones a procesos solicitantes hasta que se agote la memoria o hasta cuando se llegue al límite permisivo de multiprogramación.

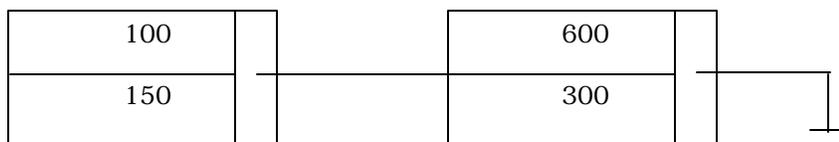
Para soportar la administración de memoria por particiones dinámicas, es necesario definir estructuras de datos adecuadas para el registro de particiones asignadas y para el control de espacios libres de memoria. Aparece entonces la **TPA** (Tabla de particiones asignadas), con el Identificador de partición, Tamaño de la partición, Dirección Base y Estado. De igual forma, una lista enlazada **LMD** (Lista de memoria

disponible), con la información de la dirección del espacio disponible y el tamaño disponible. Para el ejemplo anterior, ilustrado en la figura 2, el estado de las estructuras de datos sería:

Para **TPA**:

Id. Partición	Tamaño	Dir. Base	Estado
0	100K	0	Ocupada
1	350K	250	Ocupada
2	100K	900	Ocupada
3	-	-	Libre
4	-	-	Libre
5	-	-	Libre

Para **LMD**:



Cuando un trabajo reclama una asignación de memoria, asumiendo X K, un procedimiento de asignación sería:

- Buscar en **LMD** nodo para el cual  $X \leq$  tamaño disponible
- Si no lo encontró, es imposible asignar memoria. Espere
- Si lo encontró:

**Dirtemp = Dirección**

Si el tamaño es estrictamente menor:

$$\text{Dirección} = \text{Dirección} + X$$

$$\text{Tamaño} = \text{Tamaño} - X$$

Sino

Borre ese nodo y apunte al siguiente

- Buscar en **TPA** una entrada libre, actualice

$$\text{Dirección} = \text{Dirtemp}$$

$$\text{Tamaño} = X$$

El anterior procedimiento puede tener alguna variación dependiendo de los criterios definidos para la forma como se asignará una partición libre. Esto es: en la partición que primero ajuste o en el que mejor ajuste.

Cada uno de los anteriores criterios tiene algunas ventajas con respecto al otro, pero en general no hay una regla fija para el escogimiento de alguno específicamente. El primer ajuste barre la lista de espacios disponibles y toma el primero en el cual quepa el trabajo. La ventaja de éste es que no tiene necesidad de recorrer la totalidad de la lista para la asignación. El mejor ajuste recorre toda la lista (a menos que la lista esté ordenada por

tamaño, cosa que rara vez sucede) y toma justamente el nodo donde mejor quepa el proceso. Éste método tiene un peor tiempo de respuesta que el anterior, por aquello que debe recorrer toda la lista de disponibles, además, genera también un mayor número de fracciones pequeñas de memoria disponible, con lo cual habrá muy seguramente muchos fallos en los intentos de asignación. Esto se mejora introduciendo el concepto de **Compactación** que se estudia más adelante.

Cuando un proceso termina la ejecución, es necesario actualizar las estructuras de datos, recuerde que los datos referente a la asignación de memoria están en el bloque de descripción de procesos, allí se encuentra la identificación de la partición donde se asignó el proceso, llamémosla P. Un procedimiento de actualización de las estructuras de datos puede ser:

- Buscar en **TPA** una entrada con la identificación de partición P  
Dirtemp = Dirección  
Tamtemp = Tamaño  
Dirección = NULL  
Tamaño = NULL  
Estado = Libre
- En la **LMD** verificar si el espacio que se deja libre en memoria es adyacente con uno o dos nodos, es decir, adyacente por encima, por debajo o por ambos (ver figura 3). Si esto es cierto, actualizar un nodo

con la información de los demás, dejando un solo nodo representativo de los espacios libres adyacentes. Sino hay adyacencia, es necesario crear un nuevo nodo con la información:

Dirección = Dirtemp

Tamaño = Tamtemp

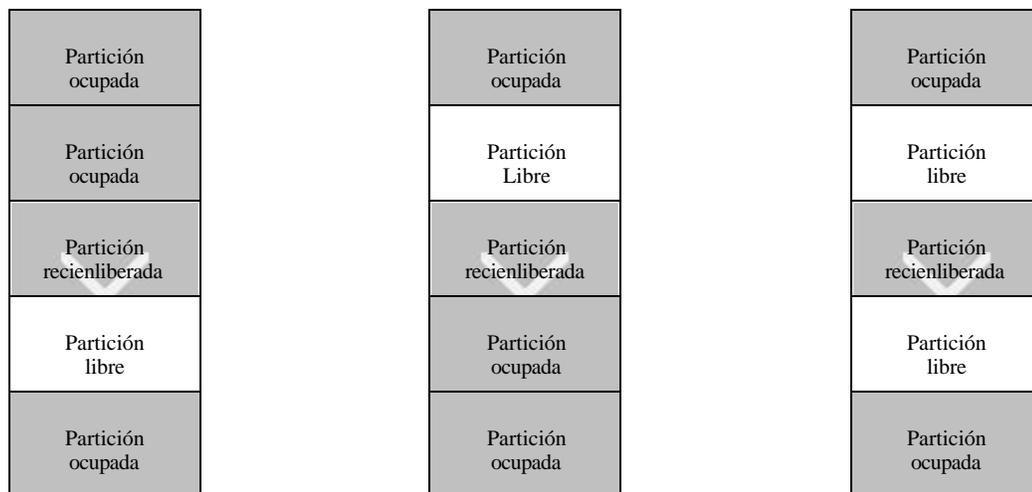


Figura 3 Particiones libres adyacentes

**1.1.4 Compactación.** Se presenta una alternativa para la solución del problema de la fragmentación, y consiste en compactar los espacios de memoria disponible con el objeto de crear espacios contiguos de mayor tamaño, se puede ejecutar este procedimiento cuando termine un proceso, o bien, cuando halla una falla de asignación por problema de tamaños disponibles no contiguos. La compactación no es más que reubicación, lo

que implica una serie de actualizaciones en las estructuras de datos que soportan esta administración. La ventaja de usar este procedimiento es que aumenta el grado de multiprogramación, pero requiere de un soporte de hardware sofisticado y costoso, además de un mayor tiempo de procesamiento (El tiempo que toma la reubicación en memoria y la actualización de las tablas).

En este tipo de asignación se requiere que los programas sean reubicables para poder cambiar de zona de memoria principal.

**1.1.5 Mecanismo de asignación por paginamiento puro.** La memoria Física se divide en una serie de porciones de tamaño fijo, llamados **Bloques**. El espacio de direcciones de un proceso se divide en porciones de igual tamaño que un bloque de memoria, a esto se le denomina **Página**. Entonces, la asignación de memoria consiste en hallar un número suficiente de bloques de memoria sin utilizar para cargar en ellos las páginas de los procesos. Aunque las páginas parezcan al usuario con ubicación contigua, esto no necesariamente es cierto en los bloques de memoria.

Para soportar la administración se tienen **TMP** (Tablas de mapa de páginas), la cual se construye al tiempo de carga del proceso. Ahora cada proceso tiene un espacio de direcciones virtuales desde donde se dividen las páginas (Figura 4).

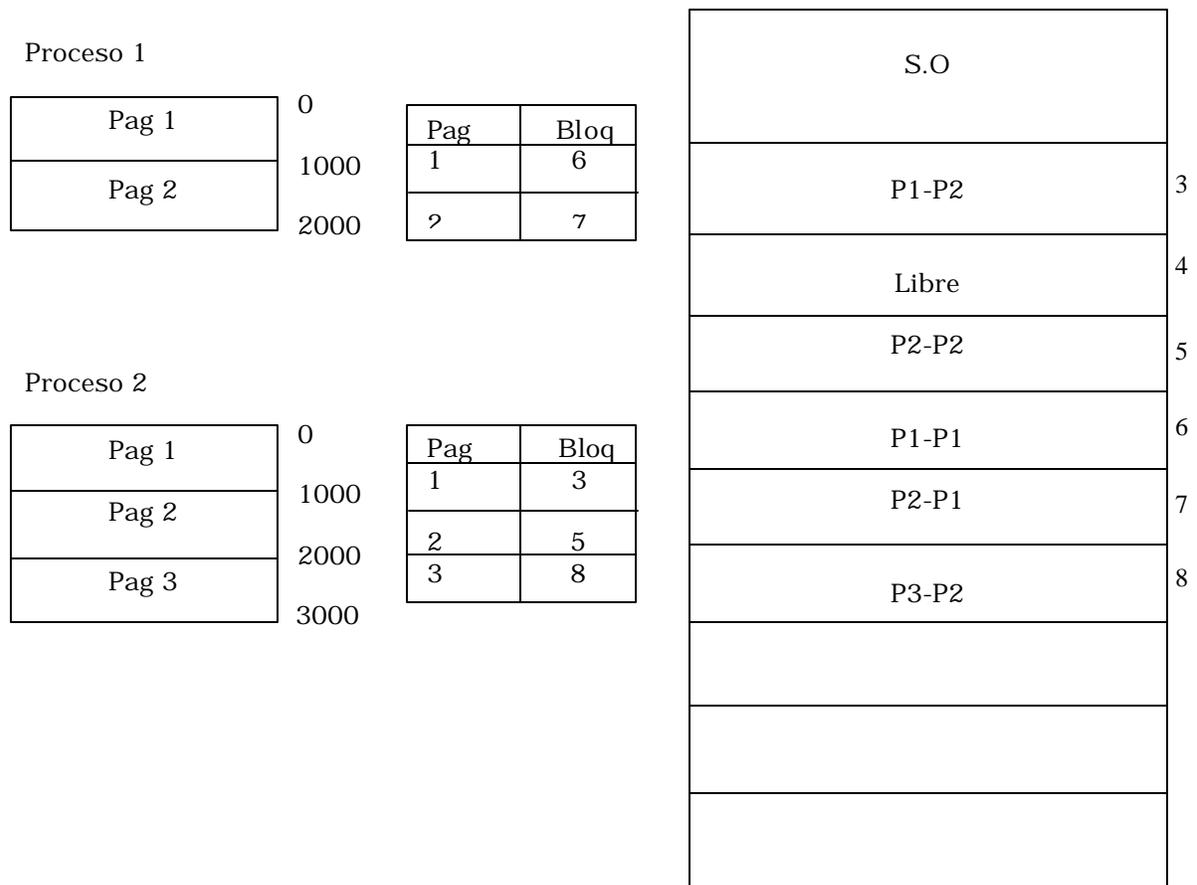


Figura 4 Paginamiento Puro

Como se puede ver en la figura 4, hay una entrada en la TMP por cada página del proceso. Existe un soporte de hardware para la paginación que se concentra en mantener la memoria necesaria para almacenar las tablas y hacer una rápida conversión de memoria virtual a memoria física.

Una dirección virtual está compuesta por un número de página y el desplazamiento dentro de esa página. Por ejemplo:

LDA 003 200

En la operación anterior del proceso 3, LDA (referencia) a su página 3, con un desplazamiento de 200 unidades. Para la conversión de ésta dirección virtual a dirección de memoria real, debe hacer un indexamiento de la página 3 en la TMP del proceso 3, con ello se obtiene el bloque de memoria donde está alojada esa página. Luego se desplaza 200 unidades en ese bloque y obtenemos la dirección real.

La cantidad de TMP existentes en sistemas con paginamiento, invitan a minimizar la memoria necesaria para su almacenamiento. Esto se puede lograr dinamizando las entradas en las TMP, teniendo solo tantas entradas por cuantas páginas halla para el proceso. Se tiene un registro límite de la tabla de mapa de página (RLTMP), que indica el número máximo de entradas para esa tabla. El acceso a una TMP se hace mediante un apuntador a una dirección para la TMP, esa dirección se aloja en un registro base de tabla de mapa de página (RBTMP). Los valores de este par de registros se inicializan en el momento que se carga el proceso para su

ejecución y se almacenan en el respectivo registro del Bloque Descriptor de Procesos.

Es necesario tener para este esquema de administración de memoria un hardware especializado que realice el direccionamiento de la memoria real por intermedio de una dirección virtual, esto con el objeto de tener unos tiempos de respuesta razonables.

Un método a considerar es el uso de una memoria asociativa de alta velocidad que almacene el subconjunto de entrada de tablas de mapas de página más frecuentes usadas, esto es básicamente un caché de traducciones. El esquema de direccionamiento a partir de la dirección virtual (No. Página, Desplazamiento) es: ver si la página referida (No. Página) aparece en caché, si esto es cierto, entonces obtener desde ese búffer el bloque de memoria donde está alojada esa página. En ambos casos, si se encuentra en el caché o no, se toma el desplazamiento dado en la dirección virtual para obtener la dirección de memoria real referida.

Otra estructura de datos básica para el soporte de esta forma de administración de memoria, es la **TBM** (Tabla de Bloques de Memoria), cuyas entradas básicas son: Identificador del bloque, Estado y Proceso. Esta tabla lleva un control de los estados de cada uno de los bloques y si

está asignado, qué proceso lo utiliza. Es sencillo observar que el tamaño de esta tabla es estático y el número de entradas es:

$$N = \text{Tamaño\_Memoria} / \text{Tamaño\_Página}$$

Cuando se solicita asignar un proceso de tamaño  $t$ , el sistema operativo debe asignar  $N_b$  bloques de memoria libre, de tal forma que:

$$N_b = t / \text{Tamaño\_Página}$$

Lo anterior indica que si el tamaño de un proceso no es múltiplo del tamaño de una página, el último bloque de memoria asignado a ese proceso va a estar subutilizado, generando algún grado de fragmentación.

Una generalización de un algoritmo de asignación fácilmente concebible sería:

·  
·  
·

- Petición de  $t$  K
- $N_b = t / \text{Tamaño\_Página}$
- Si Número de Bloques Libres  $< N_b$ , entonces “No es posible asignar memoria en este momento”. Esperar

- Sino
  - Generar TMP con Nb entradas
  - Inicializar Registros RLTMP y RBTMP
  - Inicializar BDP (Bloque Descriptor de Proceso) con datos de esta administración.
  - Asignar Nb bloques de memoria a Nb páginas
  - Actualizar TBM
  
- Regrese

Algunas ventajas de este método entre otras son: disminuye la fragmentación, aumenta el grado de multiprogramación, los procedimientos de asignación y desasignación son relativamente simples.

Algunas desventajas son: recargo de almacenamiento para la gestión de administración, debido al uso de una TMP por proceso, una Tabla de Bloques global de memoria, desperdicio de algunas fracciones de bloques de memoria cuando el tamaño del proceso no es múltiplo del tamaño de la página y requerimiento de un hardware especial para soportar un mecanismo de conversión rápido de memoria virtual a memoria real.

**1.1.6 Mecanismo de asignación por paginamiento con demanda.** Los Métodos de administración hasta ahora abordados, tienen como

característica común, el no poder asignar memoria hasta tanto no halla el espacio suficiente de memoria libre para alojar la totalidad del proceso. Ahora se muestra una forma de administración en la cual no es necesario tener la totalidad de un proceso en memoria para su ejecución.

Aquí se tiene los procesos residentes en un espacio de almacenamiento secundario de igual forma paginados como en el método anterior, pero con la diferencia de solo llevar a memoria aquellas páginas que están siendo referenciadas en la ejecución y el resto permanece residente en el medio de almacenamiento auxiliar (generalmente el Disco). Aparece entonces el concepto de **Memoria Virtual**, por el hecho de tener un medio alternativo de albergar los procesos distinto al de memoria real, esto genera una “ilusión” de una memoria de gran tamaño. Lo que sí es claro es que debe residir en memoria todo aquel elemento referenciado en el momento de ejecución. Esto último genera la posibilidad de que una instrucción cualquiera referencie un dato que está en una página que no se encuentra cargada en memoria real. Es necesario entonces, traer esa página que está en memoria virtual a memoria real, en este momento se produce una interrupción por “**Fallo de Página**”, el proceso se bloquea por una solicitud de E/S y se le da la oportunidad al planificador de procesos que conceda al procesador a otros procesos mientras se trae la página solicitada a memoria real.

Cuando se está solucionando el problema de la ausencia de una página en memoria, es posible que el gestor de memoria no encuentre un bloque libre para asignarse a la página, en cuyo caso es necesario remover algunas de las páginas residentes en memoria, siguiendo algún procedimiento específico, ya sea **LRU** (Least Recently Used), o **FIFO** (First In, First Out).

Traer una página desde Disco a memoria real, origina el concepto de **Swapping**.

Las estructuras de datos necesarias para el soporte de este tipo de administración son las mismas que las usadas en la administración por paginamiento con algunas adaptaciones.

La **TMP** contiene un campo (Bit) adicional que indica si la página está en memoria o no, estará encendido o apagado, si está presente o ausente respectivamente en la memoria real. Aparece un nuevo elemento llamado **TMA** (Tabla de Mapa de Archivo), la que contiene las direcciones de almacenamiento secundario (generalmente Disco), donde reside cada una de las páginas de un proceso. El gestor de memoria utiliza la TMA para direccionar aquellas páginas referenciadas y no existentes en memoria real. Al igual que las direcciones bases de la TMP, las direcciones bases de la TMA se almacenan en el Bloque de Control de Proceso.

El siguiente es un procedimiento que muestra paso a paso los principios de operación de este método de administración de memoria:

- ( $\beta$ ): Procesando la instrucción donde se encuentra el Apuntador
- Genere la página a partir de la dirección virtual de la instrucción
- Accese la TMP y verifique si esta página está en memoria o no
- Si la página está en memoria, entonces
  - Acceder los datos referidos y complementar la instrucción
  - Incrementar el Apuntador
  - Ir a  $\beta$
- Si no está en memoria, entonces
  - Interrupción por falta de página

Cuando ocurre una interrupción por falta de páginas se invoca al sistema operativo para que el gestor de memoria solucione la ausencia de la página, recuperándola desde el Disco. El procedimiento es el siguiente:

- Si hay bloques libres en memoria
  - Obtener No. Página necesitada
  - Obtener la página desde el Disco
  - Llevar página a memoria
  - Actualizar tablas

- El proceso vuelve a ejecución. Seguiría en  $\beta$
- De lo contrario
  - Seleccionar la página a remover según FIFO o LRU
  - Actualizar tablas
  - Si la página ha sido actualizada, rescríbala en disco
  - Obtener No. Página Necesitada
  - Obtener la página desde el disco
  - Llevar página a memoria
  - Actualizar tablas
  - El proceso vuelve a ejecución. Seguiría en  $\beta$

A continuación se describen dos formas de seleccionar la página a remover cuando hay fallo de página.

**1.1.6.1 Algoritmo de reemplazo FIFO (Primero en entrar primero en salir).** Este algoritmo remueve la página con más tiempo en memoria, para el control del tiempo se utiliza una cola de llegada, la primera de la cola será aquella que lleva más tiempo cargada y según el criterio del algoritmo es la primera en ser sustituida.

La implementación de este algoritmo es sencilla pero su eficacia es relativa, puesto que es probable que la página más antigua sea

referenciada con bastante frecuencia en cuyo caso su remoción provocará sucesivos fallos de páginas. Por lo anterior este algoritmo es inadecuado para sistemas de tiempo compartido; pero puede ser útil en entornos batch.

**1.1.6.2 Algoritmo de reemplazo LRU (La de menor uso reciente).** En este algoritmo se remueve la página menos usada recientemente, para ello el sistema contabiliza el tiempo en que se produce cada referencia a memoria para poder clasificar las páginas según su uso.

Para contabilizar el tiempo, el algoritmo requiere de hardware especial en su implementación.

**1.1.7 Mecanismo de asignación por segmentación.** Un segmento de memoria constituye un espacio independiente de direcciones, estos pueden crecer o reducirse sin afectar a los demás. Un segmento puede contener un procedimiento, un arreglo una pila, o bien un conjunto de variables escalares.

La carga de un programa en memoria exige la búsqueda de los bloques adecuados a sus segmentos y, puesto que estos son de tamaño variable se

ajustaran lo más posible a las necesidades produciéndose pequeños desperdicios. En este caso se produce fragmentación externa.

**2.1.8 Mecanismo de asignación por segmentación y paginamiento con demanda.** Con el paginamiento y la segmentación en forma independiente se tienen ventajas e inconvenientes, es decir, si los segmentos son muy grandes podría ser inconveniente e incluso imposible mantenerlos en la memoria en su totalidad, esto hace que surja la idea de paginarlos de forma que sólo permanezcan en memoria las paginas que sean necesarias.

De esta manera se distinguen procedimientos y datos para una mayor protección mediante los segmentos a la vez que se obtiene un gran espacio lineal de direcciones a través del paginamiento.

## **1.2 ALGUNAS GENERALIDADES DE ENTRADA/SALIDA**

El control de las operaciones de Entrada/Salida es otra de las misiones que debe realizar un sistema operativo para facilitar el uso de los distintos dispositivos.

Seguidamente describiremos la gestión de los dispositivos de Entrada/Salida y cómo se realiza la interfaz entre los programadores y dichos dispositivos.

**1.2.1 Dispositivos de Entrada/salida.** Los dispositivos de Entrada/Salida se pueden reunir en tres grandes grupos: Discos(Este tipo de dispositivos de E/S es abordado con detalle más adelante); terminales, la cual está constituido por un teclado por medio del cual se introducen datos y una pantalla que recibe dichos datos. Por último tenemos las líneas de comunicaciones, encargadas de conectar computadoras entre sí y estas con sus terminales cuando la distancia es grande.

Los dispositivos de Entrada/Salida también se pueden clasificar dependiendo del tipo de información que manejen en **dispositivo de bloque**, el cual almacena la información en bloques de tamaño fijo, cada uno con su propia dirección. La característica principal de estos dispositivos es la posibilidad de leer o escribir en un bloque de forma independiente de los demás, entre este tipo de dispositivos se encuentran los discos. El otro tipo de dispositivos son los **dispositivos de caracter**, los cuales envían o reciben un flujo de caracteres, sin sujetarse a una estructura de bloque, no se pueden utilizar direcciones ni tienen una

operación de búsqueda, entre estos dispositivos tenemos las terminales y teclados.

**1.2.2 Interfaz Procesador-Periferico.** Las interfaces actúan como intermediarios entre el sistema informático y los dispositivos de Entrada/Salida. Su propósito es superar las incompatibilidades de velocidad y de señalización de niveles entre el procesador y los periféricos y traducir las órdenes de Entrada/Salida genéricas emitidas por el procesador a controles específicos del dispositivo.

El software de control de Entrada/Salida se denomina **DRIVER** el cual está formado por un conjunto de rutinas y tablas que, formando parte del núcleo del sistema operativo, ejecutan y controlan todas las operaciones de entrada y salida sobre cualquier periférico conectado a la computadora, siendo particulares para cada dispositivo.

### **1.3 SISTEMA DE ARCHIVOS**

Los archivos de usuario y sistema disponibles están catalogados en directorios de archivo.

El contenido de un directorio puede ser visualizado en una terminal o en una impresora. Los directorios de archivos en sistemas operativos pertenecen generalmente a una de dos categorías: de nivel único o jerárquico, como se muestra en las figuras 5 y 6 respectivamente.

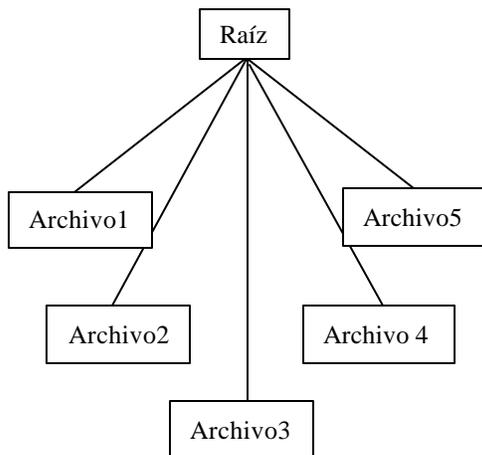


Figura 5 Sistema de archivos Plano

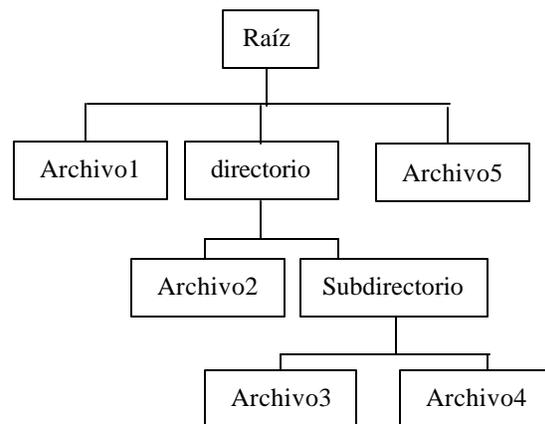


Figura 6 Sistema de archivos Jerárquicos

**1.3.1 Sistema de Archivos Planos.** En este tipo de organización se utiliza un directorio de nivel único, generalmente llamado directorio plano, este contiene todos los archivos del sistema o de un volumen dado. Su simplicidad y facilidad de implantación hace que los directorios planos sean comunes en sistemas pequeños monousuario.

Los directorios planos son inconvenientes y con frecuencia inadecuados para sistemas grandes con muchos archivos y posiblemente muchos usuarios. La inspección del contenido de un directorio plano puede ser inconveniente incluso en un sistema monousuario, ya que la lista de todos los archivos puede ser demasiado grande y la búsqueda de uno en particular puede gastar demasiado tiempo.

Los directorios planos presentan problemas más palpables en sistemas grandes entre los que se incluyen las dificultades para proporcionar un nombre único a los archivos y el construir un soporte inadecuado para compartir y proteger selectivamente los archivos en sistemas multiusuarios.

**1.3.2 Sistema de Archivos Jerárquicos.** Los inconvenientes que presentan los directorios planos pueden ser amortiguados utilizando directorios jerárquicos en el que los usuarios pueden agrupar los archivos relacionados en subdirectorios. Todos los archivos del sistema se catalogan en una jerarquía de directorios cuya estructura es similar a la de un árbol con raíz.

Entre las ventajas que presentan los directorios jerárquicos encontramos las siguientes:

Facilidad para designar unívocamente a los archivos, soporte para compartición y protección selectiva y conveniencia en la manipulación de directorios.

**1.3.3 Implantación de un Sistema de Archivos.** Desde el punto de vista de la implantación del sistema, interesa la forma de almacenamiento de los archivos y directorios, la administración del espacio en disco y la forma en que todo esto sea eficiente y confiable.

