Práctica de Memoria

Ejercicio 1 Memoria Contigua Simple

Las soluciones a estos ejercicios pueden realizarse en pseudo código o en lenguaje C

Definir la metodología, los procesos y las estructuras de datos necesarias para implementar las siguientes simulaciones

1.1- Simular la administración de la asignación de memoria contigua simple. Pedir memoria contigua que simule la memoria RAM de la máquina. El 10 % de dicha memoria será destinada al núcleo del Sistema Operativo. Guardar en una estructura de datos los punteros al comienzo de memoria del Sistema Operativo, al final del Sistema Operativo, al comienzo de memoria de usuario, al final de memoria de usuario. Mostrar los punteros indicando a que apunta cada uno de ellos.

1.2.- Modifique el programa del ejercicio 1.- para poder cargar un proceso en la memoria contigua. Modifique la estructura de datos para que se pueda registrar el proceso cargado en memoria. Ingrese el identificador y tamaño del proceso. Si la memoria de usuario es mayor o igual al tamaño del proceso actualizar los datos de la estructura y calcular la fragmentación externa en caso contrario rechazar el proceso.

Ejercicio 2 Memoria Particionada Fija con particiones de igual tamaño

2.1.- Simular la administración de la asignación de memoria particionada fija con particiones de igual tamaño. Pedir memoria contigua que simule la memoria RAM de la máquina. El 10 % de dicha memoria será destina al núcleo del Sistema Operativo. Ingresar la cantidad de particiones fijas en la cuál se dividirá la memoria de Usuario. Generar el Id de la partición automáticamente. Actualizar las la tabla de particiones. La tabla de particiones contendrá la siguiente información (id partición, dirección de comienzo de la partición, estado de la partición). Mostrar la información contenida en la tabla de particiones.

2.2.- Agregar a la