**1.3.4 Administración de archivos en el sistema operativo DOS.** La asignación del espacio a ocupar por un archivo en disco se realiza de forma contigua mientras que exista espacio, pero en el momento que al ampliar un archivo no existan sectores contiguos, el sistema le asigna sectores libres de forma no contigua. De esta forma se evitan problemas de fragmentación interna pero se complica la acción de asignar espacio a los archivos. El sistema operativo DOS cuenta con un conjunto de órdenes para mantener una estructura multinivel de directorios en forma de árbol.

**1.3.5 Administración de archivos en el sistema operativo UNIX.** Para el sistema operativo UNIX, el sistema de archivos es una partición que ha sido formateada en bloques de datos y que contiene una estructura de tablas que definen la localización de los archivos y directorios. La etiqueta del disco (VTOC) contiene la tabla de particiones del disco y se encuentra localizada en el primer sector del disco.

El disco además de VTOC, posee una tabla llamada superbloque, que contiene información acerca del sistema de archivos, tales como: El número de bloques de datos, el número de grupos de cilindros del disco, tamaño de un bloque de dato, una descripción del hardware y el nombre del Mount Point o directorio sobre el cual está montado el sistema de archivos.

En UNIX cada archivo tiene asociado un í-nodo, el cual es la descripción interna de un archivo y de sus bloques de datos. Los í-nodos contienen la siguiente información de un archivo: el tipo de archivo, el identificador del usuario al cual pertenece el archivo, el identificador del grupo al cual pertenece el archivo, el tamaño del archivo, el tiempo de su ultimo acceso y modificación.

Un í-nodo contiene dos tipos de apuntadores: directo e indirecto (Figura 7), el apuntador directo referencia directamente los bloques de datos. Para algunas versiones de UNIX se tienen hasta 12 apuntadores directos, es decir hasta 96 K. Los apuntadores indirectos se clasifican en simple indirecto, el cual se refiere a un bloque que contiene un apuntador a un bloque de datos y doble indirecto que se refiere a un bloque del sistema de archivos que contiene un apuntador simple indirecto.

Un disco se divide en grupos de cilindros para crear grupos de í-nodos que estén distribuidos por todo el disco. De esta forma los í-nodos estarán en bloques situados físicamente cerca de bloques de archivos a los que hacen referencia con esto se gana velocidad de acceso a los datos del archivo, ya que los desplazamientos de las cabezas lectoras para pasar de los bloques donde están situados los í-nodos a los bloques de datos serán más cortos.

Además, la seguridad del sistema ante pérdidas accidentales de información se ve incrementada, ya que si se dañan los bloques físicos de algunos í-nodos, no se habrán perdido todos los í-nodos

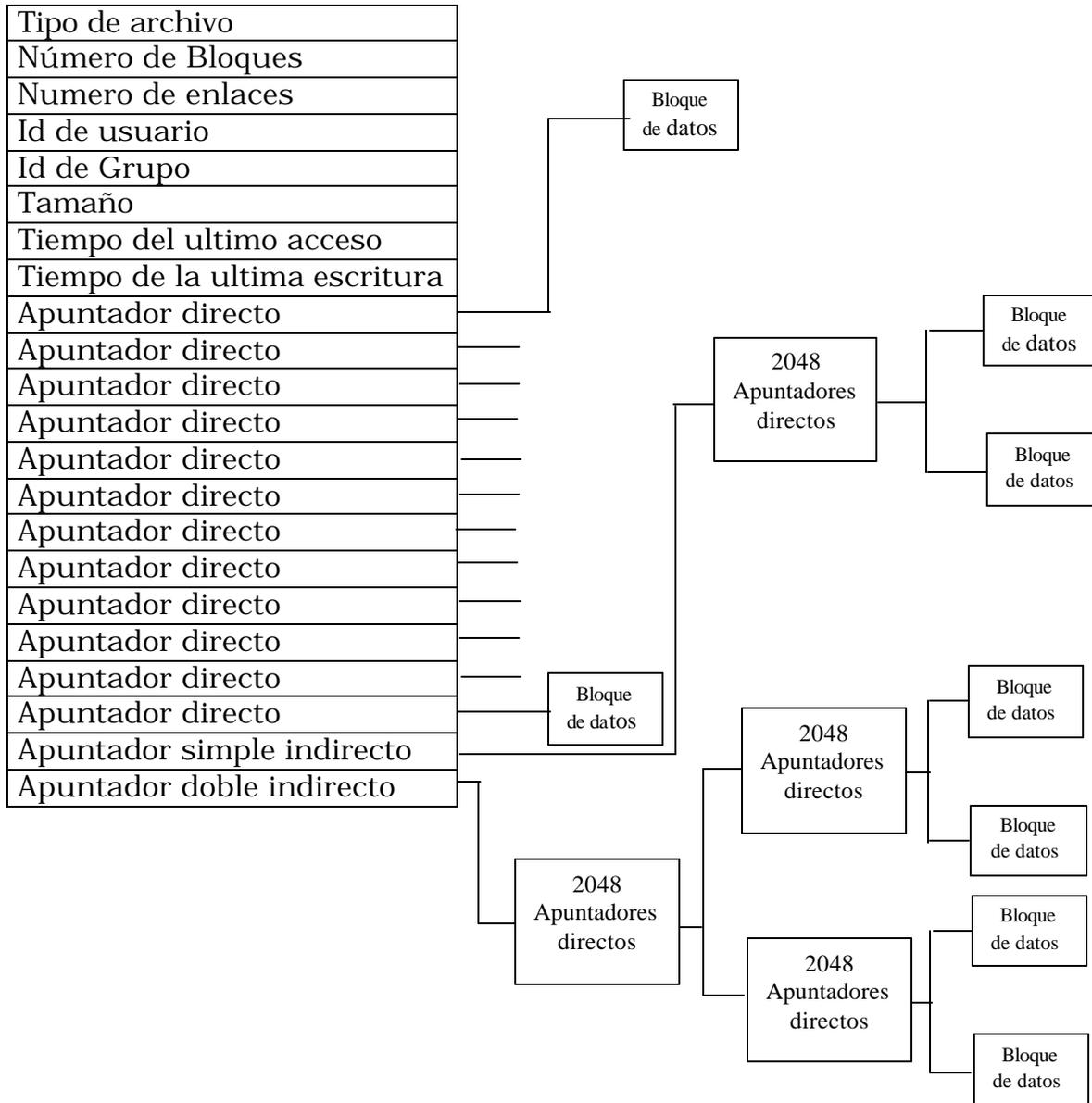


Figura 7 Representación de un í-nodo

## 1.4 ADMINISTRACIÓN DEL DISCO

Los discos son los dispositivos para almacenamiento secundario más comunes. Aceptan y recuperan datos a altas velocidades.

Los discos pueden ser fijos y removibles. Cada disco se divide en pistas concéntricas de tal forma que todas las homólogas de los distintos discos forman un cilindro.

Un acceso a disco implica un tiempo de retardo mientras se desplazan algunos elementos electromecánicos, este retardo está compuesto por tres tiempos los cuales son mostrados en la figura 8.

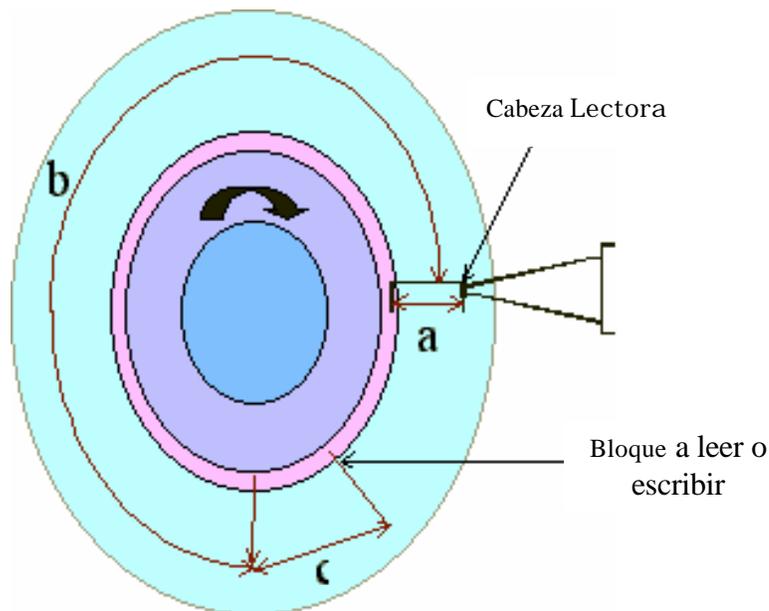


Figura 8 Tiempos de acceso a disco

- a. **Tiempo de búsqueda:** Tiempo que tarda en mover el brazo buscando la pista o cilindro correspondiente
  
- b. **Tiempo de latencia:** Tiempo en que espera a que el bloque se sitúe frente a la cabeza de lectura / escritura
  
- c. **Tiempo de Transferencia:** Tiempo que tarda en transmitir el bloque de estado

La administración del disco tiene entre sus objetivos minimizar los tiempos de acceso al disco, para cumplir con este objetivo se busca planificar los accesos al disco a través de un algoritmo eficiente.

**1.4.1 Algoritmos para atención de peticiones.** Se han propuesto varios algoritmos para atender a las peticiones de disco que realizan los procesos y que se encuentran en un momento determinado en una lista de espera. A continuación se presentan cuatro algoritmos de administración del disco y se representa los desplazamientos de la cabeza lectora (brazo del disco) al atender algunas solicitudes representadas por el número de pista en el cual se encuentra el bloque a leer o escribir.

Pistas: 98 183 37 122 14 124 65

**1.4.1.1 Primero en llegar, primero en acceder (FCFS):** Este es el algoritmo de más fácil implementación y manejo pero de eficacia relativa. Según este criterio se atienden los accesos en su orden de llegada (figura 9).

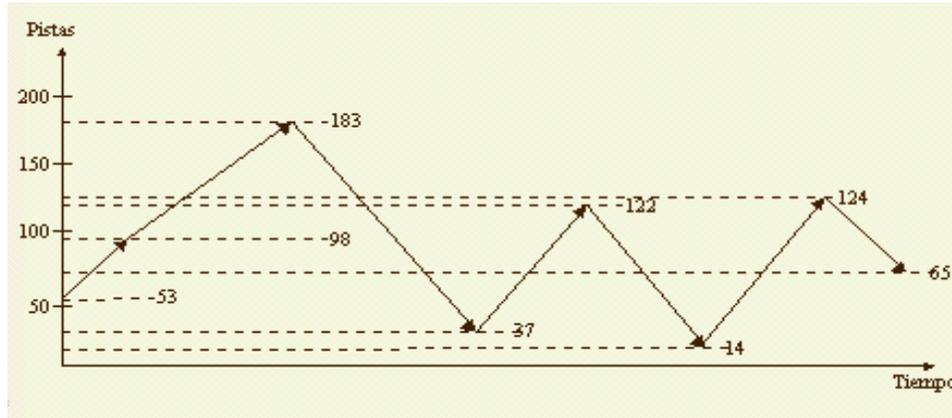


Figura 9 Planificación del disco FCFS

Para este caso el brazo se ha desplazado un total de 638 cilindros.

**1.4.1.2 Primero el de menor tiempo de búsqueda:** Su método es atender primero la petición más cercana a la última servida, o lo que es

mismo aquella en que se requiera menor desplazamiento del brazo (figura 10).

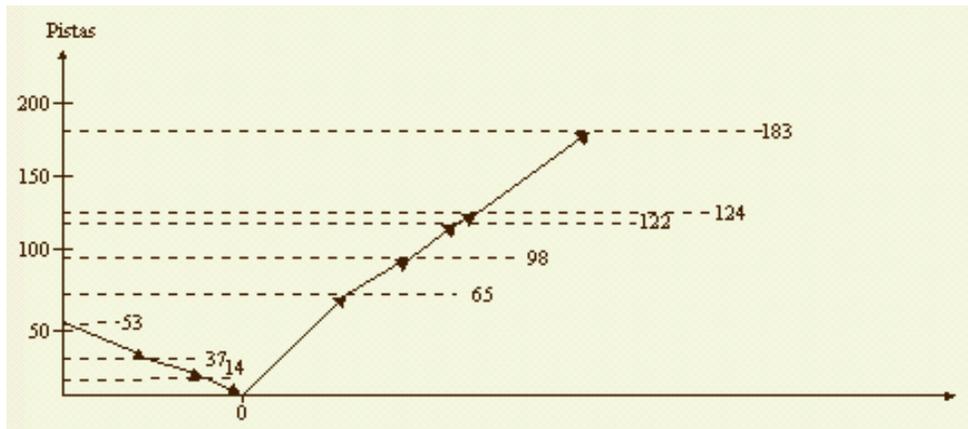


Figura No. 10 Planificación del disco SSTF

Para este caso el brazo se ha desplazado un total de 232 cilindros.

**1.4.1.3 Exploración (SCAN):** La exploración establece un recorrido predeterminado del brazo del disco que va del primer al último cilindro y vuelta. En este recorrido ira atendiendo las peticiones más cercanas según el sentido del desplazamiento (figura 11). A partir de este algoritmo se han desarrollado otros más que actualmente son utilizados.

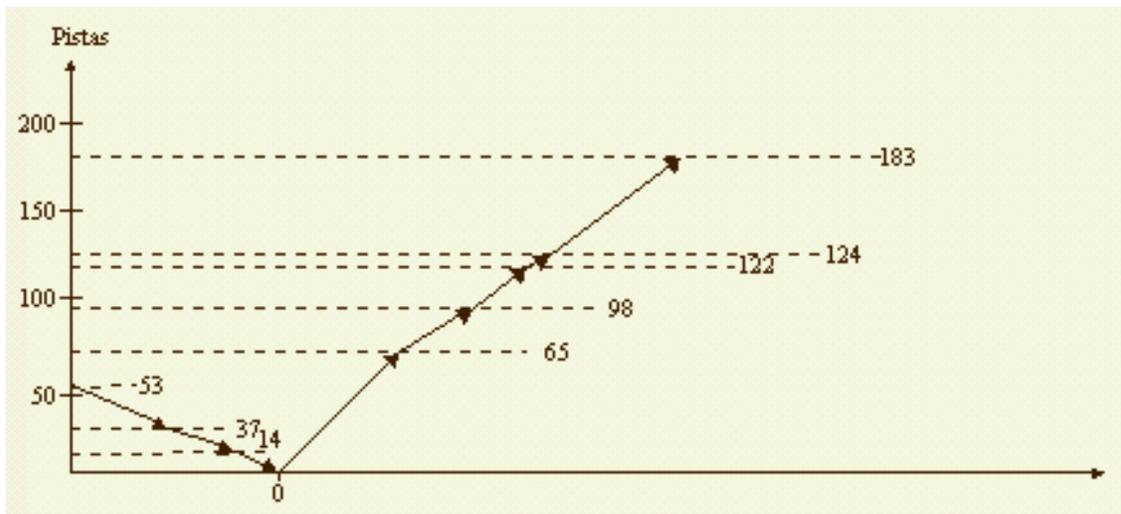


Figura 11 Planificación del disco SCAN

**1.4.1.4 Exploración circular (C-SCAN):** Este algoritmo es una variación del anterior, en este caso el movimiento del brazo sigue siendo el mismo, pero solo atiende peticiones en uno de los dos sentidos (figura 12). Al llegar al último cilindro regresará al principio sin atender petición alguna.

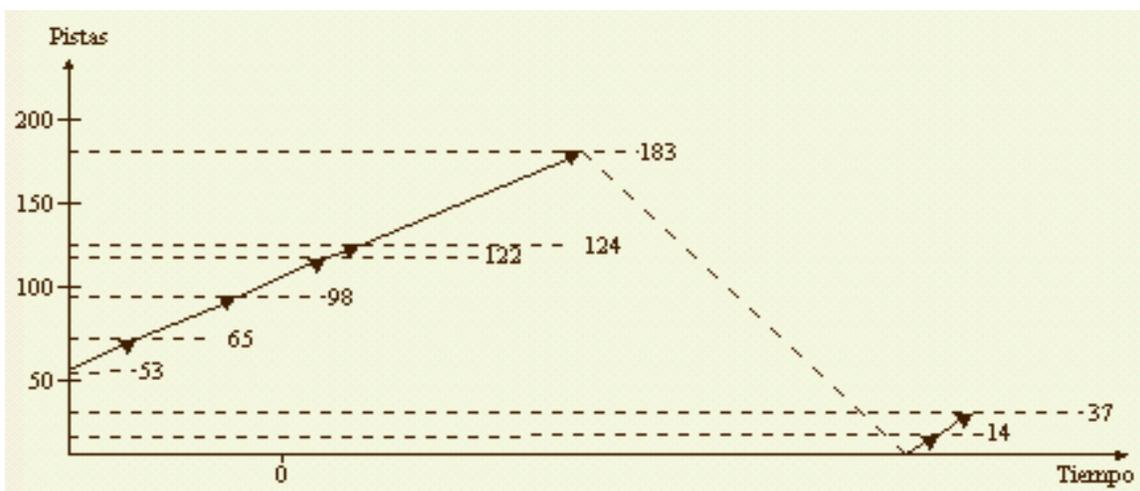


Figura 12 Planificación del disco C-SCAN

## **2. EL PROBLEMA Y OBJETIVOS DE LA INVESTIGACIÓN**

### **2.1 ANTECEDENTES DEL PROBLEMA**

Durante su evolución, los sistemas operativos han pasado por diversas etapas y han sido monolíticos, monousuarios, multiusuarios, multitarea, de red y distribuidos, han tenido diversas formas de procesamiento, interactivo o en batch.

En sus inicios se le dio más importancia a su eficiencia, actualmente se busca un equilibrio entre la eficiencia y la facilidad de uso, estos aspectos son la base fundamental por la cual se realizó este proyecto, pues, hasta ahora no existía una herramienta de apoyo para el estudio y mejor comprensión de los sistemas operativos. No se tienen precedentes de simulación de estos mecanismos de administración de los sistemas operativos. Un punto importante a tener en cuenta, es la existencia de la segunda parte de la simulación de dichos mecanismos, presentado también como proyecto de grado, en la que se trata, por ejemplo: la planificación de procesos y otra serie de aspectos que complementan la gestión de un sistema operativo.

## **2.2 IDENTIFICACIÓN DEL PROBLEMA**

El área de sistemas operativos abarca una serie de temas que si son tratados en forma puramente teórica son difíciles de comprender, puesto que no existe forma de visualizar y seguir paso a paso cada uno de los mecanismos empleados por el sistema operativo para administrar eficientemente los recursos ofrecidos por el hardware.

## **2.3 PLANTEAMIENTO DEL PROBLEMA**

Debido a la naturaleza abstracta de los temas que abarca el área de los sistemas operativos, las personas interesadas en su estudio perciben cada una de sus gestiones y tratan de representarlas a su juicio.

Lo anterior muestra la carencia de una herramienta de apoyo para afianzar el conocimiento en los conceptos de diseño e implementación de los sistemas operativos.

## **2.4 ANÁLISIS DEL PROBLEMA**

Teniendo presente la falta de una herramienta de apoyo en el estudio de los sistemas operativos, se estudio esta área y se decidió buscar el medio para mostrar en forma práctica los siguientes temas:

- Asignación de memoria
- Algunas generalidades de entrada/salida
- Sistema de archivos
- Administración del disco

Para hacer de estos temas una realidad más tangible de la que actualmente se presenta.

## **2.5 FORMULACIÓN DEL PROBLEMA**

El problema consiste en la falta de una herramienta de apoyo para afianzar el conocimiento en los conceptos de diseño e implementación de los sistemas operativos.

## **2.6 OBJETIVOS**

**2.6.1 Objetivo General.** Implementación de una herramienta de simulación de la gestión y control de un sistema operativo donde el estudiante pueda verificar y afianzar los conocimientos teóricos adquiridos en el curso de sistemas operativos.

**2.6.2 Objetivos Específicos.** A través de las simulaciones que se realizarán, se persigue alcanzar los siguientes objetivos:

- Reconocer las diferentes formas de organización de la memoria y su funcionamiento.
- Comprender la gestión de entrada/salida al nivel de software.
- Diferenciar la implantación de los sistemas de archivos planos y jerárquico.
- Visualizar las características de un disco.
- Ilustrar los diferentes métodos de planificación de los accesos al disco.

- Mostrar la forma de los sistemas de archivos para UNIX y DOS.

## **2.7 RESULTADOS DE LA INVESTIGACIÓN**

Al culminar este trabajo de grado, se entrega a la Corporación Universitaria Tecnológica de Bolívar, una herramienta de apoyo en la cual se simula algunos mecanismos de administración y control de un sistema operativo, a través de dichas simulaciones se facilita el estudio de los sistemas operativos tanto para docentes como para estudiantes y todas aquellas personas interesadas en el área. Junto con ésta herramienta de apoyo se entrega:

- Documento de tesis.
- Manual de soporte del software.
- Software en formato CD que contiene instaladores, fuentes y ejecutable.

### **3. ENTORNO DEL DISEÑO DEL SOFTWARE**

#### **3.1 INTRODUCCIÓN**

A través de esta simulación, las personas interesadas en el estudio del área de los sistemas operativos tienen la oportunidad de observar gráficamente algunas gestiones que lleva a cabo un sistema operativo, dichas gestiones son realizadas de tal manera que al usuario final le resulta difícil entenderlas con solo teoría, ya que son muchas las tablas, enlaces, direcciones y otra serie de aspectos que se deben tener en cuenta. Por ello, y sin alcanzar a ser un tutorial que se base en una metodología educativa, esta herramienta resulta indispensable en el proceso de enseñanza / aprendizaje, ya que brinda de manera muy clara una nueva opción que rompe con el estilo tradicional con que se viene impartiendo el conocimiento acerca del área.

Mediante la ejecución de las distintas simulaciones el usuario tiene la oportunidad de observar dos simulaciones ejecutándose en una misma ventana de manera iterativa, con lo cual tiene la oportunidad de obtener sus propias conclusiones en cuanto a ventajas y desventajas el uno con respecto al otro.

Pensando en los usuarios finales, las simulaciones están diseñadas de tal forma que permite saber lo que está ocurriendo en cualquier instante, a través de las gráficas, tablas y los distintos mensajes; no obstante, la mayoría de las opciones está dotada con un Visor de Log, en el cual se almacena todo lo que ha acontecido hasta ese instante en la simulación, permitiendo así el mayor entendimiento de los mecanismos de gestión de un sistema operativo.

Además de lo anterior, se suma el hecho de poder detener la simulación en cualquier momento que el usuario lo desee, esto conlleva a saber que está sucediendo exactamente en un determinado instante de tiempo.

### **3.2 ACERCA DEL LENGUAJE**

En el desarrollo de este proyecto se utilizó el lenguaje de programación **Delphi**, el cual se ha popularizado debido a las ventajas que ofrece para el

diseño de interfaces y su forma de programación basada en el lenguaje *Pascal*.

**Delphi** es un lenguaje de programación orientado a objetos con todas las ventajas de su tipo, entre ellas, dos que fueron muy útiles para este proyecto:

Construir visualmente la interfaz de usuario.

Reutilizar los objetos, evitando de esta forma volver a diseñarlos.

**Delphi** posee una clase abstracta para todos los componentes llamada **TControl** que permite manipularlos en tiempo de ejecución, y que es de gran utilidad cuando se desea manipular ciertas propiedades de un conjunto de objetos, esta clase fue de gran utilidad para la realización de este software ya que por las condiciones aleatorias de las simulaciones, la mayoría de los objetos usados varían en tiempo de ejecución.

Otro punto crítico para el diseño del software fue el uso de objetos controladores de tiempo como son los **Timers** ofrecidos por este lenguaje; dichos **timers** permiten retardar la simulación para hacer un seguimiento y una mejor apreciación de la misma.

Además de lo anterior, se suma la ventaja que ofrece Delphi de permitir la creación de objetos visuales en tiempo de ejecución, disminuyendo la

sobrecarga producida al colocar de manera previa objetos no utilizados en un momento dado.

### **3.3 DESTINATARIOS**

Esta herramienta de simulación vá dirigida a todas las personas interesadas en el estudio del área de los sistemas operativos con los conocimientos básicos relacionados con dicha área.

### **3.4 PRESENTACION**

Esta herramienta es un gran apoyo al área de los sistemas operativos y de los diferentes mecanismos que éste utiliza para llevar a cabo sus tareas, ya que contiene las simulaciones de cuatro de los campos más importantes en que se basan prácticamente la mayoría de los sistemas operativos actuales como son: Gestión de la Memoria, gestión de las operaciones de E/S, Gestión de archivos, además, por su importancia, el disco es tratado en un modulo particular.

El programa contiene una entrada principal en la cual el usuario podrá escoger según su interés cuál de los cuatro campos anteriormente mencionados desea simular, con el fin, ya sea de afianzar conocimientos,

despejar dudas o cualquier otro aspecto que motive el uso de esta herramienta.

Además de esto existe un proyecto que complementa este, *Simulación de los mecanismos de administración y control de un sistema operativo parte II*, el cual abarca otros aspectos importantes de los sistemas operativos.

### **3.5 AREAS DEL CONTENIDO**

Este proyecto busca enriquecer y fortalecer el nivel académico en la Corporación Universitaria Tecnológica de Bolívar y de todas aquellas personas y/o instituciones interesadas en el estudio del diseño e implementación de los sistemas operativos.

### **3.6 COMPLEJIDAD**

Con el fin de obtener el mejor provecho y el máximo rendimiento de esta herramienta, es importante que los usuarios tengan conocimientos básicos del área de los sistemas operativos y sus diferentes mecanismos de gestión, ya que en ello está basada esta simulación.

### **3.7 CONTROL INSTRUCCIONAL**

Este es un software totalmente gráfico que puede ser maniobrado por parte del usuario de manera práctica y segura. El software consta de una serie de ventanas que lo llevan en su respectivo orden de una simulación a otra, es decir, desde la ventana principal se tiene la oportunidad de escoger cual simulación se desea ejecutar, luego se despliega una nueva ventana en la cual se puede simular la opción escogida.

A través de controles (botones) que poseen las diferentes simulaciones, se puede, bien sea, dar inicio a la simulación, detenerla y/o continuar (esto en caso de que se haya detenido la simulación a través de una pausa).

Cada simulación, al igual que el proyecto en general, puede ser abandonado en cualquier momento pero los datos obtenidos hasta el momento no podrán ser recuperados.

### **3.8 AYUDAS**

Cada una de las gestiones simuladas en este software cuenta con una ventana principal, la cual posee un control que lleva al usuario a una

nueva ventana, en la que obtiene una breve descripción teórica de los temas tratados en dicha simulación.

## **4. DESCRIPCIÓN DE ASPECTOS DE EJECUCIÓN DEL SOFTWARE**

### **4.1 ASPECTOS GENERALES**

- La aplicación provee una interfaz amigable que facilita el manejo por parte del usuario.
- Las ventanas principales de cada simulación poseen un control que lleva al usuario a una breve descripción de los conceptos básicos que enmarca dicha simulación.
- El usuario puede detener cualquier simulación en el momento que lo desee, activándose automáticamente un control para que de manera análoga pueda reanudarla.
- El usuario puede leer lo que ha acontecido hasta ese instante en la simulación que esté ejecutando a través de un Visor de Log, el cual guarda dichos acontecimientos hasta que se decida ejecutar una nueva simulación o se abandone, bien sea esa ventana o el proyecto.

- A través de mensajes se muestra al usuario lo que está ocurriendo en la simulación en curso.
- El manual del usuario describe aspectos generales de la ejecución del software de forma detallada.

## **4.2 ASPECTOS PARTICULARES**

### **4.2.1 Asignación de Memoria**

- Se muestra una forma principal a través de la cual se puede acceder a las simulaciones de los diferentes mecanismos de asignación de memoria.
- Cada simulación se muestra en una ventana la cual posee diferentes marcos de páginas en los que se encuentran las diferentes tablas y el mapa de memoria respectivamente.
- Cada simulación posee un mapa o mapas de memoria y tablas que representan las estructuras de datos utilizadas por el mecanismo o mecanismos.
- Asignación de memoria es el único campo que le ofrece al usuario la posibilidad de interactuar con el software permitiéndole tomar uno de dos caminos, como es el caso del mecanismo de particiones dinámicas,

donde se pregunta al usuario, a través de una ventana si desea o no compactar.

#### **4.2.2 Algunas Generalidades de Entrada/Salida**

- La simulación se lleva a cabo en una sola ventana con dos marcos de página.
- El primer marco de página muestra:
  - La capa del gestor de operaciones de E/S se encuentra la petición.
  - Gráficamente el dispositivo que generó la petición de E/S.
  - El identificador del dispositivo que generó la petición de E/S
  - El tipo de operación (Entrada o Salida).
- El segundo marco de página muestra gráficamente en que parte del proceso de E/S se encuentra la petición.

#### **4.2.3 Sistema de Archivos**

- La simulación se lleva a cabo en una sola ventana con dos marcos de página cada uno de los cuales contiene a su vez dos submarcos.
- El primer marco de página muestra:
  - Un submarco con la organización de los archivos en forma plana, y el otro con la organización de los archivos en forma jerárquica.

- El segundo marco de página muestra:
  - Un primer submarco alusivo al sistema operativo UNIX, muestra la información del i-nodo del archivo que está siendo accesado en ese instante.
  - Un segundo submarco alusivo al sistema operativo DOS, muestra la asignación de archivos en disco.

#### **4.2.4 Administración del disco**

- La simulación se lleva a cabo en una sola ventana con cinco marcos de página.
- El primer marco de página muestra:
  - Las solicitudes que serán atendidas.
  - Animación de un disco.
  - La diferencia de cilindros entre los diferentes algoritmos de planificación de acceso a disco.
- Los cuatro marco de página restantes muestran:
  - En escala mediante un plano cartesiano los desplazamientos del brazo del disco entre los diferentes cilindros en los que se encuentran las peticiones.

### 4.3 MENSAJES EN LAS SIMULACIONES

El software contiene los siguientes mensajes dirigidos a los usuarios:

- **Debe elegir una o ambas opciones.** Algunas simulaciones de asignación de memoria ofrecen al usuario la posibilidad de elegir entre una política u otra para ejecutar el mecanismo, este mensaje aparece en el caso en que sin haber elegido ninguna de las opciones se presiona el botón simular.
- **Ha finalizado la Simulación.** Indica al usuario que se han atendido todas las solicitudes o peticiones (memoria, operaciones sobre archivo, disco, entrada/salida) y que se puede dar inicio a una nueva simulación.

## 5. DESCRIPCIÓN DE ASPECTOS DEL DISEÑO DEL SOFTWARE

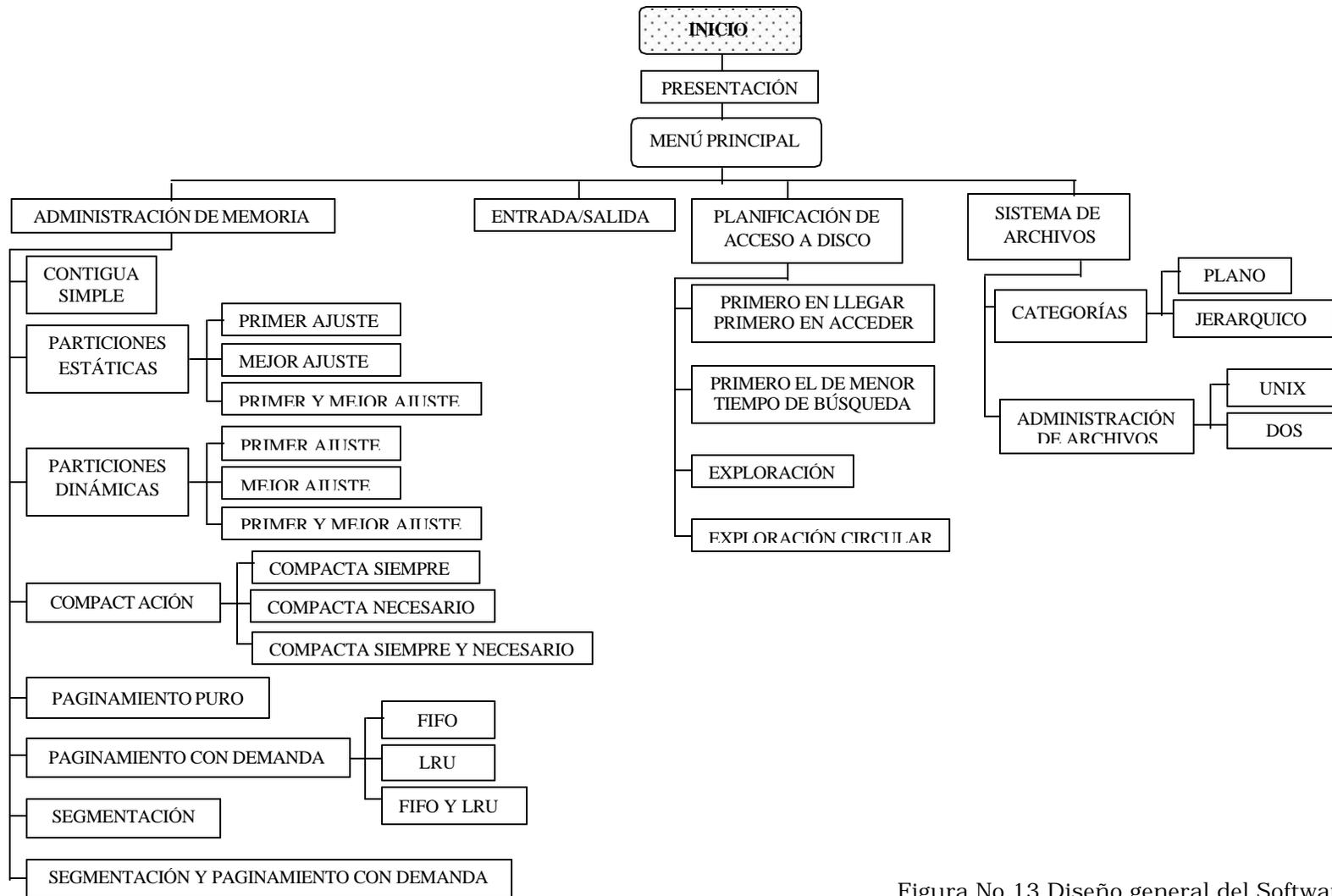


Figura No.13 Diseño general del Software

## 5.1 ASPECTOS GENERALES

- La simulación no es tiempo real.
- El diseño general del software está basado en reguladores de tiempo (objetos **timers**) que, junto con algunos controles permiten detener la simulación en cualquier instante y su posterior reanudación a partir del punto en el cual fue detenida.
- Las características bajo las cuales se ejecutan las simulaciones son aleatorias.

El diseño esta realizado de tal forma que cada vez que se inicie una nueva simulación los parámetros son diferentes, estando en capacidad de tomar decisiones basadas en las nuevas especificaciones aleatorias que se le presenten.

- Se diseñaron dos unidades con especificaciones genéricas, dichas unidades son utilizadas según los requerimientos por cualquier otra unidad del software, logrando con ello eficiencia en el código y en tiempo de ejecución.

## 5.2 ASPECTOS PARTICULARES

A continuación se mencionan aspectos del diseño que son propios de cada simulación y también supuestos o parámetros que se establecieron para llevar a cabo las simulaciones.

**5.2.1 Asignación de Memoria.** La asignación de memoria comprende dieciséis simulaciones de las cuales cabe destacar los siguientes aspectos:

- En el mecanismo de asignación mediante particiones estáticas cuando se activa la ventana se crean las particiones y permanecen iguales hasta que se abandona la simulación. Para la asignación de las particiones se implementaron los algoritmos *Primer ajuste* y *mejor ajuste*. El diseño de la simulación que contiene las dos políticas controla que los procesos sean los mismos y permanezcan el mismo tiempo en memoria, de manera que sólo varía la forma en que son atendidos de acuerdo a las características del algoritmo, y se puedan establecer diferencias.
- Al igual que el mecanismo de asignación mediante particiones estáticas, el mecanismo de asignación mediante particiones dinámicas comprende tres simulaciones y la implementación de los algoritmos

*Primer ajuste y mejor ajuste.* Ofreciendo además la opción de compactarla memoria siempre y cuando haya espacio disponible para alojar el proceso pero dicho espacio no esté contiguo.

- En el diseño de la simulación del mecanismo de compactación se implementaron tanto el criterio de compactar la memoria siempre como el de compactar la memoria cuando sea necesario.

La compactación se realiza a través del siguiente algoritmo recursivo.

**Compacta**(indice, direccion, num\_part, inicio: entero, tabla de particiones);

**inicio**

**si** indice<=num\_part **entonces**

**inicio**

**si** tabla de particiones[indice]=libre **entonces**

**inicio**

tabla de particiones [inicio].direccion=direccion;

tabla de particiones [inicio].tamano= tabla de particiones[indice].tamano;

tabla de particiones [inicio]. segmento= tabla de particiones [indice].segmento;

tabla de particiones [inicio]. libre= tabla de particiones [indice].libre;

direccion= direccion+ tabla de particiones[indice].tamano;

inicio=inicio+1;

indice=indice+1;

compacta(inicio, indice, direccion, num\_part, tabla de particiones);

**fin-si**

**sino**

**inicio**

indice= indice+1;

compacta(inicio, indice, direccion, num\_part, tabla de particiones);

**fin-si;**

**fin-si;**

**fin;**

- Con el objetivo de mostrar la característica de asignación no contigua de las paginas de los procesos, en el diseño de asignación mediante

- paginamiento puro se implementó una lista con las direcciones de memoria en forma no contigua.
- El diseño del mecanismo de asignación mediante paginamiento con demanda; quien es el primero de los mecanismos de asignación de memoria virtual aloja todos los procesos solicitantes en disco y son alojadas en memoria únicamente aluella que fFtán siendo referenciadas- La elección de la paginaD a remover de memoria se lleva a cabo a través de los algoritmos FIFO y LRU, cada uđÀ de ellos simulados deYmanera' independiente y una tercerS simulación don4e se muestran los dos criterios.
- El mecanismo de asignación mediante segmentación tiene una variación en la asignación de memoria; es decir se alojan todos los segmentos de los procesos solicitantes en memoria mientras haya espacio; de lo contrario son alojados en disco. Debido a que cuando un proceso finaliza el espacio que ocupaban sus segmentos en memoria queda libre para alojar otros segmentos; la asignación de estos segmentos se realiza mediante el algoritmo mejor ajuste, y en caso de que haya espacio disponible no contiguo, ofrece la opción de compactar.
- El mecanismo de asignación mediante segmentación y paginamiento con demanda está diseñado para que solo muestre al usuario las tablas del proceso que está siendo referenciado en ese momento y la forma en que se encuentra el disco en cuanto al porcentaje ocupado por cada proceso.

### **5.2.1.1 Estructuras de datos relevantes.**

#### **1. Estructura de datos: Solicitantes**

**Descripción:** Define los procesos que serán atendidos por el sistema operativo, dependiendo del mecanismo de asignación de memoria usado.

#### **Definición de los campos**

##### **1.1 Nombre del campo: Tamano**

**Descripción:** Define el espacio que ocupará el proceso solicitante en memoria (KB).

**Tipo:** Entero

##### **1.2 Nombre del campo: Atendida**

**Descripción:** Indica si la solicitud ha sido alojada en memoria.

**Tipo:** Boolean

##### **1.3 Nombre del campo: Tiempo**

**Descripción:** Indica la cantidad de tiempo que permanecerá el proceso alojado en Memoria.

**Tipo:** Byte

##### **1.4 Nombre del campo: Num\_Seg**

**Descripción:** Indica el número de páginas o segmentos del proceso.

**Tipo:** Byte

### **1.5 Nombre del campo:** Segmentos

**Descripción:** Utilizado por los mecanismos de asignación de memoria Paginamiento Puro (para indicar que fracción del proceso se encuentra en determinada página) y Segmentación (para indicar el tamaño del segmento).

**Tipo:** Arreglo de enteros

## **2. Estructura de datos:** Particion

**Descripción:** Define las características (dirección, tamaño, turno, libre, análoga y segmento) y el estado de las diferentes partes del mapa de memoria.

### **Definición de los campos**

#### **2.1 Nombre del campo:** Dirección

**Descripción:** Define la dirección en memoria de una partición(K).

**Tipo:** Entero

#### **2.2 Nombre del campo:** Tamano

**Descripción:** Indica el tamaño de la partición en el mapa de memoria (KB).

**Tipo:** Entero

### **2.3 Nombre del campo:** Turno

**Descripción:** Indica el número que le corresponde al proceso según su orden de llegada.

**Tipo:** Byte

### **2.4 Nombre del campo:** Libre

**Descripción:** Determina cuando un proceso liberará la porción de memoria que ocupa.

**Tipo:** Byte

### **2.5 Nombre del campo:** Análoga

**Descripción:** Utilizado en las simulaciones que manejan más de un mecanismo de asignación de memoria, para guardar el número de la partición en la que se alojó el proceso o parte del proceso mediante el mecanismo contrario.

**Tipo:** Byte

### **2.6 Nombre del campo:** Segmento

**Descripción:** Determina el número de segmento o de página de un proceso que se encuentra alojado en esa partición.

**Tipo:** Byte

### **3. Estructura de datos: Puntero**

**Descripción:** Define las particiones de memoria libres y además es usado en el mecanismo de asignación de particiones dinámicas para describir el proceso en memoria.

### **Definición de los campos**

#### **3.1 Nombre del campo: Tamano**

**Descripción:** Define el tamaño de las particiones libres en memoria (KB) y para Particiones Dinámicas define el tamaño del proceso.

**Tipo:** Entero

#### **3.2 Nombre del campo: Dirección**

**Descripción:** Indica la dirección de las particiones libres en memoria.

**Tipo:** Entero

#### **3.3 Nombre del campo: Ant**

**Descripción:** Indica el nodo anterior en la lista de particiones libres.

**Tipo:** Proc\_Puntero

### **3.4 Nombre del campo:** Sig

**Descripción:** Indica el nodo siguiente en la lista de particiones libres

**Tipo:** Proc\_Puntero

### **3.5 Nombre del campo:** Num\_Trab

**Descripción:** Indica el número que le corresponde al proceso según su orden de llegada.

**Tipo:** Byte

## **4. Estructura de datos: Direcciones**

**Descripción:** Usada para generar direcciones aleatorias en las que se alojarán las páginas de un proceso.

### **Definición de los campos**

#### **4.1 Nombre del campo:** Dirección

**Descripción:** Contiene la dirección del proceso o parte de éste que se encuentre en memoria (K).

**Tipo:** Entero

#### **4.2 Nombre del campo:** Bloque

**Descripción:** Indica el número del bloque al cual corresponde la dirección mencionada en el campo anterior.

**Tipo:** Byte

#### **4.3 Nombre del campo:** Segmento

**Descripción:** Indica el número del segmento o la página de un proceso que se encuentra alojado en memoria.

**Tipo:** Byte

#### **4.4 Nombre del campo:** Libre

**Descripción:** Indica si una dirección de memoria está libre u ocupada.

**Tipo:** Byte

### **5. Estructura de datos: Tab\_MP**

**Descripción:** Arreglo de registros cuyos campos son Página, Bloque y Estado, los cuales se utilizan para manejar las tablas de mapa de página y las tablas de mapa de segmento.

#### **Definición de los campos**

##### **5.1 Nombre del campo:** Página

**Descripción:** Contiene el número de la página o segmento de un proceso que se encuentre en memoria.

**Tipo:** Byte

### **5.2 Nombre del campo:** Bloque

**Descripción:** Indica el número del bloque en memoria en el cual se encuentra ubicada determinada página o segmento de un proceso.

**Tipo:** Byte

### **5.3 Nombre del campo:** Estado

**Descripción:** Indica si la página o segmento de un proceso se encuentra alojado en memoria o en disco.

**Tipo:** String

## **6. Estructura de datos: Tab\_MA**

**Descripción:** Arreglo de registros cuyos campos son Segmento, Bloque y Dirección, los cuales se utilizan para manejar las tablas de mapa de Archivos.

### **Definición de los campos**

#### **6.1 Nombre del campo:** Segmento

**Descripción:** Contiene el número de la página o segmento de un proceso que se encuentre en disco.

**Tipo:** Byte

### **6.2 Nombre del campo:** Bloque

**Descripción:** Indica el número del bloque en memoria en el cual se encuentra ubicada determinada página o segmento de un proceso.

**Tipo:** Byte

### **6.3 Nombre del campo:** Dirección

**Descripción:** Indica la dirección de una página o segmento en disco.

**Tipo:** Entero

## **7. Estructura de datos: Apuntador**

**Descripción:** Describe los procesos que solicitan memoria.

### **Definición de los campos**

#### **7.1 Nombre del campo:** Num\_Trab

**Descripción:** Indica el número que le corresponde al proceso según su orden de llegada.

**Tipo:** Byte

**7.2 Nombre del campo:** Tamano

**Descripción:** Indica el tamaño del proceso (KB).

**Tipo:** Entero

**7.3 Nombre del campo:** Estado

**Descripción:** Usado por el mecanismo de asignación Paginamiento y Segmentación con Demanda para indicar si un segmento está en disco o en memoria.

**Tipo:** String

**7.4 Nombre del campo:** Dir\_Tmp

**Descripción:** Indica la dirección de la tabla de mapa de página.

**Tipo:** Entero

**7.5 Nombre del campo:** Dir\_Tma

**Descripción:** Indica la dirección de la tabla de mapa de archivo.

**Tipo:** Entero

**7.6 Nombre del campo:** TMP

**Descripción:** Arreglo de registros que se utilizan para manejar las tablas de mapa de página y las tablas de mapa de segmento.

**Tipo:** Tab\_MP

**7.7 Nombre del campo:** TMA

**Descripción:** Arreglo de registros que se utilizan para manejar las tablas de mapa de Archivos.

**Tipo:** Tab\_MA

**7.8 Nombre del campo:** Num\_Pag

**Descripción:** Indica el número de páginas o segmentos que tiene el proceso.

**Tipo:** Byte

**7.9 Nombre del campo:** Libre

**Descripción:** Indica la cantidad de tiempo que permanecerá el proceso alojado en memoria.

**Tipo:** Byte

**7.10 Nombre del campo:** Ant

**Descripción:** Indica el nodo anterior en la lista de procesos solicitantes.

**Tipo:** Apuntador

**7.11. Nombre del campo:** Sig

**Descripción:** Indica el nodo siguiente en la lista de procesos solicitantes.

**Tipo:** Apuntador

## **8. Estructura de datos: Bloques**

**Descripción:** Arreglo de registros cuyos campos son Estado, Página, Segmento y Trabajo; se utiliza para describir las particiones de memoria en los mecanismos de asignación de Memoria Virtual.

### **Definición de los campos**

#### **8.1 Nombre del campo:** Estado

**Descripción:** Indica si una partición de memoria está libre u ocupada.

**Tipo:** String

#### **8.2 Nombre del campo:** Página

**Descripción:** Indica el número de página o de segmento del proceso que se encuentra en determinada partición de memoria.

**Tipo:** Byte

#### **8.3 Nombre del campo:** Segmento

**Descripción:** Usado por el mecanismo de asignación Paginamiento y Segmentación con Demanda para indicar a que segmento del proceso pertenece la página que se encuentra alojada en esa partición de memoria.

**Tipo:** byte

#### **8.4 Nombre del campo:** Trabajo

**Descripción:** Indica el número que le corresponde al proceso según su orden de llegada.

**Tipo:** Byte

### **8.5 Nombre del campo:** Tamano

**Descripción:** Indica tamaño de la partición de memoria.

**Tipo:** Byte

### **8.6 Nombre del campo:** Elección

**Descripción:** Usado por el mecanismo de asignación Paginamiento con demanda. En FIFO para ilustrar el orden de llegada de una página y en LRU para ilustrar la cantidad de veces que ha sido referenciada una página para el momento en que haya que remover alguna.

**Tipo:** Byte

## **9. Estructura de datos: lista\_Ref**

**Descripción:** Describe las páginas o segmentos o páginas de determinado segmento que se encuentran en memoria.

### **Definición de los campos**

#### **9.1 Nombre del campo:** Num\_Pag

**Descripción:** Indica el número de una página de determinado proceso que se encuentra en memoria.

**Tipo:** Byte

### **9.2 Nombre del campo:** Num\_Seg

**Descripción:** Indica el número del segmento al cual pertenece determinada página de un proceso.

**Tipo:** Byte

### **9.3 Nombre del campo:** Trabajo

**Descripción:** Indica el proceso al cual pertenece una página o segmento o página de un segmento.

**Tipo:** Byte

### **9.4 Nombre del campo:** Remover

**Descripción:** Usada por el mecanismo de asignación FIFO para indicar el orden de llegada de una página a memoria.

**Tipo:** Byte

### **9.5. Nombre del campo:** Cont\_Ref

**Descripción:** Indica el número de veces que ha sido referenciada una página.

**Tipo:** Byte

### **9.6 Nombre del campo:** Ant

**Descripción:** Indica el nodo anterior en la lista de páginas.

**Tipo:** Lista\_Ref

### **9.7 Nombre del campo:** Sig.

**Descripción:** Indica el nodo siguiente en la lista de páginas.

**Tipo:** Lista\_Ref

## **10. Estructura de datos: lista**

**Descripción:** Describe los procesos que solicitan memoria mediante el mecanismo de asignación: Segmentación y Paginamiento con Demanda.

### **Definición de los campos**

#### **10.1 Nombre del campo:** Num\_Trab

**Descripción:** Indica el número que le corresponde al proceso según su orden de llegada.

**Tipo:** Byte

#### **10.2 Nombre del campo:** Num\_Seg

**Descripción:** Indica el número del segmento al cual pertenece determinada página de un proceso.

**Tipo:** Byte

### **10.3 Nombre del campo:** Direccion

**Descripción:** Indica la dirección de una página o segmento en disco.

**Tipo:** Entero

### **10.4 Nombre del campo:** PtrL

**Descripción:** Indica el primer nodo en la lista de procesos solicitantes

**Tipo:** Apuntador

### **10.5 Nombre del campo:** Tamano

**Descripción:** Indica el tamaño del proceso (KB).

**Tipo:** Entero

### **10.6 Nombre del campo:** Libre

**Descripción:** Indica la cantidad de tiempo que permanecerá el proceso alojado en memoria.

**Tipo:** Byte

### **10.7 Nombre del campo:** Ant

**Descripción:** Indica el nodo anterior en la lista de procesos solicitantes.

**Tipo:** Lista

### **10.8 Nombre del campo:** Sig

**Descripción:** Indica el nodo siguiente en la lista de procesos solicitantes.

**Tipo:** Lista

## **11 Estructura de datos: Tipo\_registro**

**Descripción:** Registro que indica las páginas de determinado segmento de un proceso para el mecanismo de asignación: Segmentación y Paginamiento con Demanda.

### **Definición de los campos**

#### **11.1 Nombre del campo: Tamano**

**Descripción:** Indica el de una página (por defecto, tamaño de páginas=3).

**Tipo:** Entero

#### **11.2 Nombre del campo: Paginas**

**Descripción:** Arreglo de enteros que contiene las páginas de determinado segmento de un proceso.

**Tipo:** Entero

#### **11.3 Nombre del campo: Num\_pag**

**Descripción:** Indica el número de página de determinado segmento de un proceso.

**Tipo:** Byte

## **12 Estructura de datos: Tipo\_proc**

**Descripción:** Define los procesos que serán atendidos por el sistema operativo en el mecanismo de asignación de memoria: Segmentación y Paginamiento con Demanda.

### **Definición de los campos**

#### **12.1 Nombre del campo: Tamano**

**Descripción:** Define el espacio que ocupará el proceso solicitante en memoria (KB).

**Tipo:** Entero

#### **12.2 Nombre del campo: Atendida**

**Descripción:** Indica si la solicitud ha sido alojada en memoria.

**Tipo:** Boolean

#### **12.3 Nombre del campo: Tiempo**

**Descripción:** Indica la cantidad de tiempo que permanecerá el proceso alojado en Memoria.

**Tipo:** Byte

#### **12.4 Nombre del campo: Num\_Seg**

**Descripción:** Indica el número de segmento de un proceso.

**Tipo:** Byte

## **12.5 Nombre del campo:** Segmentos

**Descripción:** Arreglo de enteros que contiene los segmentos de un proceso

**Tipo:** Arreglo de enteros(Tipo\_registro)

**5.2.1.2 Supuestos.** Existe un espacio reservado de memoria que es señalado en los mapas de memoria como **S.O** para indicar que hay una parte de la memoria que no está disponible para los procesos solicitantes, y en la cual reside el sistema operativo.

- La unidad de medida utilizada para definir el espacio que ocupan los procesos es el Kilobyte.
- Se estableció un espacio de memoria reservado de 5K y otro disponible para el usuario de 27K
- En las simulaciones de los mecanismos de asignación de memoria se asume que el área de memoria disponible para el usuario se encuentra libre antes de que la simulación genere la primera solicitud de memoria.
- Los tamaños de las paginas en las simulaciones que asignan memoria virtual es 3K.

**5.2.2 Algunas generalidades de Entrada/Salida.** Esta simulación muestra el proceso en que incurre cualquier petición (Figura No.14), ya sea de entrada o de salida, a la vez, muestra el identificador del dispositivo que realizó la petición y si es orientado a bloque o a caracter. Las imágenes que representan en que estado del proceso de Entrada/Salida se encuentra la petición, son cargadas desde disco cuando son requeridas, buscando con ello mejorar el desempeño.

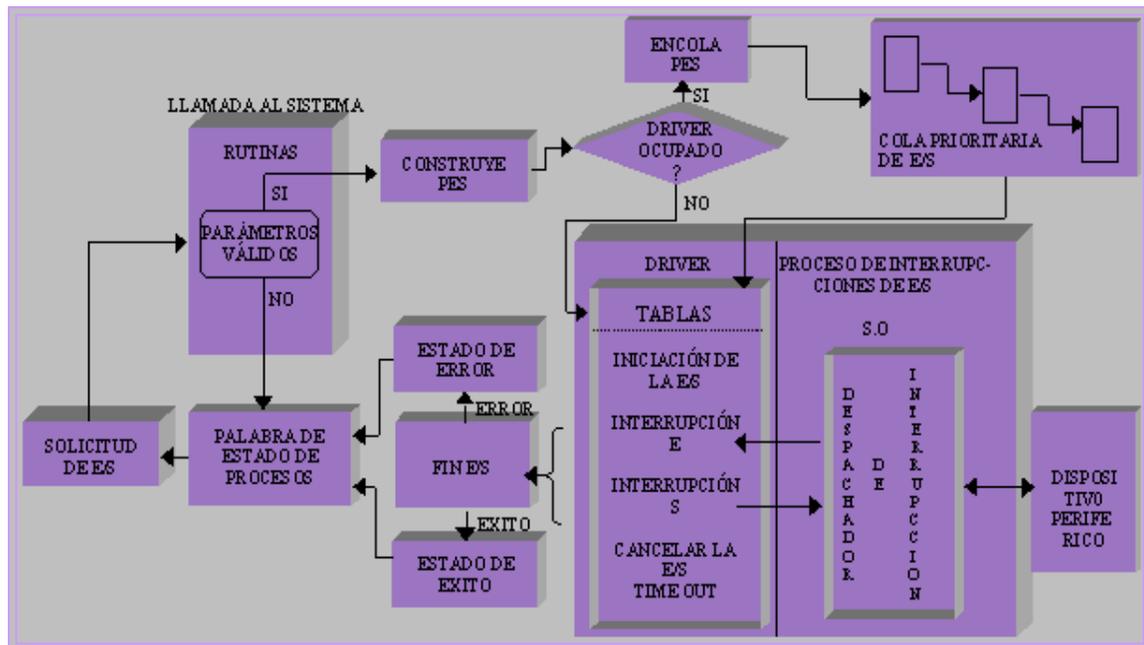


Figura No.14 Proceso de Entrada/Salida

### 5.2.2.1 Estructuras de datos relevantes

#### 1. Estructura de datos: Solicitud

**Descripción:** Contiene la información de cada solicitud

#### Definición de los campos

##### 1.1 Nombre del campo: Parametros

**Descripción:** Indica si la solicitud tiene los parámetros válidos o no

**Tipo:** String

##### 1.2 Nombre del campo: Identificador

**Descripción:** Indica el dispositivo que realizó la petición de entrada o de salida.

**Tipo:** Byte

##### 1.3 Nombre del campo: DriverOc

**Descripción:** Indica si el driver está libre o está ocupado

**Tipo:** String

##### 1.4 Nombre del campo: Tipo

**Descripción:** Indica si la solicitud es de entrada o de salida

**Tipo:** String

**1.5 Nombre del campo:** Estado

**Descripción:** Indica si la solicitud ha sido o no atendida

**Tipo:** Boolean

**5.2.2.2 Supuestos.** Se asume que las solicitudes de entrada/salida son realizadas por los siguientes dispositivos: disco, impresora, monitor, scanner y teclado.

**5.2.3 Sistema de archivos.** Esta simulación contempla la clasificación de los sistemas de archivos en las categorías plano y jerárquico, Además, la forma en que son asignados los archivos en los sistemas operativos UNIX y DOS. Se trabaja con diez archivos cuyo tamaño es aleatorio cada vez que se simula. Las operaciones que se realizan sobre los archivos son de escritura, lectura y/o modificación, dichas operaciones son aleatorias. Existen diez usuarios, un administrador y los demás están organizados en tres grupos, cada uno con tres usuarios.

En UNIX se muestra la información del archivo sobre el cual se está realizando una operación, es decir: tamaño, número de bloque, número de enlaces, el dueño del archivo, el grupo al cual pertenece, hora del último

acceso y de la última modificación, además, se muestra el tipo de enlace del archivo, ya sea simplemente indirecto o doblemente indirecto.

En DOS se muestra la forma en que son asignados los archivos en el disco y las operaciones que se realicen sobre ellos.

### **5.2.3.1 Estructuras de datos relevantes**

#### **1. Estructura de datos: Inf\_Arch**

**Descripción:** Contiene las características de un archivo.

#### **Definición de los campos**

##### **1.1 Nombre del campo:** Tipo

**Descripción:** Indica el tipo de archivo (Excel, Word, Tipo Log)

**Tipo:** Cadena

##### **1.2 Nombre del campo:** Enlaces

**Descripción:** Indica el número de enlaces que tiene el í-nodo correspondiente a ese archivo.

**Tipo:** Entero.

##### **1.3 Nombre del campo:** Bloques

**Descripción:** Indica el número de bloque que tiene el í-nodo correspondiente a ese archivo.

**Tipo:** Entero.

#### **1.4 Nombre del campo: UID**

**Descripción:** Identificador del dueño archivo

**Tipo:** Entero.

#### **1.5 Nombre del campo: GID**

**Descripción:** Identificador del grupo al cual pertenece el dueño del archivo.

**Tipo:** Entero.

#### **1.6 Nombre del campo: Tamano**

**Descripción:** Tamaño del archivo en Kilobytes

**Tipo:** Entero.

#### **1.7 Nombre del campo: TUA**

**Descripción:** Hora del ultimo acceso al archivo.

**Tipo:** Time.

#### **1.8 Nombre del campo: TUE**

**Descripción:** Hora de la última escritura en el archivo

**Tipo:** Time

### **1.9 Nombre del campo:** TUC

**Descripción:** Hora del último cambio en el archivo

**Tipo:** Time.

### **1.10 Nombre del campo:** Num\_bloque

**Descripción:** Cantidad de bloques de datos no contiguos del archivo que hay en el disco.

**Tipo:** Entero.

### **1.11 Nombre del campo:** Bloques\_usados

**Descripción:** Indica que bloques del disco usa cada archivo

**Tipo:** Entero.

## **2. Estructura de datos: Reg\_Usuarios**

**Descripción:** Contiene la información de cada usuario

### **Definición de los campos**

#### **2.1 Nombre del campo:** Nombre

**Descripción:** Nombre del usuario

**Tipo:** Cadena

## **2.2 Nombre del campo:** UID

**Descripción:** Nombre del usuario

**Tipo:** Cadena

**5.2.3.2 Supuestos.** Los nombres de los archivos al igual que la estructura jerárquica de directorios que estos conforman es la misma en todas las simulaciones. Este supuesto permite referirse a los archivos por nombre propio y dar una mayor claridad al usuario, sin establecer previamente los tamaños de los archivos como tampoco las operaciones realizadas sobre los archivos.

**5.2.4 Administración del disco.** El diseño de esta simulación permite al usuario visualizar las solicitudes que serán atendidas, las cuales están representada por el número de pista donde se encuentra el bloque a leer o escribir. El número de pista en el cual se encuentra el brazo del disco al comienzo de la simulación es aleatorio, al igual que las solicitudes a atender.

Para la simulación de la planificación de los accesos a disco se implementaron los algoritmos *Primero en llegar primero en acceder*, *Primero*

*el de menor tiempo de búsqueda, Exploración y Exploración circular*; estos algoritmos atienden las solicitudes 'al mismo tiempo' según cada criterio y se calculan las diferencias de cilindros para cada algoritmo.

#### **5.2.4.1 Estructuras de datos relevantes**

##### **1. Estructura de datos:** Tipo\_Disco

**Descripción:** Contiene la información de cada solicitud

#### **Definición de los campos**

##### **1.1 Nombre del campo:** pista

**Descripción:** Número de pista de donde se lee o escribe el bloque de datos

**Tipo:** Entero

##### **1.2 Nombre del campo:** Atendida

**Descripción:** Indica si la solicitud ha sido o no atendida

**Tipo:** Boolean

**5.2.4.2 Supuestos.** Antes de que los algoritmos de administración del disco atiendan su primera petición se supone que la cabeza lectora del

brazo del disco se encuentra en una pista de inicio generada por la simulación.

## **RECOMENDACIONES**

Luego del estudio e investigación preliminar y realización de este proyecto, se recomienda:

En la parte de memoria, es importante hacer énfasis en las estadísticas que usan la mayoría de los sistemas operativos para asignar y desasignar las direcciones en memoria real, pues, todos los procesos no tienen el mismo comportamiento en cuanto a sus patrones de referenciación; para llevar dichas estadísticas el administrador de memoria hace uso de bits, de estructuras de datos e incluso de hardware adicional, este conjunto de aspectos de más bajo nivel pueden ser estudiados con el fin de hacerlos más tangibles.

La información se puede tener disponible no sólo cuando se ejecute una aplicación sino en cualquier momento, para ello se hace uso de unidades de almacenamiento a largo plazo como son los archivos. Es recomendable el estudio de la forma de almacenamiento no secuencial en el sistema operativo DOS, como también la forma de controlar la compartición y protección de dichos archivos no sólo para el sistema operativo DOS sino también para UNIX.

Un aspecto que marca una estrecha relación entre el sistema de archivos y el disco, son los métodos de acceso a los archivos; los cuales también pueden ser relacionados en una simulación.

El desarrollo de la tecnología trae consigo la invención de muchos artefactos, y un grupo representativo de estos funcionan asociados a una computadora, de manera que para el sistema operativo son dispositivos que deben ser tenidos en cuenta en la gestión de Entrada/Salida. Una gestión que se hace un poco más compleja y a la cual se debe que hoy por hoy el computador sea indispensable en todo tipo de labores. El manejo de las operaciones de Entrada/Salida además de darse por programas (caso mostrado es este proyecto), también se puede hacer mediante interrupciones, las cuales facilitan la multiplexación del procesador entre varios dispositivos, debido a esta última consideración, se invita a las personas interesadas a profundizar en el campo de la entrada/salida controlada por interrupciones.

Otro punto a tener en cuenta es que los mecanismos de administración simulados no son los únicos, existen otros que resultan de la variación o combinación de ellos; por lo tanto se recomienda que las personas interesadas en este estudio y que hagan uso de esta herramienta

propongan sus propias variantes las cuales a su vez podrían ser simuladas.

Una opción alterna a este software podría ser la elaboración de un tutor con fines netamente académicos que posea el soporte de una metodología educativa.

Para estudios posteriores sería interesante llevar a cabo el diseño e implementación de un prototipo de sistema operativo.

Para un mejor manejo de la herramienta *Simulación S.O Parte I* el usuario debe tener conocimientos básicos acerca de los temas que éste abarca, además, revisar el contenido del manual del soporte del software, el cual encuentra en este documento como *Anexo A*, en donde se encuentra expuesto

## CONCLUSIONES

En el desarrollo de la investigación e implementación de este proyecto de grado, se pueden extraer las siguientes conclusiones:

- El usuario puede observar a través de una interfaz gráfica el funcionamiento de la memoria en cuanto a la asignación y desasignación de procesos, pues se implementaron los mecanismos de administración de memoria, apoyados en mapas de memoria y tablas que permiten apreciar de manera clara el comportamiento de dichos mecanismos ante todo tipo de peticiones. Además algunos mecanismos de asignación son mostrados simultáneamente de tal forma que se pueden establecer comparaciones, ventajas y desventajas de uno con respecto a otro.
- A través de un diagrama se ilustra de manera general los diferentes estados o pasos por los que transcurre una petición de Entrada/Salida a nivel de software. También se indica la capa de gestión de operaciones de E/S en la cual se encuentra la petición;

permitiéndole al usuario comprender como funciona la entrada / salida en este nivel y la clasificación de los dispositivos de acuerdo al tipo de información que maneja, bloque o carácter.

- A través de un grupo de archivos organizados en forma plana y jerárquica los usuarios pueden establecer las diferencias, ventajas y desventajas entre estas formas de organización, además, se aprecian las diferentes operaciones realizadas sobre dichos archivos, a saber lectura, escritura y modificación; como también la forma en que el sistema operativo UNIX registra los bloques y atributos asociados a cada archivo mediante í-nodos, y la asignación contigua de los archivos en disco en el ámbito del sistema operativo DOS.
- El plano cartesiano permite apreciar distancias o recorridos, esto constituye una ventaja para este proyecto en cuanto a la representación de los desplazamientos del brazo del disco.

Las peticiones de acceso a disco son atendidas según diferentes criterios de programación, colocando a disposición del usuario la diferencia de cilindros o desplazamiento producida por dichos criterios, de forma que se pueden establecer comparaciones.

La simulación muestra además, a través de un sistema de ventanas las características del disco, como son sus partes y la forma en que organiza la información.

- Las simulaciones de los mecanismos de administración y control de un sistema operativo están dotadas de una serie de ventanas, mensajes y gráficos, que facilitan el proceso de enseñanza / aprendizaje, porque a través de estas los usuarios pueden ver representados conceptos que eran poco tangibles permitiéndoles formarse imágenes de la forma en que el sistema operativo realiza sus funciones.
- Los objetivos propuestos se alcanzaron en su totalidad obteniendo como resultado un software de fácil instalación y manejo.



## **BIBLIOGRAFÍA**

ALCALDE, Eduardo MORERA, Juan, PÉREZ, Juan, CAMPANERO, Atanasio. Introducción a Los Sistemas Operativos. España: McGraw-Hill, 1992. P119-147

<http://www.ac.uma.es/academia/educacion/cursos/DisSistOper.html>

<http://www.clubdelphi.com>

<http://www.cs.purdue.edu/homes/dec/essay.dissertation.html>.

<http://www.dia.uned.es/autom/asignaturas/SOI/Temario/tema6.htm>

[http://www.inf.udec.cl/~chernand/sist\\_computacionI/c5.html](http://www.inf.udec.cl/~chernand/sist_computacionI/c5.html)

<http://www.geocities.com/SouthBeach/Castle/4775/cursos/hardware/hardware2.htm>

<http://www-lifia.info.unlp.edu.ar/~fidel/FP/IO.html>

<http://w3.mor.itesm.mx/~fglez/EqPerif/info-curso.html>

LAROUSSE. Diccionario práctico de Sinónimos y Antónimos. Marsella

México D.F: Larousse, 1988. 386 P

Martínez, Lambraño Juan. SN. SL, 1999. 60P

MÁRQUEZ, GARCÍA Francisco Manuel. UNIX Programación Avanzada.

USA: Addison Wesley Iberoamericana, 1994. 484P

MILENKOVIC´ Milan. Sistemas Operativos Conceptos y Diseño. 2ed.

Madrid: McGraw-Hill, 1994. P203-303, P423-465

OSIER, Dan, BATSON, Steve, GROBMAN, Steve. Aprendiendo Delphi 3 en

14 días. México:Prentice hall, 1997. 706P

SUN EDUCATIONAL SERVICES. Independence Through Knowledge-

Solaris 2.X System Administration. USA: Sun Microsystems, 1998. P198-

240

TANENBAUM, Andrew S. Sistemas Operativos Modernos. Naucalpan de

Juárez- México: Prentice hall, 1993. P85-268

**ANEXO A**  
**MANUAL DE SOPORTE DEL SOFTWARE**

***SIMULACIÓN S.O***  
Parte I

***SOFTWARE DE SIMULACIÓN DE LOS MECANISMOS DE  
ADMINISTRACIÓN Y CONTROL DE UN SISTEMA OPERATIVO PARTE I***

***Autores:***

***YENIS ÁLVAREZ JIMÉNEZ***  
***YASMÍN MOYA VILLA***

---

***Ingenieros de Sistemas***

***Director:***  
***JUAN MARTÍNEZ LAMBRAÑO***  
***Maestro en ciencias computacionales***

***CORPORACION UNIVERSITARIA TECNOLOGICA DE BOLIVAR***  
***FACULTAD DE INGENIERÍA DE SISTEMAS***  
***CARTAGENA DE INDIAS D.T. y C.***

**2000**

## CONTENIDO

	Pág.
<b>INTRODUCCIÓN</b>	9
<b>1. GENERALIDADES</b>	10
1.1 REQUERIMIENTOS TÉCNICOS	10
1.2 INSTALACIÓN DE <i>SIMULACIÓN S.O PARTE I</i>	10
1.3 EJECUCIÓN DE <i>SIMULACIÓN S.O PARTE I</i>	15
1.4 CÓMO REMOVER <i>SIMULACIÓN S.O PARTE I</i> DE SU PC	15
1.5 PRESENTACIÓN	16
1.6 SISTEMA DE VENTANAS	17
1.6.1 Controles de una ventana	17
1.6.2 Manejo de Marcos de Página	18
1.6.3 Manejo de Teclas Abreviadas	18
1.6.4 Manejo de la Barra de Botones	19
<b>2. MENÚ PRINCIPAL</b>	23
2.1 APLICACIÓN	23
2.2 MEMORIA	25
2.2.1 Contigua Simple	33
2.2.2 Particiones Estáticas	33
2.2.2.1 Primer ajuste	34

2.2.2.2	Mejor Ajuste	35
2.2.2.3	Primer Ajuste y Mejor Ajuste	36
2.2.3	Particiones Dinámicas	37
2.2.3.1	Primer Ajuste	38
2.2.3.2	Mejor Ajuste	39
2.2.3.3	Primer Ajuste y Mejor Ajuste	40
2.2.4	Compactación	41
2.2.4.1	Compacta Siempre	42
2.2.4.2	Compacta Necesario	42
2.2.4.3	Compacta Siempre y Compacta Necesario	43
2.2.5	Paginamiento Puro	44
2.2.6	Paginamiento Con demanda	45
2.2.6.1	FIFO	46
2.2.6.2	LRU	49
2.2.6.3	FIFO y LRU	51
2.2.7	Segmentación	54
2.2.8	Segmentación y Paginamiento con Demanda	56
2.3	DISCO	61
2.4	ENTRADA / SALIDA	71
2.5	ARCHIVOS	79

## LISTA DE FIGURAS

	<b>Pag.</b>
<b>Figura 1.</b> Pantalla de Presentación	16
<b>Figura 2.</b> Modelo de un Marco de Página	18
<b>Figura 3.</b> Estados de la Barra de Botones	19
<b>Figura 4.</b> Visor de Log.	21
<b>Figura 5.</b> Menú Principal	23
<b>Figura 6.</b> Submenú Aplicación	24
<b>Figura 7.</b> Acerca de Simulación S.O Parte I	24
<b>Figura 8.</b> Administración de Memoria	25
<b>Figura 9.</b> Mapa de memoria	29
<b>Figura 10.</b> Estructura de Datos	29
<b>Figura 11.</b> Finalización	30
<b>Figura 12.</b> Error de Selección	31
<b>Figura 13.</b> Información de Asignación de Memoria	31
<b>Figura 14.</b> Contigua Simple	33
<b>Figura 15.</b> Particiones Estáticas	34
<b>Figura 16.</b> Particiones Estáticas. Primer Ajuste	34
<b>Figura 17.</b> Particiones Estáticas. Mejor Ajuste	35

<b>Figura 18.</b> Primer Ajuste y Mejor Ajuste. Mapas de Memoria (PE)	36
<b>Figura 19.</b> Primer Ajuste y Mejor Ajuste. Tablas de Particiones (PE)	36
<b>Figura 20.</b> Particiones Dinámicas	37
<b>Figura 21.</b> Particiones Dinámicas. Primer Ajuste	38
<b>Figura 22.</b> Particiones Dinámicas. Mejor Ajuste	39
<b>Figura 23.</b> Desea compactar	39
<b>Figura 24.</b> Primer Ajuste y Mejor Ajuste. Mapas de Memoria (PD)	40
<b>Figura 25.</b> Primer Ajuste y Mejor Ajuste. Tablas de Particiones (PD)	40
<b>Figura 26.</b> Compactación	41
<b>Figura 27.</b> Compacta Siempre	42
<b>Figura 28.</b> Compacta Necesario	42
<b>Figura 29.</b> Compacta Siempre y compacta Necesario. Mapas de Memoria.	43
<b>Figura 30.</b> Compacta Siempre y Compacta Necesario. Tablas de Particiones.	43
<b>Figura 31.</b> Paginamiento Puro	44
<b>Figura 32.</b> Paginamiento con Demanda	45
<b>Figura 33.</b> Paginamiento con Demanda. FIFO	46
<b>Figura 34.</b> FIFO. Tablas de Procesos y de Bloques	47
<b>Figura 35.</b> FIFO. Tablas de Mapa de Página y de Archivos	47
<b>Figura 36.</b> Paginamiento con Demanda. LRU	49
<b>Figura 37.</b> LRU. Tablas de Procesos y de Bloques	49
<b>Figura 38.</b> LRU. Tablas de Mapa de Página y de Archivos	50
<b>Figura 39.</b> FIFO y LRU. Mapas de Memoria	51
<b>Figura 40.</b> FIFO y LRU. Tablas de Procesos	52
<b>Figura 41.</b> FIFO y LRU. Tablas de Bloques	52
<b>Figura 42.</b> FIFO y LRU. Tablas de Mapa de Página	53
<b>Figura 43.</b> FIFO y LRU. Tablas de Mapa de Archivos	53
<b>Figura 44.</b> Segmentación. Mapa de Memoria	54
<b>Figura 45.</b> Segmentación. Tablas de Procesos y de Bloques	55
<b>Figura 46.</b> Segmentación. Tablas de Mapa de Segmentos y de Archivos	55
<b>Figura 47.</b> Segmentación y Paginamiento con Demanda (SPCD)	56
<b>Figura 48.</b> SPCD. Porcentaje en Disco	56
<b>Figura 49.</b> SPCD. Descriptor de Segmento y Tabla de Mapa de Página	58
<b>Figura 50.</b> Información de Fallo de Página	59
<b>Figura 51.</b> SPCD. Tablas de Bloques y de Procesos	60
<b>Figura 52.</b> Planificación de los Accesos a Disco	61
<b>Figura 53.</b> Disco. Primero en Llegar, Primero en Acceder	63
<b>Figura 54.</b> Disco. Primero el de Menor Tiempo de Búsqueda	66
<b>Figura 55.</b> Disco. Exploración	66
<b>Figura 56.</b> Disco. Exploración Circular	67

<b>Figura 57.</b> Disco. Información General	68
<b>Figura 58.</b> Disco. Información Tiempos de Acceso	69
<b>Figura 59.</b> Disco. Información General	70
<b>Figura 60.</b> Entrada/Salida (E/S)	71
<b>Figura 61.</b> Gestión de Operaciones de E/S	72
<b>Figura 62.</b> Proceso de E/S	73
<b>Figura 63.</b> Gestión de Operaciones de E/S. 2	74
<b>Figura 64.</b> Proceso de E/S. 2	74
<b>Figura 65.</b> E/S. Parámetros no Válidos	75
<b>Figura 66.</b> E/S. Visor de Log	77
<b>Figura 67.</b> E/S. Información General	78
<b>Figura 68.</b> Archivos. Categoría	79
<b>Figura 69.</b> Archivos. Organización	79
<b>Figura 70.</b> Archivos. Categoría Jerárquica	80
<b>Figura 71.</b> Archivos. Categoría Plana	81
<b>Figura 72.</b> Archivos. Organización en S.O UNIX	82
<b>Figura 73.</b> Archivos. Organización en S.O DOS	84
<b>Figura 74.</b> Archivos. Información General	85
<b>Figura 75.</b> Archivos. Organización en Disco	86

## INTRODUCCIÓN

*SIMULACIÓN S.O PARTE I* es un software totalmente gráfico, basado en la programación orientada a objetos y desarrollado como proyecto de tesis en el lenguaje visual **Delphi**. En él se simulan los mecanismos de administración y control de un Sistema Operativo, debido a la magnitud de éste tema, se encuentra desarrollado en dos fases, de las cuales, *SIMULACIÓN S.O PARTE I*, como su nombre lo indica, abarca la primera fase, la cual contiene la simulación de la asignación de memoria, algunas generalidades de entrada/salida, administración del disco y administración de archivos en los sistemas operativos DOS y UNIX.

*SIMULACIÓN S.O PARTE I* es la herramienta de apoyo que usted necesita para afianzar el conocimiento en los conceptos de diseño e implementación de los sistemas operativos.

El presente manual contiene toda la guía necesaria para manejar y comprender *SIMULACIÓN S.O PARTE I*. Puede consultar el Documento de Tesis, en donde se encuentra todo el soporte teórico necesario para el desarrollo del software.

## 1. GENERALIDADES

### 1.1 REQUERIMIENTOS TÉCNICOS

Para operar normalmente el programa requiere de computadores PC con la siguiente configuración:

- ☞ Procesador Pentium de 100 MHZ o superior.
- ☞ 32 MB o más de memoria RAM
- ☞ 30 MB de espacio en el Disco duro.
- ☞ La configuración del monitor debe estar ajustada a una resolución mínima de 800x600.

### 1.2 INSTALACIÓN DE **SIMULACIÓN S.O PARTE I**

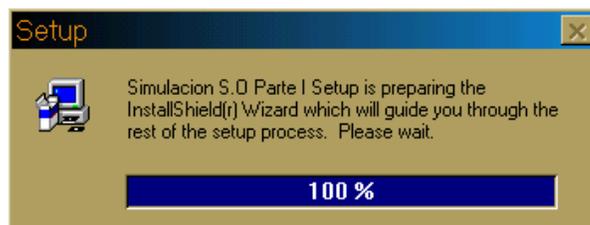
- ☞ Inserte el CD-ROM o el disco 1 del programa en la unidad correspondiente.
- ☞ Haga clic en el botón *Inicio* de Windows.
- ☞ Elija *Configuración* y después haga clic en *Panel de Control*.
- ☞ Haga doble clic en *Agregar o Quitar Programas*.



- ☞ Haga clic en el botón Instalar que se muestra en la figura anterior y haga clic en Siguiente en la posterior ventana.

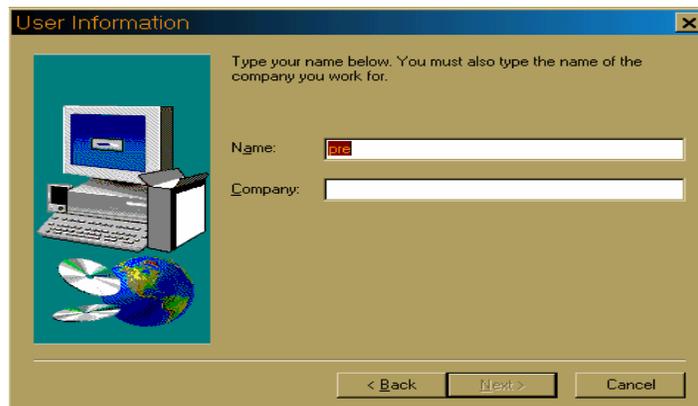


- ☞ Escoja la opción *Setup.exe* y espere hasta que se complete en 100% la operación



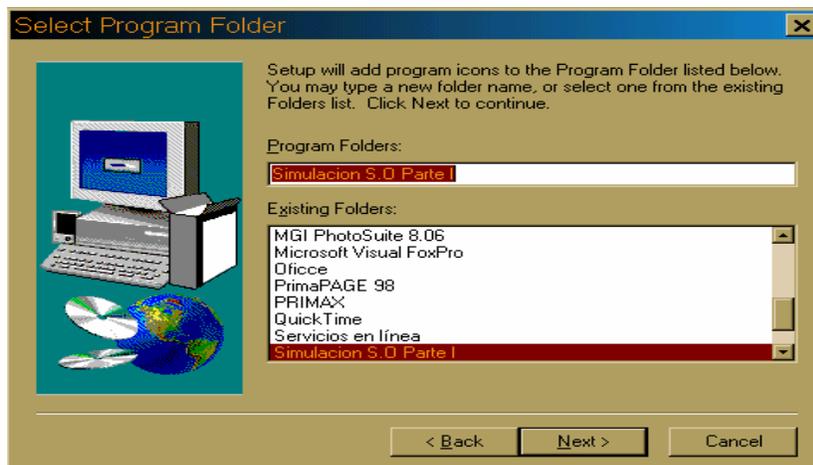


☞ Haga clic en Next de la ventana de bienvenida (Welcome)



☞ Complete la información de la ventana de información del usuario (User Information) con su nombre y el de su compañía o universidad. Haga clic en el botón Next.

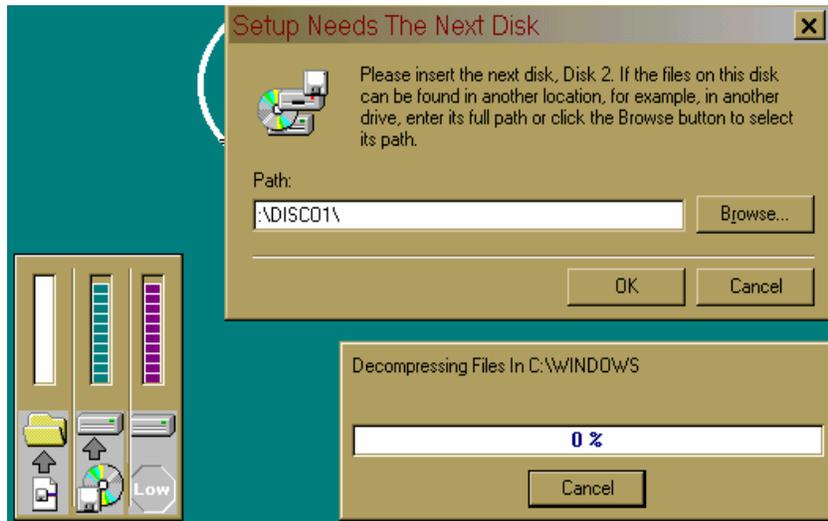
☞ En la siguiente ventana (Choose Destination Location), puede cambiar la ruta en la cual quedará instalado *simulación S.O parte I* en su PC, por defecto es C:\Archivos de programa\Simulacion S.O Parte I.



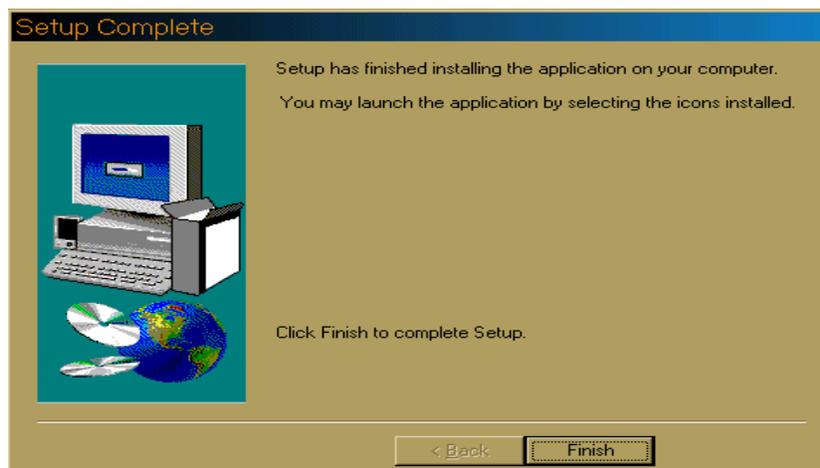
- En la ventana de selección del folder del programa (Select Program Folder), deje la opción que aparece por defecto y oprima el botón Next.



- En la ventana Start Copying Files, verifique la información, si ésta es correcta oprima Next, sino oprima Back para regresar a las ventanas anteriores y corregir.



- Para continuar con la instalación de *Simulación S.O Parte I*, el Setup necesita el disco 2, en la ruta que aparece por defecto en Path, cambie disco1 por disco2 y oprima el botón Ok de la ventana 'Setup Needs the Next Disk'. (Observación: Esta opción sólo es válida cuando se instala desde disquete)



- ☞ Para finalizar con la instalación, oprima Finish en la ventana Setup Complete.

### 1.3 EJECUCIÓN DE *SIMULACIÓN S.O PARTE I*



Al terminar la instalación, se puede ejecutar el programa de así:

Por el botón de *Inicio*, escoger la opción *Programas* y aparecerá ***SIMULACIÓN S.O PARTE I*** donde se encuentra el ejecutable.

### 1.4 CÓMO REMOVER *SIMULACIÓN S.O PARTE I* DE SU PC.

- ☞ Haga clic en el botón *Inicio* de Windows.
- ☞ Elija *Configuración* y después haga clic en *Panel de Control*.
- ☞ En la lista, escoja la opción *SIMULACIÓN S.O PARTE I*
- ☞ Haga doble clic en *Agregar o Quitar*.





## 1.6 SISTEMA DE VENTANAS

*Simulación S.O Parte I*, utiliza el sistema de ventanas como mesa para mostrar las diferentes simulaciones; de esta manera tendremos cada simulación desarrollándose en una ventana diferente, pero el manejo de dichas ventanas es igual independientemente de cual sea, es decir, se controlan de la misma forma.

**1.6.1 Controles de una ventana.** Debido a la naturaleza del software, es poca la interactividad que éste tiene con el usuario, ya que su objetivo primordial es mostrar cómo se llevan a cabo ciertas actividades que gestiona un Sistema Operativo, por ello, el usuario solo puede ejecutar los siguientes controles sobre una ventana:

☞ **Cerrar la ventana:** Con el Mouse: haga clic en el botón , el cual se encuentra en la parte superior derecha de la ventana o en el botón Cerrar que se encuentra en la parte inferior derecha de la ventana. Con el teclado: oprima las teclas ALT + E.

☞ **Desplazar la ventana en diferentes posiciones de la pantalla del monitor:** Coloque el Mouse en la Barra de Títulos de dicha ventana, haga clic en el botón izquierdo del Mouse manteniéndolo presionado, arrastre la ventana hasta el lugar deseado, finalmente, libere la presión del Mouse y donde lo suelte quedará ubicada la ventana.

### 1.6.2 Manejo de marcos de página



La figura 2 muestra un modelo de un Marco de página, el cual está conformado por tres páginas y se encuentra activa la tercera, titulada Mapa de Memoria. Para ver el contenido de determinada página, solo debe

hacer clic en el título de página deseado, ésta se activará inmediatamente permitiendo ver su contenido. Al activar una nueva página, se desactiva la que en ese instante se esté usando. Sabemos que una página se encuentra activa porque el recuadro en el que está su título se encuentra resaltado de los demás.

**1.6.3 Manejo de teclas abreviadas.** Usted puede ejecutar comandos a través del teclado haciendo uso de teclas abreviadas, por ejemplo: en la barra de botones aparece el botón simular así: Simular, esto significa que además de poder ejecutar las acciones de dicho botón haciendo click sobre él, también puede hacerlo con las teclas ALT + la tecla que se encuentre subrayada, en este caso sería ALT + S.

#### 1.6.4 Manejo de la Barra de Botones

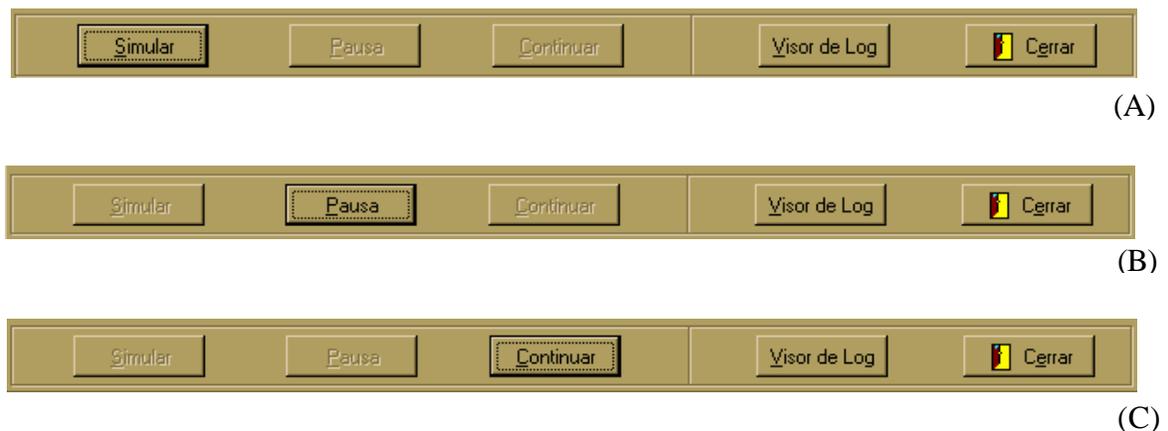


Figura 3. Estados de la Barra de Botones

La Barra de Botones está ubicada en la parte inferior de todas las ventanas que la usan en el software. Esta barra consta de dos bloques, el primer bloque se encuentra a la izquierda, éste puede estar en tres estados: Simular, Pausa y Continuar, mostrados en los estados A, B y C de la figura 3 respectivamente; el segundo bloque se encuentra a la derecha, posee dos botones: Visor de Log, y Cerrar, los cuales permanecen siempre activos.

Al iniciar cualquiera de las ventanas que posea la Barra de Botones, aparece activo únicamente el botón Simular en el primer bloque, mientras que los otros dos permanece inactivos, como se muestra en la figura 3 (A). Al ejecutar el botón Simular éste se inactiva y automáticamente se activa el botón Pausa, como se muestra en la figura 3 (B). Al ejecutar el botón Pausa éste se inactiva y automáticamente se activa el botón Continuar, como se muestra en la figura 3 (C), los botones del segundo bloque siempre permanecen activos. Miremos ahora las características de cada botón de la barra de botones:

**Simular:** A través de este botón se da inicio a cualquier de las simulaciones del software, por tanto se encuentra activo al iniciar cualquiera de las ventanas contenedoras de simulaciones, permanece inactivo durante el desarrollo de la simulación al final de la cual se activa nuevamente indicando que puede dar inicio a una nueva simulación. Para

ejecutar las acciones de este botón a través del Mouse haga clic sobre él. A través del teclado con las teclas ALT + S.

**Pausa:** Este botón detiene la simulación que se esté ejecutando por el instante de tiempo y en momento que lo decida el usuario, por tanto se activa automáticamente al iniciar una simulación. Si el usuario decide ejecutarlo, éste botón, permanece inactivo hasta que decida continuar con la simulación. Para ejecutar las acciones de este botón a través del Mouse haga clic sobre él. A través del teclado con las teclas ALT + P.

**Continuar:** Luego que el usuario detiene la simulación y desea continuar, la única forma es a través de este botón, el cual permanece automáticamente activo cuando se realiza una pausa hasta que el usuario decida continuar, momento en el cual se inactiva indicando que la simulación sigue su curso. Para ejecutar las acciones de este botón a través del Mouse haga clic sobre él. A través del teclado con las teclas ALT + C.

### **Visor de Log:**

A través de este botón se despliega una ventana (figura 4) que contiene un resumen de lo que está ocurriendo hasta ese instante con la simulación en curso. El contenido del Visor se borra únicamente



**Cerrar:** Este botón al igual que el botón del Visor, puede ser pulsado en cualquier momento que el usuario lo desee, con lo cual se interrumpe automáticamente la simulación que se esté llevando a cabo cerrando esa ventana y pasa el control a la ventana que le hizo la referencia.



## 2. MENÚ PRINCIPAL



Figura 5. Menú Principal

*Simulación S.O Parte I* está dotado de un menú principal desplegable, el cual está ubicado en la parte superior de la pantalla de presentación. Como se observa en la figura 5, dicho menú contiene 5 ítems, a través de los cuales, usted puede llegar a las diferentes simulaciones que enmarca éste software. Veamos entonces el contenido de cada ítem.

### 2.1 Aplicación

Para tener acceso este ítem basta con hacer clic sobre él con el mouse. A través del teclado solo tiene que presionar las teclas ALT + A.

Al desplegar el ítem Aplicación, aparecerá el siguiente submenú:





Figura 6. Submenú Aplicación

Como se observa en la figura 6, el submenú Aplicación consta de los ítems Acerca de..., en donde usted encuentra información acerca de *Simulación S.O Parte I* (figura 7), el segundo y último ítem se titula Salir, como su nombre lo indica, usted puede abandonar la aplicación con solo hacer clic sobre ella o con el teclado con la ayuda de las teclas ALT + S. Para llegar a la ventana Acerca de... de la aplicación, basta con hacer clic sobre él o presionando las teclas ALT + A.



Figura 7. Acerca de Simulación S.O Parte I



## 2.2 Memoria Memoria

Al hacer clic en éste ítem o al presionar las teclas ALT + M, se despliega la siguiente ventana



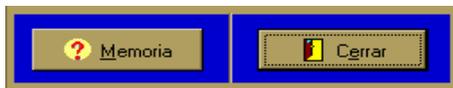
Figura 8. Administración de memoria

La figura 8 corresponde a la ventana principal de las simulaciones de los mecanismos de administración de memoria. Para acceder a cada

mecanismo solo tiene que hacer click en la opción deseada en el recuadro llamado Mecanismos de asignación de memoria.

La ventana también contiene una guía de colores en los cuales se explica el significado de cada color utilizado por los mecanismos de asignación en los mapas de memoria.

En la parte inferior de la ventana se encuentran los botones Memoria y Cerrar



A través del botón Memoria, tiene acceso a una ventana de apoyo (figura 13) que contiene información acerca de los mecanismos de asignación de memoria. Por medio del botón Cerrar se cierra esta ventana y se regresa al menú principal.

Para la mayoría de los mecanismos de asignación de memoria que se describen a continuación, existen ciertas características las cuales tienen en común. Veamos cuales son.

El estado de la memoria se observa en el Mapa de memoria el cual está representado en la figura 9. Cada espacio de color blanco significa memoria desocupada o disponible para alojar procesos, las partes de color azul representan la memoria que está siendo ocupada por los procesos solicitantes y las partes de color negro representan espacios de memoria

desperdiciados. En el caso de la figura 9, la página 3 del proceso 3 tiene un tamaño de 2k pero el tamaño de cada bloque en memoria es de 3k, entonces hay 1k de memoria desperdiciado.

La parte superior del mapa de memoria, marcado con las iniciales de Sistema Operativo (S.O), representan todas las direcciones de memoria reservada, las cuales no están disponibles para alojar procesos de usuario, a ello se le asignó los primeros 5k del total del tamaño de la memoria que es 32k, es decir, que existe un total de 27k disponibles para los procesos de usuario.

Para los mecanismos de asignación Contigua Simple, Particiones Estáticas y Particiones Dinámicas, la letra P que aparece en el interior de las particiones asignadas, Significa Proceso y el número que aparece al lado derecho de dicha P, significa el número del proceso, de esta manera P3 por ejemplo, significa proceso tres.

Para la paginación se representa primero la página y luego el proceso así: P1-P1 y se lee página uno del proceso uno. Para la segmentación se representa y se lee de la siguiente forma: S1-P2 segmento uno del proceso dos. Por último, para la segmentación y paginamiento con demanda, primero está la página, luego el segmento al cual pertenece esa página y

termina con el proceso al cual pertenece dicho segmento así: P1-S1-P1 y se lee página uno del segmento uno del proceso uno y así sucesivamente.

El número que aparece al lado del mapa de memoria representa la direcciones en memoria ya sea de una partición, un bloque, una página o un segmento.

La figura 10 es un modelo de una estructura de datos utilizada por algunos mecanismos de asignación, en este caso representa una tabla de particiones, la cual contiene el número de la partición (# Partición), la dirección base (Dir\_Base), el tamaño(Tam\_partición) y el estado de dicha partición (Estado), es decir si está asignada o no. Estas tablas son usada como estructura de datos para llevar el estado de las particiones por los mecanismos de asignación particiones estáticas, particiones dinámicas y por la técnica de compactación. Cabe recordar que la compactación no es un mecanismo de asignación de memoria pero que dada su importancia, *Simulación S.O Parte I* tiene destinada una simulación completa para mostrar las características de ésta técnica. En la parte inferior de la figura 10 se encuentra un tablero que informa el estado en que se encuentra la simulación desde que se hace click en el botón Simular.

En caso de que toda las particiones que se encuentren en un momento dado no alcance a ser visualizadas en la tabla de particiones, usted puede

desplazarse a lo largo de la tabla a través de la barra de scroll ubicada en la parte derecha de dicha tabla.

*Simulación S.O Parte I* le ofrece la oportunidad de poder simular un mecanismo de asignación de memoria mediante dos políticas en una sola venta, en cuyo caso se presenta la ventana con mínimos dos marcos de página, uno para los mapas de memoria y el o los demás para las tablas (ver por ejemplo figura 21), recalcando nuevamente que las tablas son usadas para representar las estructuras de datos utilizadas por los mecanismos.

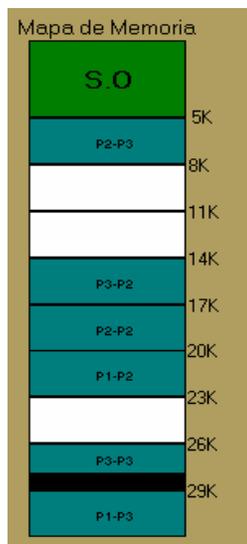


Figura 9. Mapa de memoria

Tabla de Particiones

# Partición	Dir_Base	Tam-particion	Estado
1	5K	8	Asignada
2	13K	6	Asignada
3	-	-	-

Se asignó un proceso de tamaño 6 en la particion 2

Figura 10. Estructura de datos

Todas las simulaciones cuentan con una ventana que le informa al usuario cuando termina la simulación, siempre son usadas diez peticiones, ya sea

en memoria, disco, entrada/salida o archivos, al ser atendidas dichas peticiones, aparece un mensaje de información (figura 11) al cual se debe hacer click en Ok para continuar, acción con la cual se activa nuevamente el botón Simular, lo cual indica que puede dar inicio a otra simulación.

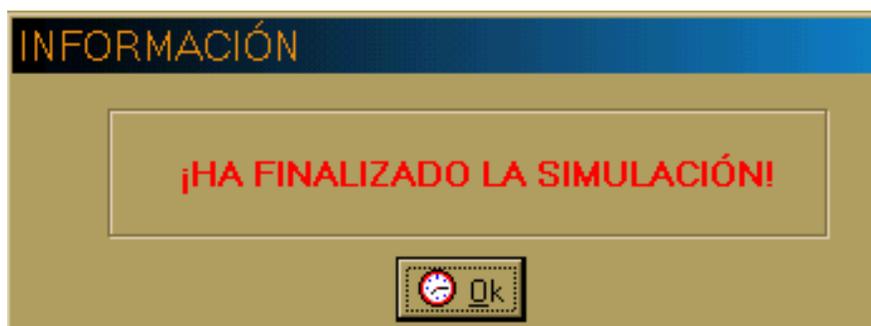


Figura 11. Finalización

Todas las simulaciones de asignación de memoria que utilizan dos políticas, como son particiones estáticas y dinámicas, paginamiento con demanda y la técnica compactación, requieren de una ventana auxiliar, por ejemplo la mostrada en la figura 14, en donde se debe elegir mínimo una política para su simulación, pues, en caso de que oprima el botón Aceptar sin señalar ninguna política, aparece un mensaje de error (figura 12), el cual desaparecerá en cuanto usted presione el botón Ok o utilice las teclas abreviadas.



Figura 12. Error de Selección

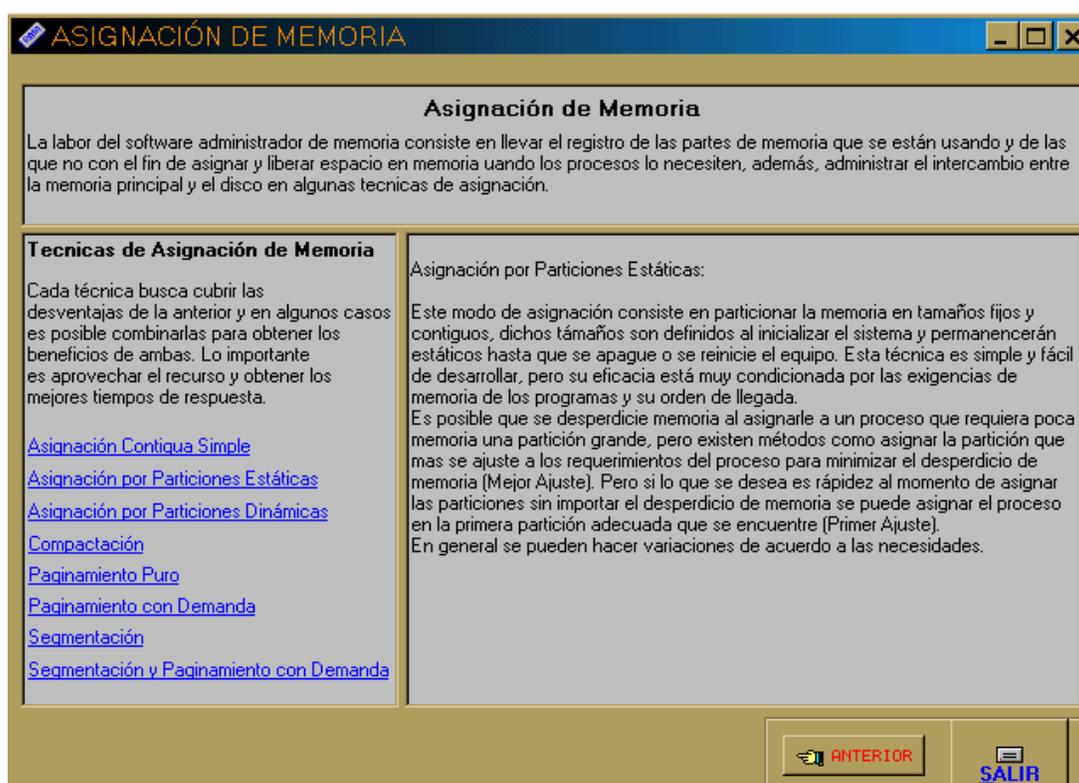


Figura 13. Información de Asignación de Memoria

La figura 13 muestra información acerca de los mecanismos de asignación de memoria, usted como usuario puede tener acceso a dicha información haciendo click en los diferentes subtítulos que aparecen en la parte izquierda de la ventana en color azul. La información correspondiente a

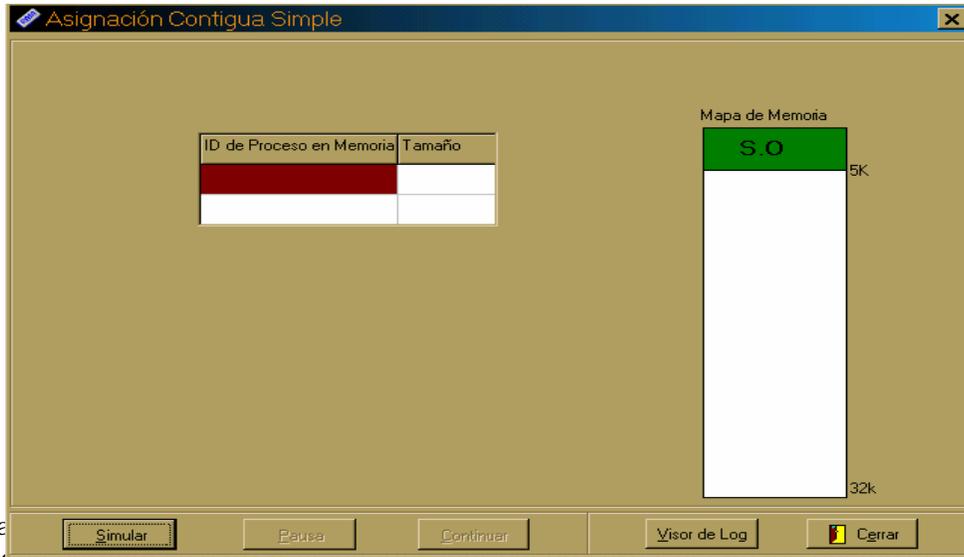
cada subtítulo aparece en el recuadro que se encuentra a la derecha de la ventana.

Para llegar hasta la ventana que muestra la figura 13, basta con utilizar las teclas abreviadas ALT + M o hacer click en el botón Memoria de la ventana que se encuentra en la figura 8.

Con Los botones ANTERIOR y Salir, usted puede regresar a la ventana principal de memoria (figura 8).

Ahora si pasemos a la descripción de cada una de las simulaciones de asignación de memoria.

### 2.2.1 Contigua Simple Contigua Simple



La ventana muestra una tabla que permite visualizar el identificador del proceso y el tamaño del mismo.

Figura 14. Contigua Simple

### 2.2.2 Particiones Estáticas

Particiones Estáticas

Al seleccionar particiones estáticas aparece la siguiente ventana



A través de este dialogo se decide que política de asignación empleará la simulación: Primer Ajuste, Mejor ajuste o si lo prefiere la simulación utilizara las dos políticas y mostrará la secuencia de asignación para ambas.

Figura 15. Particiones Estáticas

#### 2.2.2.1 Primer Ajuste Primer Ajuste



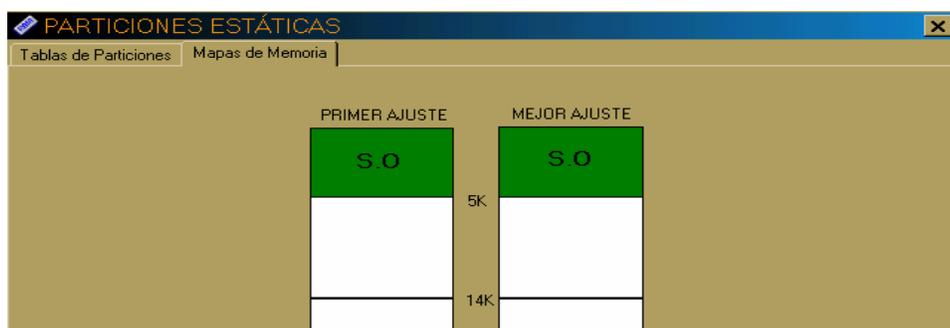
A lo largo de toda esta simulación se muestra el estado de la memoria a través de un mapa de memoria, a la vez que se ilustra la tabla de particiones; que es una de las estructuras de datos empleada por este mecanismo.

**2.2.2.2 Mejor Ajuste**  Mejor Ajuste



Figura 17. Particiones estáticas. Mejor ajuste

**2.2.2.3 Primer ajuste y mejor ajuste**





Al seleccionar ambas opciones de la Figura 10 se despliega la ventana anterior, cuyo marco de página ofrece dos vistas, una en la que se puede apreciar las tablas de particiones(Figura 12) y en la otra los Mapas de memoria de la simulación (Figura 13) de ambas políticas de asignación.

### 2.2.3 Particiones Dinámicas

Particiones Dinámicas

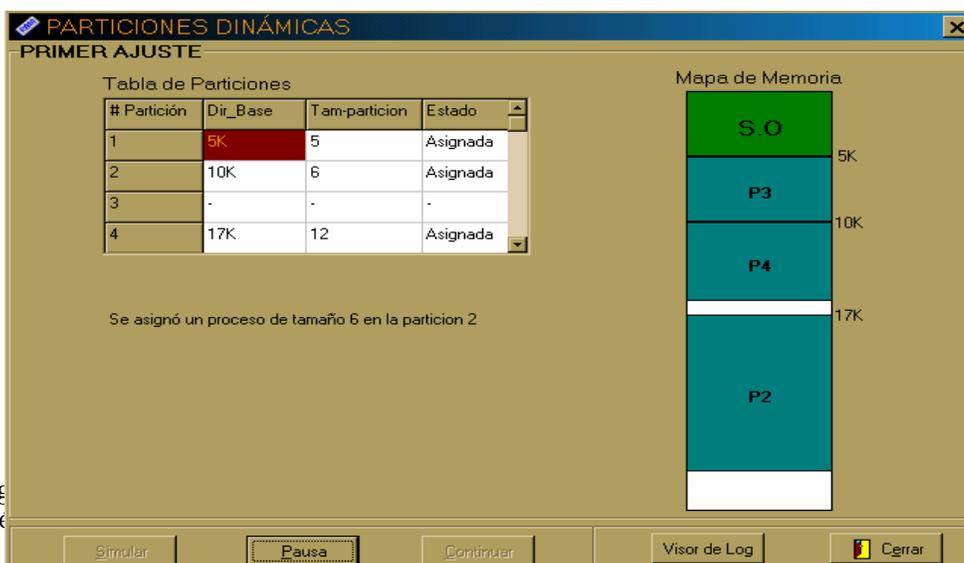


Usted Puede simular la asignación de procesos a las particiones dinámicas de manera que cuando quedan espacios de memoria disponibles, estos sean asignados mediante la política primer ajuste y/o mejor ajuste.

Figura 20. Particiones Dinámicas

#### 2.2.3.1 Primer Ajuste

Primer Ajuste



Análogamente, se puede observar que se encuentran alojados tres procesos, P1,P2 y P3, ocupando las particiones 1, 2 y 3 respectivamente. En la tabla se encuentra el estado de dichas particiones.

Figura 21. Particiones Dinámicas. Primer Ajuste

La simulación también posee un tablero que le informa lo que está ocurriendo en un momento dado, en este caso se está informando que acaba de ser alojado un proceso de tamaño 10K en la partición 2.

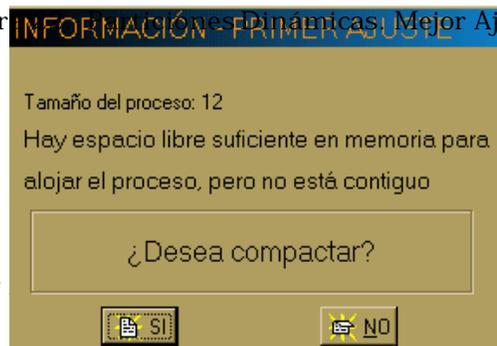
**2.2.3.2 Mejor Ajuste**  Mejor Ajuste



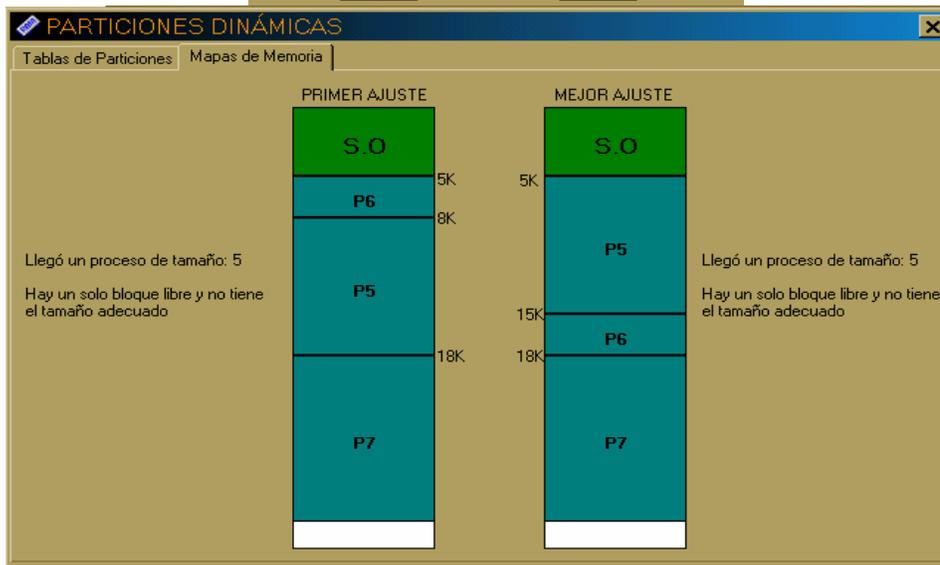
Puede p  
suficier  
simulac  
dicho c  
desea c

espacio  
a, en  
gir si

Figura 2.2.3.2. Particiones Dinámicas, Mejor Ajuste



**2.2.3.3 Primer Ajuste**





Al elegir ambas políticas de asignación, *Simulación S.O Parte I* le ofrece la oportunidad de apreciar ambos mecanismos de asignación en una sola ventana (figuras 21 y 22) con el fin de que usted pueda sacar sus propias conclusiones respecto a las características de cada una, así como ventajas y desventajas.

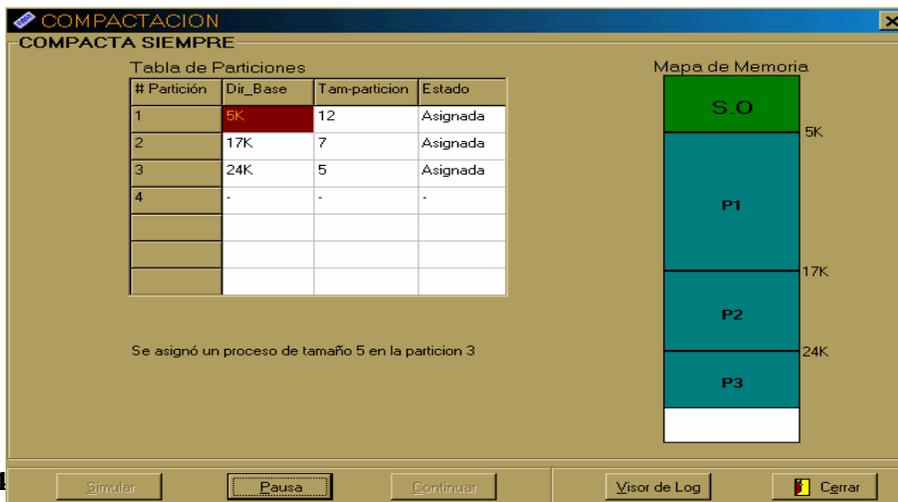
### 2.2.4 Compactación

Compactación



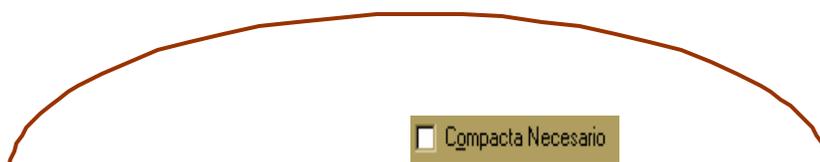
Dependiendo de la elección que usted haga en la ventana de la figura 26, puede llegar a una de las siguientes simulaciones. Es de la 26 Compactación:

#### 2.2.4.1 Compacta Siempre



2.2.4

Figura 27. Compacta Siempre



El mensaje señalado en la figura 28, le informa al usuario que para que el proceso solicitante pueda ser atendido es necesario compactar la memoria y por tanto se va a llevar a cabo dicha actividad.

### 2.2.4.3 Compacta Siempre y Compacta Necesario

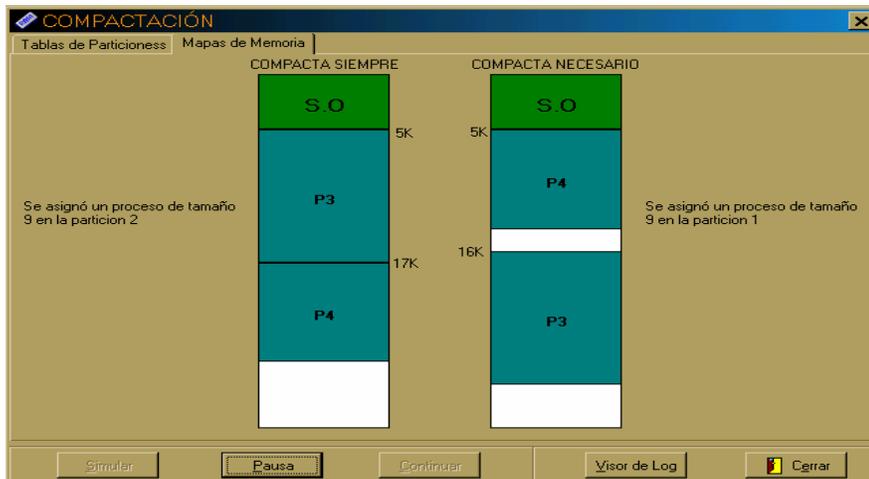


Figura 29. Compacta Siempre y Compacta Necesario. Mapas de Memoria

2.2.5

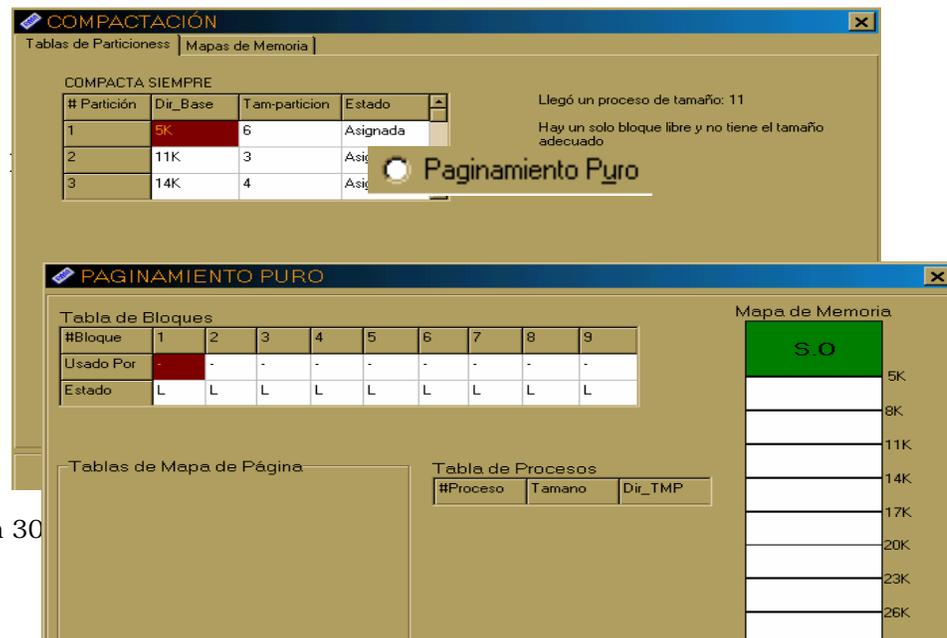


Figura 30

La figura 31 A, muestra la ventana del mecanismo Paginamiento Puro sin haber dado inicio a la simulación, allí se destaca la Tabla de Bloques, en la cual se describe el número del Bloque, por quien está siendo usado, es decir que página de que proceso se encuentra alojada en él y por ultimo se muestra el estado de dicho bloque (si está libre u ocupado L/O). Mientras que en la figura 31 B, la ejecución se está llevando a cabo. Se puede apreciar la dinamicidad de las tablas de Mapa de Página en donde se describe el número de página y de bloque en donde se encuentra. La numeración que aparece en la esquina superior de las tablas de mapa de página, corresponde a la dirección de memoria en donde se encuentra dichas tablas.

La tabla de procesos por su parte, describe el número del proceso que se encuentra en memoria, su tamaño y la dirección de la tabla de página de dicho proceso (Dir\_TMP).

## 2.2.6 Paginamiento con Demanda

Paginamiento con Demanda



### 2.2.6.1 FIFO FIFO 32. Paginamiento con Demanda



En el mecanismo de asignación Paginamiento con demanda, los procesos a medida que llegan, sus páginas son alojadas en disco y estando en el marco de página del Mapa de Memoria, la forma de avisarle a usted como usuario de dicho evento es a través de la pequeña ventana resaltada en la figura 33.

Al igual que en el mecanismo de asignación anteriormente descrito, en Paginamiento con demanda también se actualizan las tablas a medida que ocurren ciertos eventos, como son la llegada de un proceso, remoción de páginas en memoria para alojar alguna desde disco, la finalización de un proceso, etc. Para informar al usuario que ha finalizado un proceso, también se recurre a la ventana resaltada en la figura 33.



Figura 34. FIFO. Tablas de Procesos y de Bloques



El campo elección de la tabla de bloques (figura 34) indica el orden de llegada de las páginas a memoria, de esta forma, al momento de remover una página, se escoge la que tenga el campo elección más bajo. En el caso mostrado en la figura 34, se removería la página 3 del proceso1, que está ubicada en el bloque 9.

La tabla de Procesos (figura 34) posee el campo Dir\_TMA, el cual se refiere a la dirección de la tabla de mapa de archivos en donde se encuentran las páginas de determinado proceso. Las TMP (tabla de mapa de página), al igual que las TMA (tabla de mapa de archivo), se pueden observar en la figura 35. En el campo Ubicación de las TMP, la letra M significa que la página está en memoria y la letra D que la página está en disco. El campo Dirección de la TMA se refiere a la dirección en disco de determinada página.

Para las TMP, el campo Número de Bloque (#Bloque) de determinada página, aparece con el símbolo ' - ' cuando una página se encuentra en disco. De forma análoga, para la TMA ocurre cuando dicha página se encuentra en memoria.

### 2.2.6.2 LRU

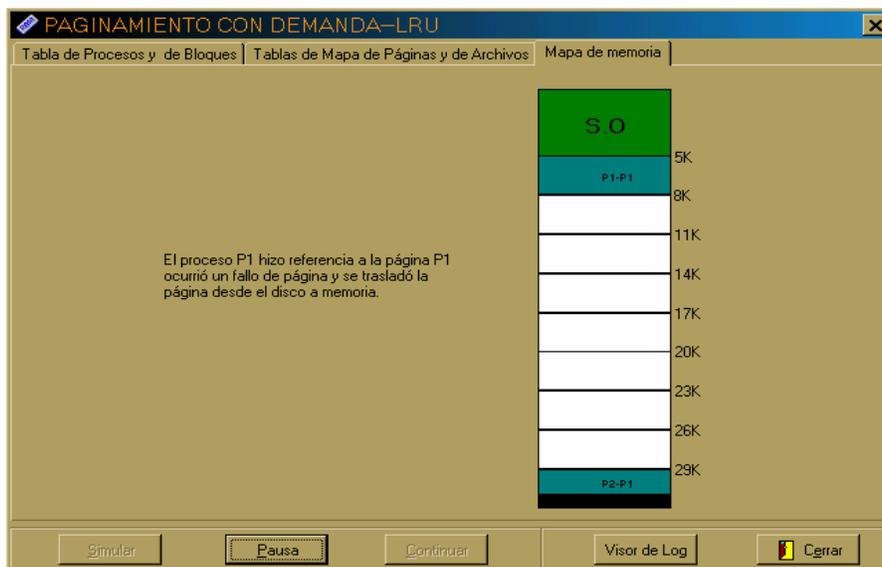
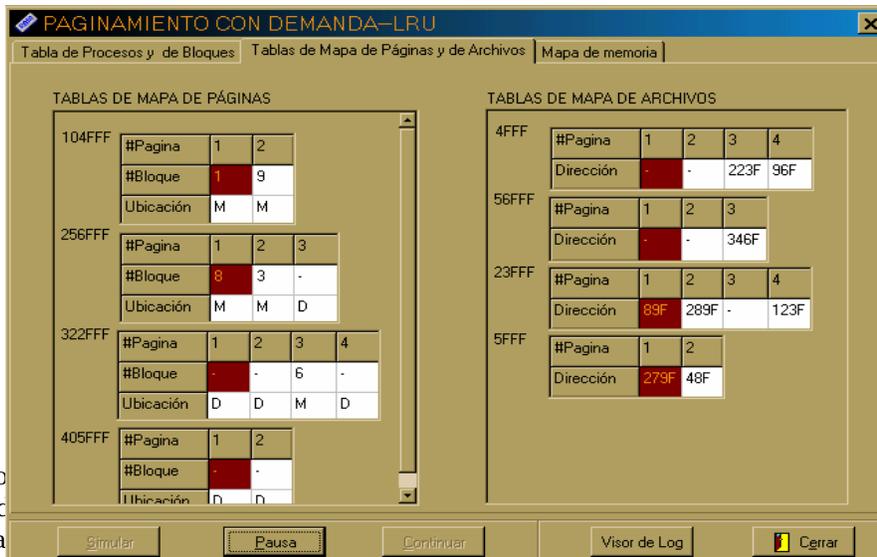


Figura 36. Paginamiento con Demanda. LRU

#Proceso	Tamaño	Dir_TMP	Dir_TMA
1	5	104FFF	4FFF



Para la po... Elección c... referencia... valor más bajo. En el caso particular que ilustra la figura 37 en la Tabla de Procesos, en donde dos páginas han sido referenciadas por el mismo número de veces, se baja a disco la que fue cargada primero en memoria.

### 2.2.6.3 FIFO y LRU

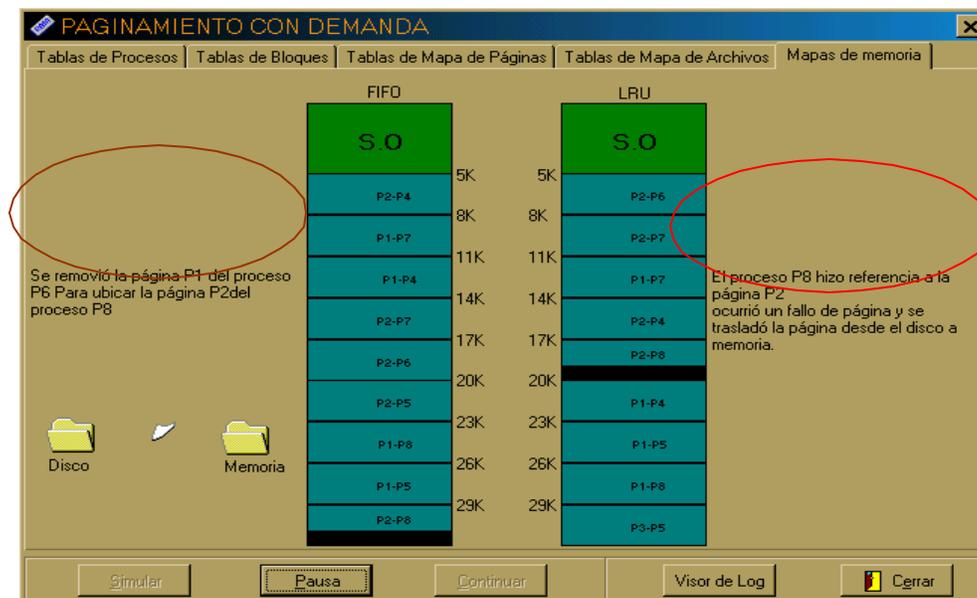


Figura 39. FIFO v LRU. Mapas de Memoria

En los mecanismos de asignación en donde interviene la memoria virtual, como son Paginamiento con Demanda, Segmentación y la combinación de ambos Segmentación y Paginamiento con demanda, los instantes en que ocurren los fallos de página pero existe espacio en memoria, se le informa al usuario el evento través de mensajes, como el resaltado en color rojo en la figura 39. En Paginamiento con Demanda, en caso de que no exista espacio en memoria para alojar la página solicitada, además de informarle al usuario a través del mensaje resaltado en la parte izquierda de la figura 39, se muestra la animación (carpetas en amarillo) del momento en que se está trasladando la página solicitada desde el disco a memoria.

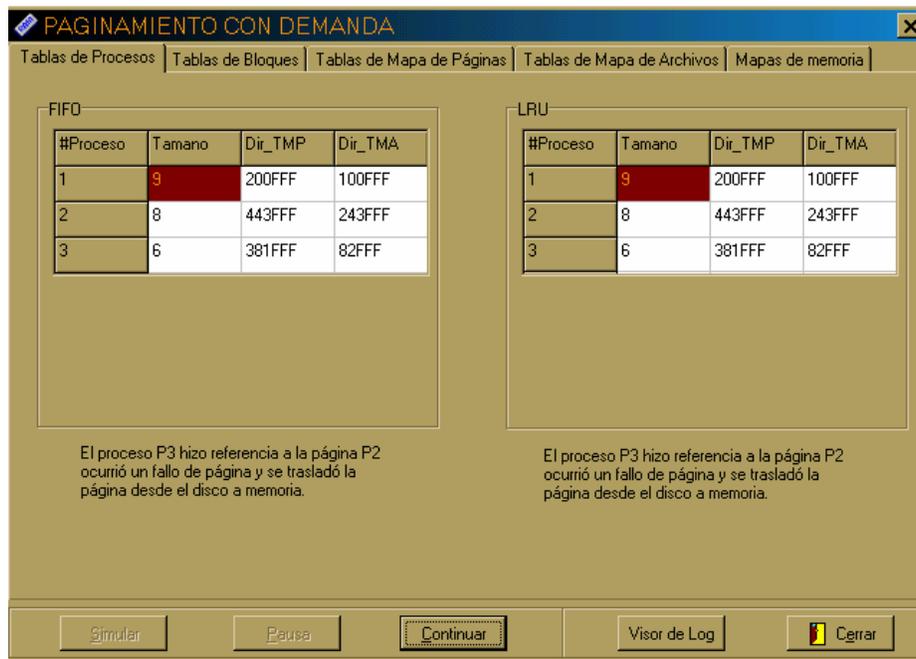


Figura 40. FIFO y LRU. Tablas de Procesos



### 2.2.7 Segmentación Segmentación

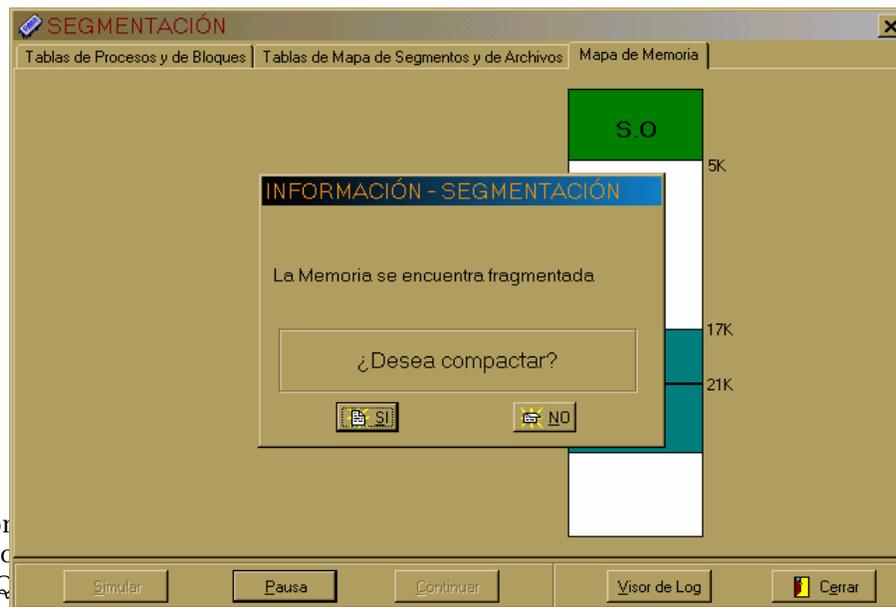


Figura 44. Segmentación. Mapa de Memoria

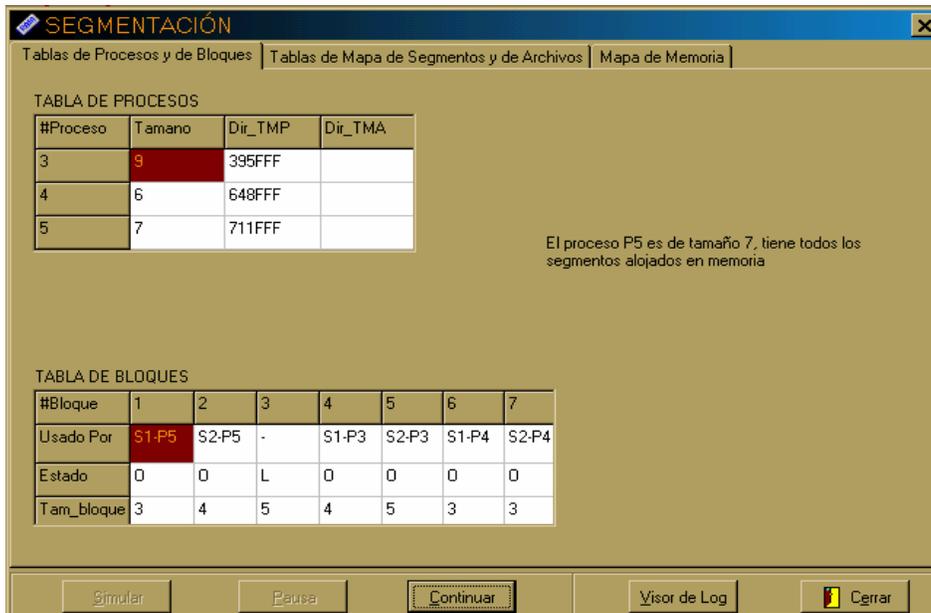
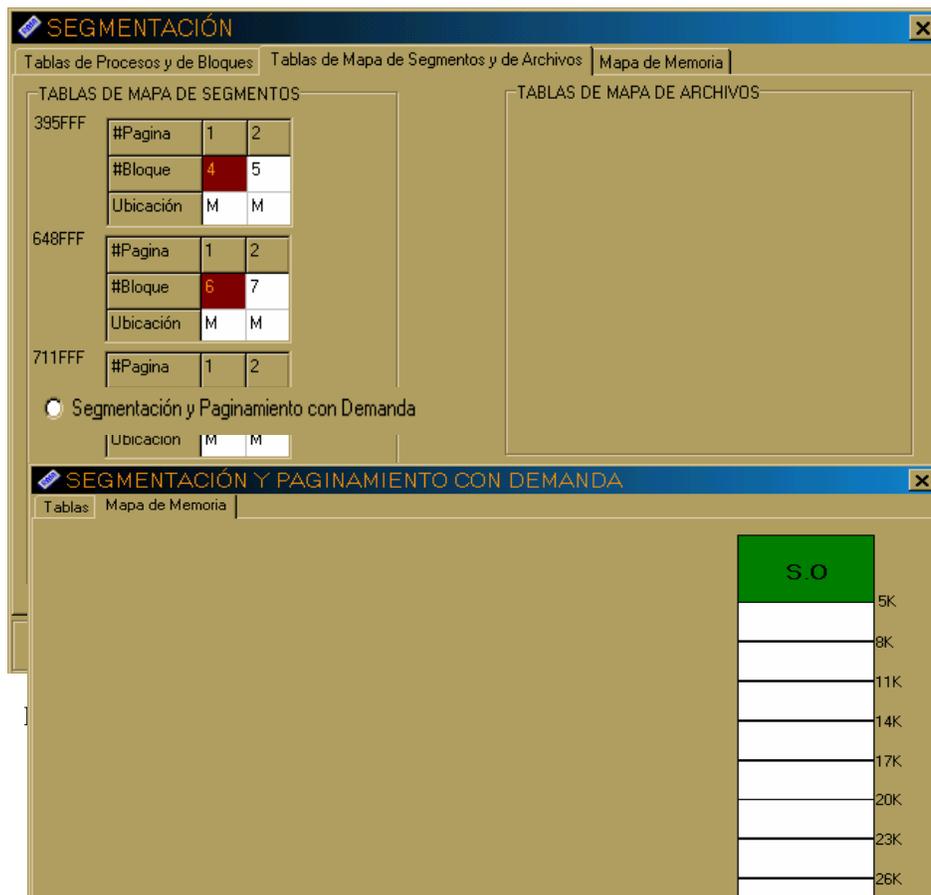


Figura 45. Segmentación. Tablas de Procesos y de Bloques

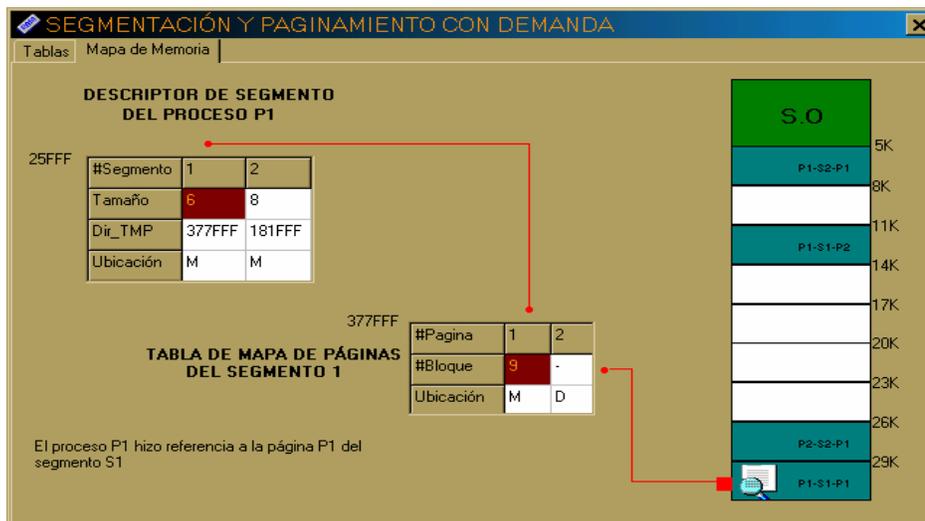
2.2.8



Ya conocida la dinámica de las tablas a través de los mecanismos de asignación anteriormente descritos, el contexto de SPCD (Segmentación y Paginamiento con Demanda) varía un poco respecto a las demás simulaciones concernientes a Memoria, pues aquí solo es ilustrada la tabla de Mapa de Página y el Descriptor de Segmento concernientes al proceso en cuestión en un momento dado, es decir, el proceso que haga una referencia a una de sus páginas.

En la figura 47 se ilustra la forma de entrada de la simulación del mecanismo SPCD; Cuando se presiona el botón Simular, empiezan a llegar las peticiones, las cuales son alojadas en Disco, éste a la vez está representado por la forma circular que aparece en el centro de la figura 48 (porcentaje ocupado en disco), en él, el color blanco representa espacio disponible, el color gris es la parte ocupada por los procesos solicitantes que van llegando y el color azul corresponde a otras aplicaciones allí alojadas. En las etiquetas amarillas aparece el porcentaje que cada una de las partes descritas ocupa en disco.

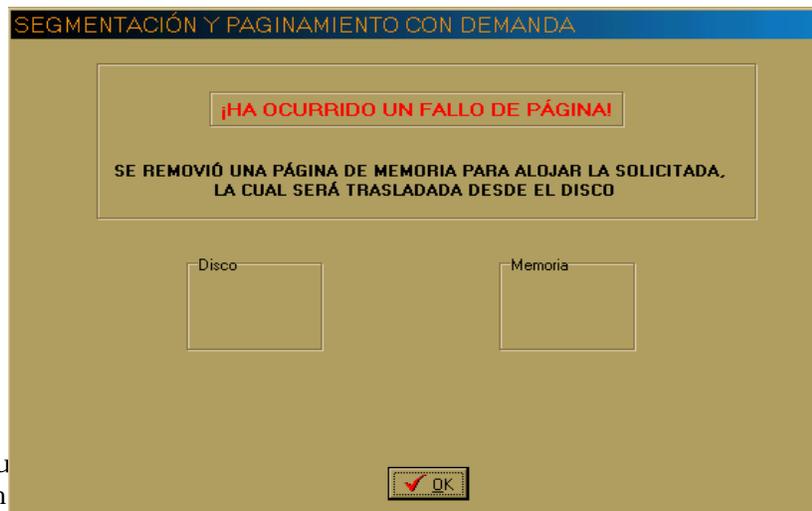
Cada vez que llega un proceso, se divide en segmentos y éstos en páginas, las cuales son alojadas en disco, por ello siempre se muestra el porcentaje ocupado en disco por dicho proceso.



La figura 49 ilustra el instante en que una página está siendo referenciada, en este caso es la página uno del segmento uno del proceso uno (P1-S1-P1), en esta simulación aparece una nueva estructura de datos por para cada proceso, representada por la tabla denominada Descriptor de Segmento, éste tiene cuatro campos en los que describe el número del segmento (#Segmento), tamaño del segmento (Tamano), la dirección de la tabla de mapa de página en donde se encuentra las páginas en la que está dividido el segmento (Dir\_TMP) y la ubicación de dicho segmento (Ubicación), es decir, si el segmento está en disco o en memoria. Recuerde que un segmento se considera ubicado en memoria con que sólo una de sus páginas lo estén.

Cuando un proceso referencia una de sus páginas, aparece una animación como la que aparece en la figura 49, en donde se traza un camino con líneas de color rojo, que parte desde el Descriptor de Segmento, apuntando el segmento en cuestión, luego ubica la página referenciada en la tabla de Mapa de Página que corresponda con la dirección contenida en el Descriptor y por último llega al Mapa de Memoria, apuntando al bloque que contiene dicha página, momento en el cual aparece una animación de describe página encontrada.

El camino trazado en líneas de color rojo es recorrido por una figura cuadrada para ilustrar el orden en que suceden los eventos cuando es referenciada una página.



Si al hacer un clic en el botón de 'OK' se muestra una animación que ilustra el traslado de la página referenciada desde el disco hasta la memoria.

Figura 50. SPCD. Información de Fallo de Página

SEGMENTACIÓN Y PAGINAMIENTO CON DEMANDA									
Tablas   Mapa de Memoria									
TABLA DE BLOQUES									
#Bloque	1	2	3	4	5	6	7	8	9
Usado Por	P2-S1-P3	P2-S1-P4	P2-S2-P2	P1-S2-P5	P1-S2-P4	P1-S2-P3	P1-S1-P3	P2-S1-P5	P1-S1-P2
Estado	0	0	0	0	0	0	0	0	0
Eleccion	8	6	11	13	9	12	5	10	7
TABLA DE PROCESOS									

La figura 51 ilustra el marco de página de SPCD denominado Tablas, el cual contiene la Tabla de Bloques, en donde el campo Eleccion ilustra el número de veces que ha sido referenciada una página, Luego, cuando sea necesario, se remueve la página que esté ubicada en el Bloque que tenga éste campo con el valor más bajo para alojar la solicitada.

### 2.3 Disco Disco



Figura 52 A. Planificación de los Accesos a Disco



La figura  
Planifica

o el

botón Simular, Recordemos que para llegar a esta ventana solo debe escoger la opción Disco desde el Menú Principal.

La ventana de esta simulación consta de 5 Marcos de Página, de los cuales, el titulado Disco es el que aparece por defecto al iniciar y al finalizar la simulación. En los 4 Marcos de página restantes se encuentran los desplazamientos de la Cabeza Lectora (Brazo del disco) según los criterios de cada algoritmo de planificación, dichos desplazamientos se muestran en un plano que está descrito más adelante.

La figura 52 B también muestra la forma principal del Item Disco pero con la simulación en ejecución. En la parte superior de la ventana se encuentra un tablero que contiene en orden de llegada (de izquierda a derecha) las solicitudes que serán atendidas por el Disco, dichas solicitudes están representadas por el número de pista donde se encuentra el bloque que se va a leer o escribir.

En la ventana también se muestra la diferencia de cilindros en que incurre cada algoritmo de Planificación, esta diferencia es actualizada cada vez que se atiende una petición.

A través del botón Almacenamiento  se accede a una serie de ventanas que contienen información acerca



Figura 53. Disco. Primero en Llegar, Primero en Acceder

Las peticiones que van siendo atendidas, aparecen en el tablero que se encuentra en la parte superior del Marco de Página. Al finalizar la simulación el usuario puede comparar el orden en que fueron atendidas las peticiones por cada algoritmo. Recordemos que para desplazarse en los diferentes Marcos de Página que se encuentran en una ventana, basta con hacer click en el respectivo titulo o a través del teclado, pulsando la tecla de tabulación ( `Tab` ), hasta poseionares en cualquiera de los títulos, luego se ubica en los diferentes Marcos de Página con las teclas de desplazamiento horizontal, ya sea a la izquierda o a la derecha ( `←` / `→` ).

El usuario también sabe cuál es la Pista de la cual se parte para simular los diferentes algoritmo y poder obtener las diferencias de Cilindro para cada uno. La pista de la cual se parte se conoce como Pista de Inicio y se puede apreciar en los diferentes Marcos de Página desde que se pulsa el botón Simular. En el caso particular de la figura 53, la Pista de Inicio es igual a 75.

Los desplazamientos de la Cabeza Lectora se aprecian en el plano en escala que aparece también en la figura 53, es decir, la escala del plano es dinámica, ya que cambia a conveniencia cada vez que se atiende una petición.

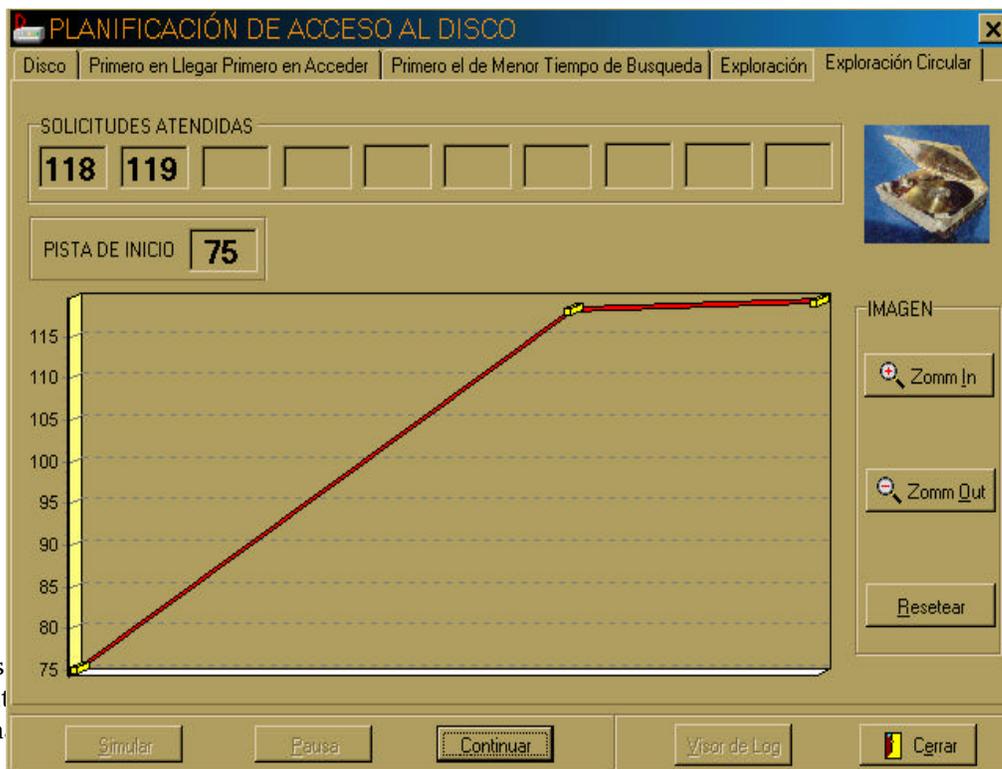
Las simulaciones de Planificación de Accesos a Disco cuentan también con una serie de controles que permiten manipular la imagen que representa los desplazamientos de la cabeza lectora. En el recuadro llamado Imagen, se encuentran tres botones, el primero es un Zoom In `+`, con el que puede acercar la imagen, el segundo Zoom Out `-` para alejar la imagen y el tercer botón Resetear `↺`, con el cual puede obtener la imagen original después de haber hecho un Zoom In o un Zoom Out.

A continuación se encuentran las figuras 54, 55 y 56 en las que se encuentra los Marcos de Página contenedores de los algoritmos de Planificación del Disco Primero el de Menor Tiempo de Búsqueda (SSTF), Exploración (Scan) y Exploración Circular (C-Scan), respectivamente.



El Visor de Log siempre permanece inactivo en las simulaciones de planificación, ya que todas las características de cada algoritmo son mostradas de manera totalmente clara y todas las acciones que se realizan (atención de peticiones), aparecen en los tableros de atención de peticiones (Solicitudes Atendidas).





Des  
vent  
alm

La figura 57 muestra la primera de dichas ventanas, en ella se encuentran los controles para acceder a las demás, por medio de los siguientes botones:



Con el botón Anterior se regresa a la anterior ventana, para ir a la ventana posterior es a través del botón Siguiente y para abandonar las ventanas de información acerca del Disco, el botón Salir. Si pulsa dos veces consecutivas el botón Siguiente, aparecen las ventanas mostradas en las figuras 58 y 59 respectivamente. Con el botón Anterior obtiene la secuencia inversa de las ventanas de información.

Al pasar  
significa  
Mouse c

tra su  
l

**Disco**

Los discos son los dispositivos de almacenamiento más comunes. Aceptan y recuperan datos a alta velocidad. Los datos son transferidos entre el disco y la memoria en bloques.

Los discos pueden ser fijos y removibles. Los primeros no se pueden cambiar, mientras que los segundos pueden ser intercambiados, con lo que se consigue un mayor volumen de almacenamiento. Pueden reunirse varios discos en un paquete (disk pack) para conseguir con ello un mayor almacenamiento en un reducido espacio. Todos los discos de un disk pack giran a la misma velocidad. Cada disco se divide en pistas concéntricas de tal forma que todas las homólogas de los distintos discos forman lo que se denomina cilindro. Cada pista, a su vez, se divide en sectores y cada sector contiene un bloque de información.

[¿DiskPack?](#)

**PARTES DEL DISCO**

- 1 Caja metálica
- 2 Plato metálico recubierto
- 3 Cabeza de lectura
- 4 Sistema de rotación de los platos
- 5 Chip de control

**ANTERIOR** **SALIR** **SIGUIENTE**

**Disco**

Brazos de acceso

Pista

Discos

Movimiento del peine

Cabezales de lectura/escritura

Giro

Cilindro

[¿DiskPack?](#)

**PARTES DEL DISCO**

- 1 Caja metálica
- 2 Plato metálico recubierto
- 3 Cabeza de lectura
- 4 Sistema de rotación de los platos
- 5 Chip de control

**ANTERIOR** **SALIR** **SIGUIENTE**

Figura 57 B. Disco. Información General

**Disco**

**Algoritmos de Planificación** **Tiempos de Acceso a Disco**

En la ventana que ilustra la figura 58, haciendo click en los subtítulos que aparecen subrayados de color azul, obtiene una breve referencia al respecto, Por ejemplo, si hace click en Tiempo de Búsqueda, obtiene la definición de ese término y así sucesivamente.

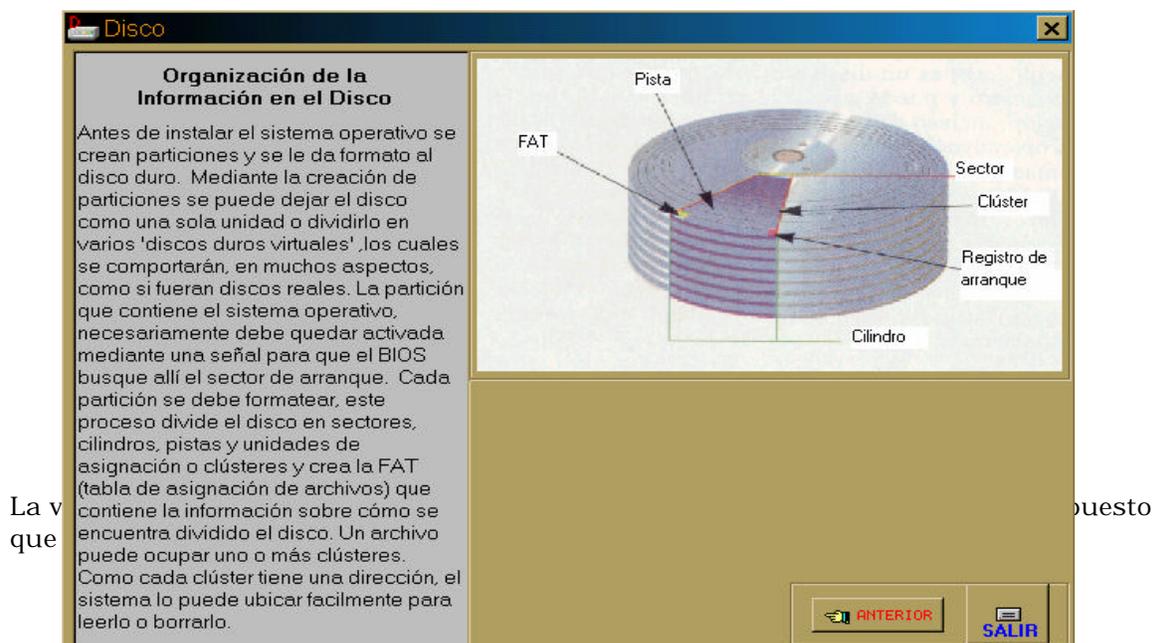
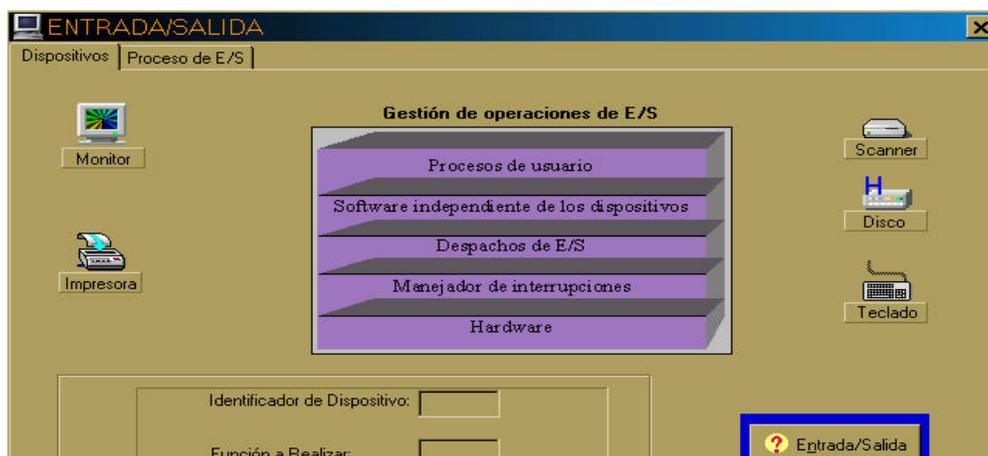


Figura 59. Disco. Organización de la información

## 2.4 Entrada / salida

### Entrada/Salida



La figura 60 muestra la forma principal de la simulación de Algunas Generalidades de Entrada / salida. Allí se muestra el identificador del dispositivo que realizó la petición y la función que va a realizar, es decir, si es una Entrada o una Salida.

Los dispositivos que realizan las peticiones son los ilustrados en la ventana de la figura 60, además, para mayor comprensión, el dispositivo que está realizando una petición en un momento dado se resalta de los demás, iluminándolo de color verde, como se observa en la figura 61.

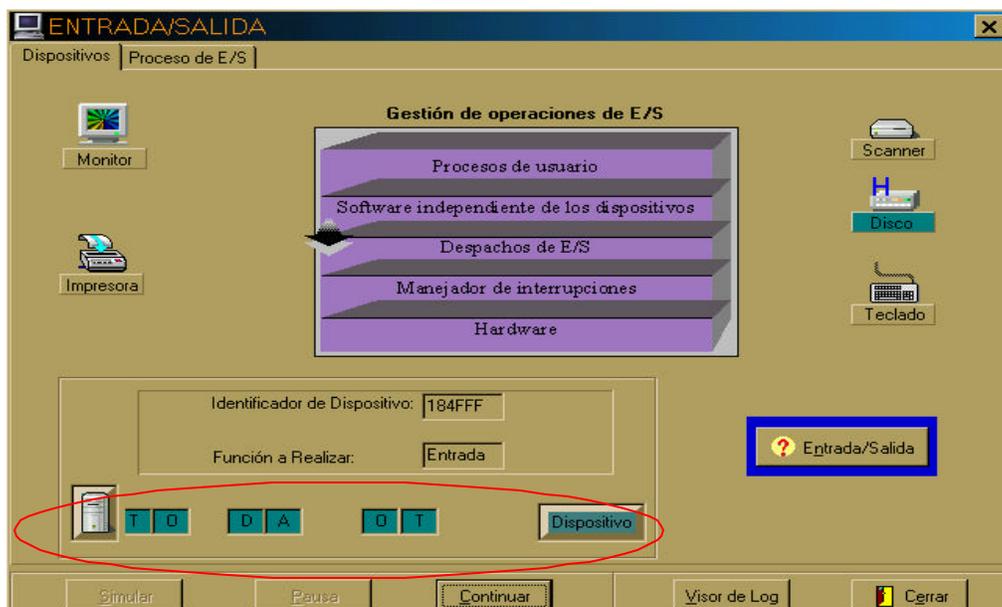


Figura 61. Gestión de Operación de E/S

En la parte resaltada en color rojo, se muestra una animación concerniente a, si el dispositivo es orientado a bloque (caso que está resaltado) o a carácter (como aparece en la figura 63).

El proceso de gestión de operaciones de Entrada / salida se encuentra representado también en la figura 61, en donde a medida que se lleva a cabo una petición, se indica a través de flechas el bloque en el que se encuentra dicha petición.

El proceso de Entrada / salida se ilustra en la figura 62, en donde se resalta en color rojo el lugar en donde Está la petición en un momento dado y se indica la dirección que seguirá dicha petición con un apuntador.

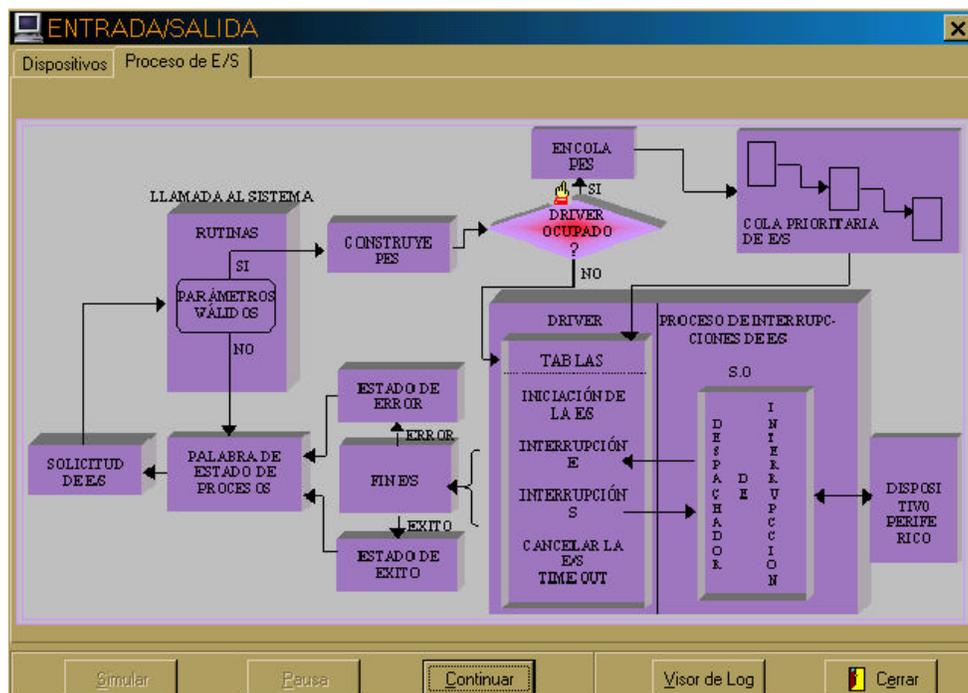


Figura 62. Proceso de E/S

En el caso particular de la figura 62, la petición encontró el Driver ocupado, así que como se indica, pasará a encolarse y luego según su prioridad podrá ser tratada por el Driver.

A través de las figuras 63 y 64, se puede apreciar la concordancia que existe entre la Gestión de operaciones de E/S y el Proceso de E/S, ya que si la petición se encuentra en la capa de Despachador de interrupciones (figura 63), debe coincidir con el estado en que se encuentre en el Proceso de E/S, en este caso se aprecia en la figura 64.

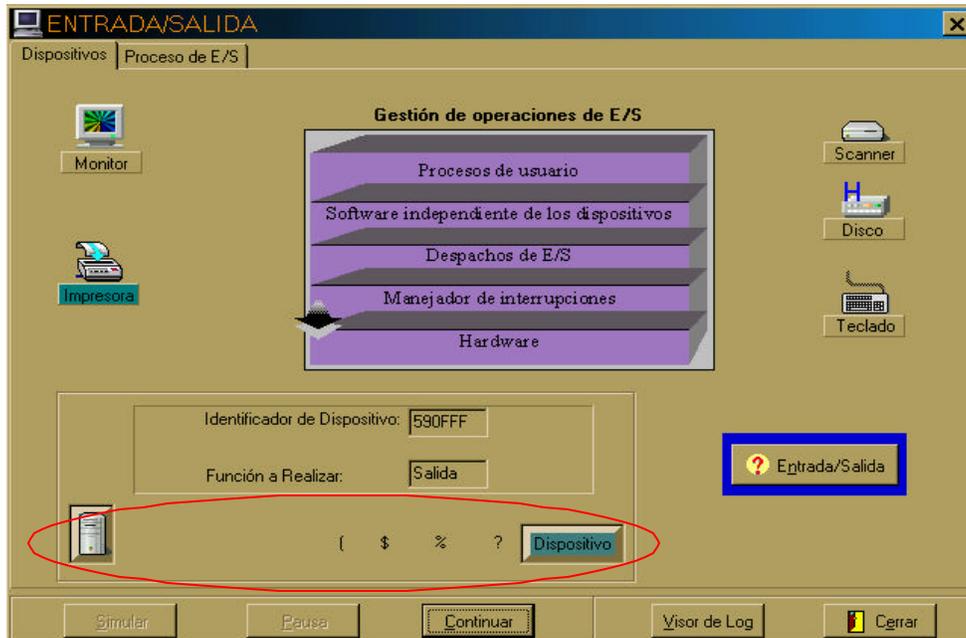
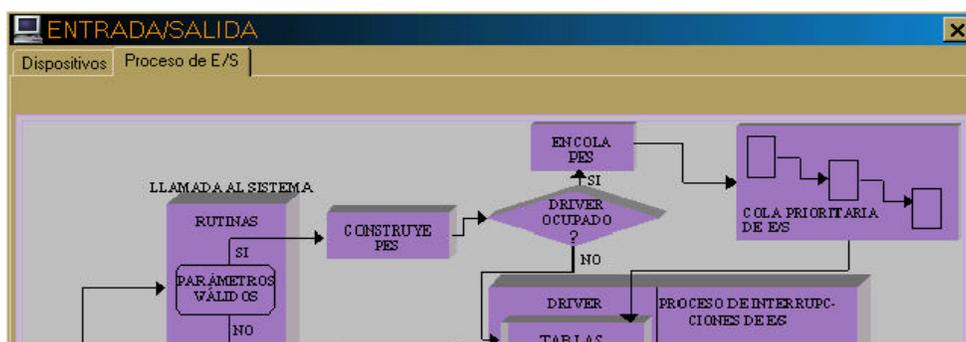


Figura 63. Gestión de operación de E/S. 2



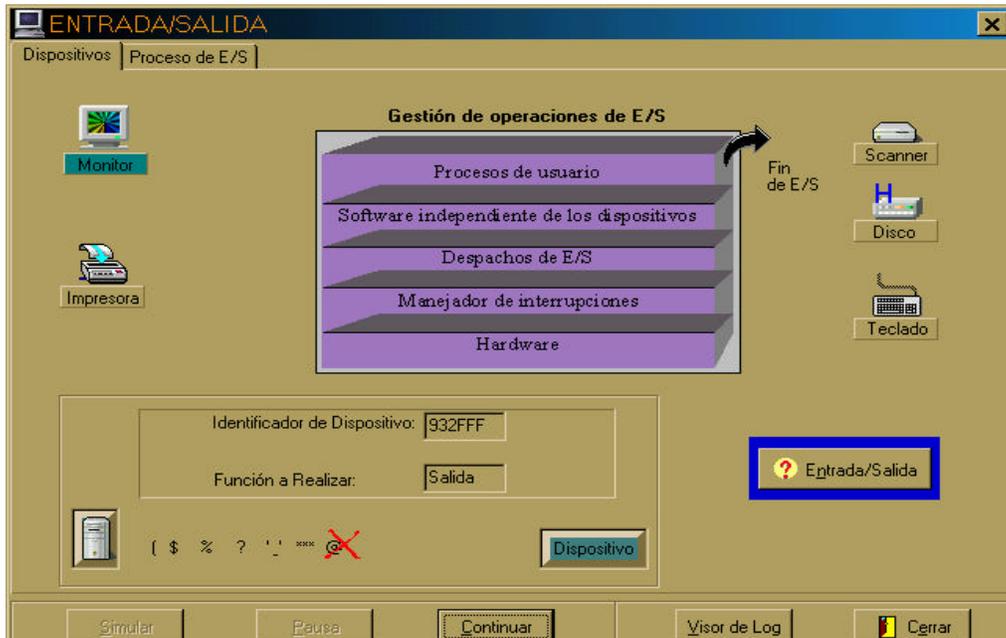
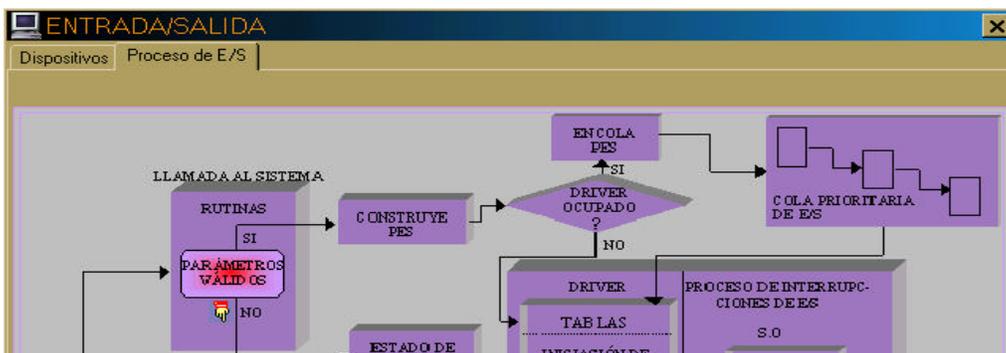


Figura 65 A. E/S. Parámetros no Válidos



En caso de que la solicitud de E/S no tenga los parámetros válidos, se interrumpe el envío de bloques o de caracteres según corresponda y se da fin a la petición, como se muestra en la figura 65 A.

En el Proceso de E/S, sucede lo ilustrado en la figura 65 B, es decir, se detecta que la petición no tiene los parámetros válidos y sigue el camino indicado por el apuntador.

Al finalizar la simulación, se activa automáticamente el Marco de Página titulado Dispositivos. Después de pulsar el botón Ok de la ventana de diálogo de finalización, se borra toda la información al respecto, excepto en el Visor de Log, recordemos que el Visor puede ser leído en cualquier momento, en él encuentra todo lo acontecido durante la simulación con las diferentes peticiones.

La figura 66 muestra el Visor de Log activo después de finalizada una simulación. En él se puede mover con las flechas de desplazamiento horizontal y vertical o con el Mouse mediante las barras de Scroll.

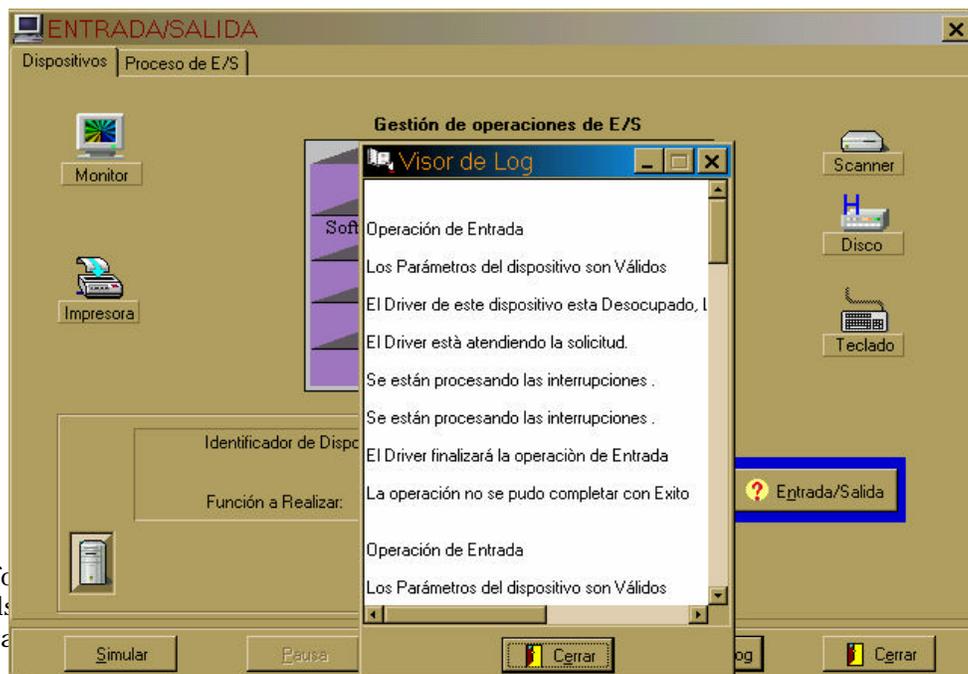


Figura 66. E/S. Visor de Log

A través del botón Entrada / salida que se encuentra en el Marco de Página llamado Dispositivos, el usuario tiene acceso a la información que aparece en la figura 67,



dicha ventana contiene información general acerca de algunos temas concernientes a la Entrada / salida.

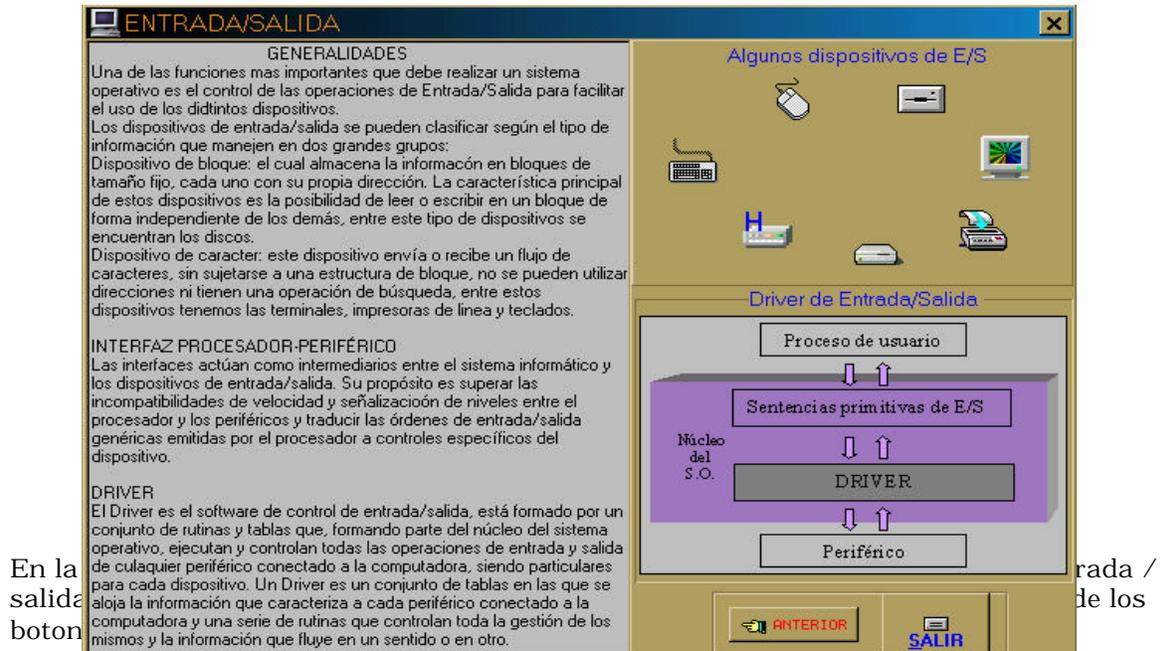


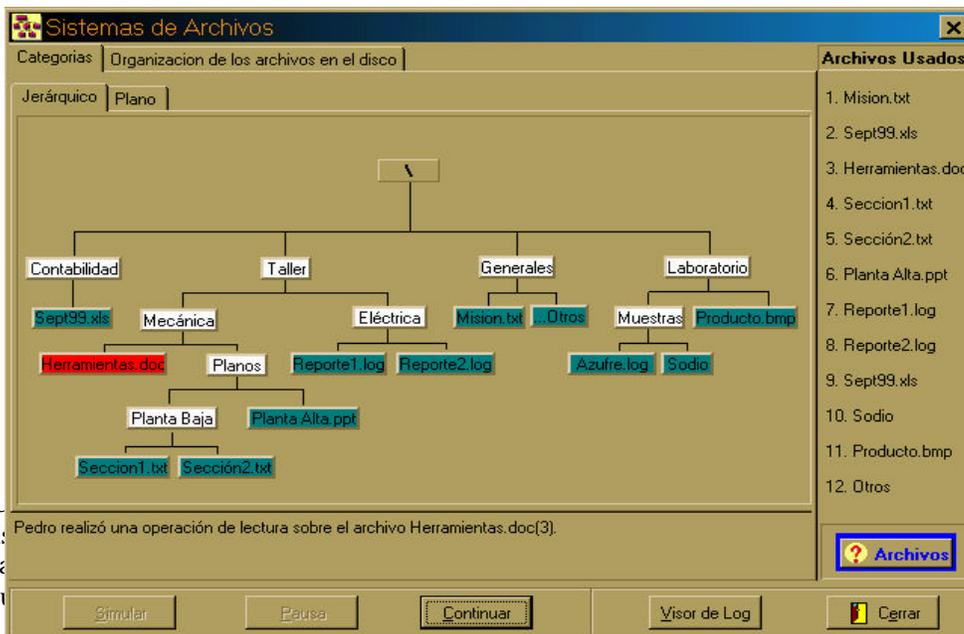
Figura 67. E/S. Información General

## 2.5 Archivos

Archivos



La simulación de los Sistemas de Archivos se encuentra en una ventana, la cual consta de dos Marcos de Página principales, titulados Categorías y Organización de los Archivos en el disco. Cada uno de estos Marcos de Página tienen a la vez dos submarcos, titulados Jerárquico y Plano y UNIX y DOS, respectivamente. Tal como lo muestran las figuras 68 y 69.

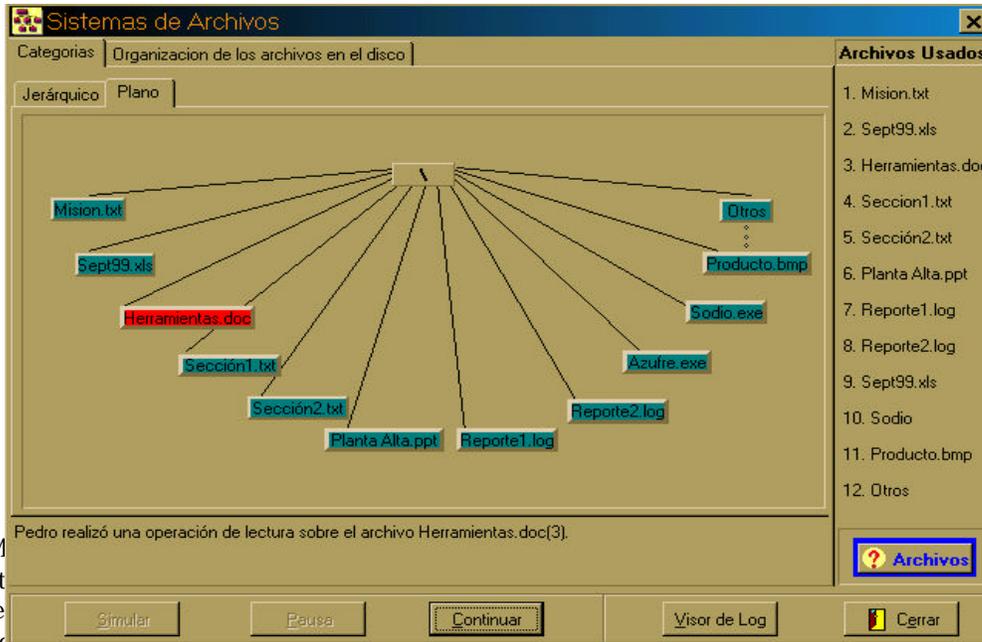


La figura 70 muestra la simulación de los Sistemas de Archivos en una ventana, la cual consta de dos Marcos de Página principales, titulados Categorías y Organización de los Archivos en el disco. Cada uno de estos Marcos de Página tienen a la vez dos submarcos, titulados Jerárquico y Plano y UNIX y DOS, respectivamente. Tal como lo muestran las figuras 68 y 69. La figura 70 muestra la simulación de los Sistemas de Archivos en una ventana, la cual consta de dos Marcos de Página principales, titulados Categorías y Organización de los Archivos en el disco. Cada uno de estos Marcos de Página tienen a la vez dos submarcos, titulados Jerárquico y Plano y UNIX y DOS, respectivamente. Tal como lo muestran las figuras 68 y 69. La figura 70 muestra la simulación de los Sistemas de Archivos en una ventana, la cual consta de dos Marcos de Página principales, titulados Categorías y Organización de los Archivos en el disco. Cada uno de estos Marcos de Página tienen a la vez dos submarcos, titulados Jerárquico y Plano y UNIX y DOS, respectivamente. Tal como lo muestran las figuras 68 y 69.

Figura 70. Archivos. Categoría Jerárquica

Cuando los archivos terminan de organizarse, los usuarios empiezan a hacer uso de ellos, esta acción aparece descrita en la parte inferior de la ventana. En el caso particular que ilustra la figura 70, el usuario Pedro realizó una operación de lectura sobre el archivo Herramientas.doc. Como se observa, cada vez que se está haciendo acceso a un archivo,

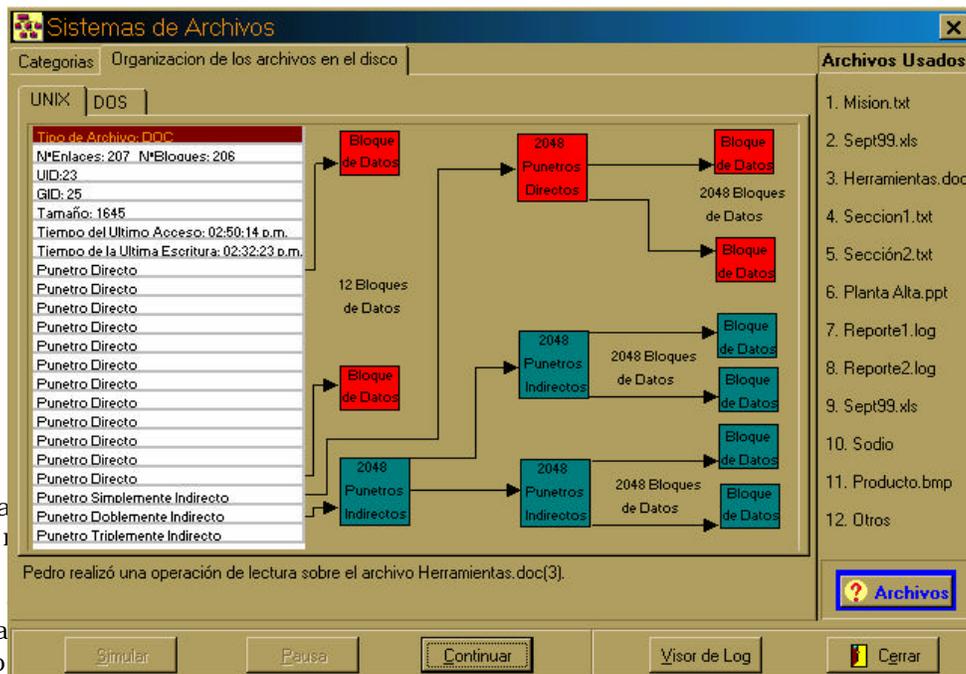
éste se resalta de los demás en color rojo. De igual manera sucede en la organización de los archivos en forma plana, como lo ilustra la figura 71.



El M  
ilust  
mue  
Nodos.

lisco, se  
ual  
a de I-

Figura 71. Archivos. Categoría Plana



En la ta  
siendo  
tipo de  
utiliza,  
grupo a  
archivo  
modificación a dicho archivo.

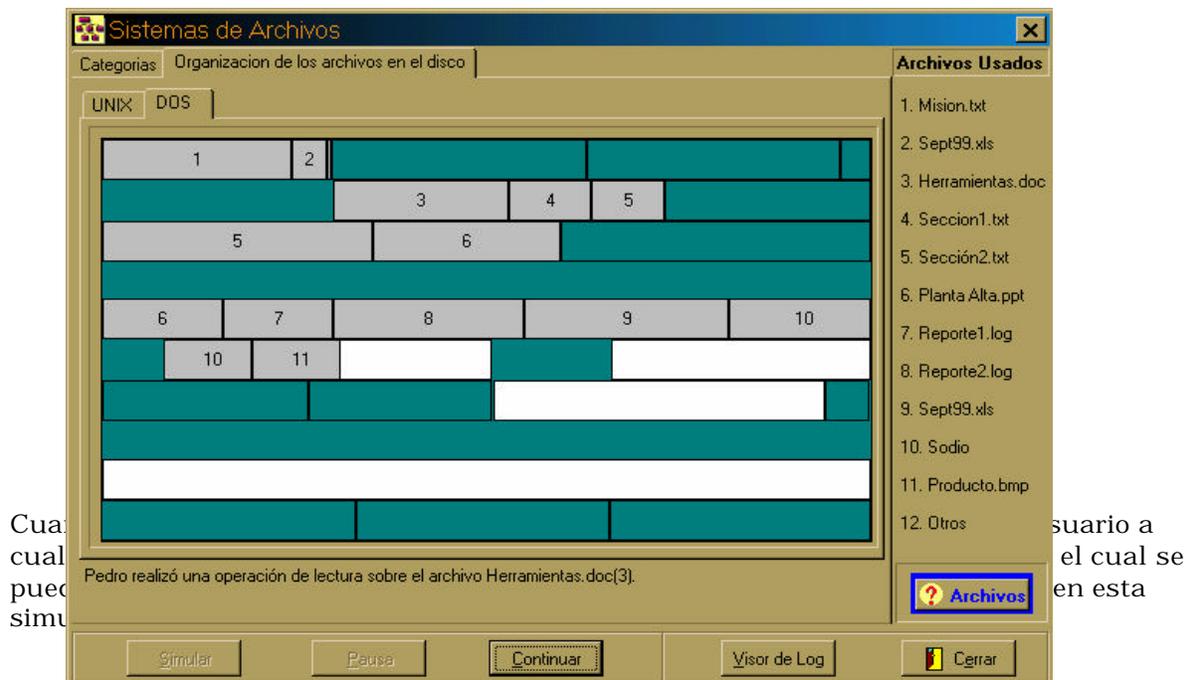
e está  
ra el  
que  
or del  
ño del  
tima

Figura 72. Archivos. Organización en S.O UNIX

La simulación también muestra gráficamente el número de bloques de datos usado por el archivo que está siendo referenciado en un momento dado. En la figura 72 se encuentran iluminados los bloques de datos que utilizan punteros directos, pues, el tamaño de herramientas.doc en este caso es de 1645K, lo cual significa que por su tamaño no necesita usar punteros simplemente indirectos, pues, no alcanza siquiera a copar los punteros directos, ya que solo usa 206 bloques de datos.

El segundo submarco de página que se encuentra en Organización de los archivos en el disco, se ilustra en la figura 73. Allí se encuentra activo el submarco de página DOS, el cual muestra la forma en que el sistema operativo DOS organiza los archivos en forma contigua.

A la derecha de la ventana se encuentra un tablero titulado Archivos Usados, en donde aparecen enumerados los nombres de los archivos contenidos en las estructuras de directorios plano y jerárquico. La numeración de los archivos corresponde a la que se encuentra en el submarco de página para el sistema operativo DOS.



Todos los espacios en blanco que aparecen en forma de triángulo en la figura 72, corresponden a espacios en disco libres para ser usados por los archivos de usuario que se manejen. El resto de los espacios en disco (color verde), está ocupados.

Al pulsar el botón **Archivos** nueva ventana, ilustrada en la figura 74, que contiene información general de la asignación de los archivos en los sistemas operativos DOS y UNIX,



En la información de la izquierda encuentra etiqueta del disco y Superbloque en color azul, al hacer click en cualquiera de las dos referencia, se despliega un pequeño tablero que contiene información al respecto.

En la información que aparece a la izquierda de la ventana, también encuentra una referencia en color azul que corresponde a la FAT, al hacer click en esa referencia, aparece la ventana que ilustra la figura 72, la cual contiene información respecto a la organización de la información en el disco y la relación que tiene con la FAT.

Puede hacer uso de los botones Salir o Anterior, ya sea para abandonar las ventana de información y regresar a la simulación de Archivos o bien para regresar a la anterior ventana.

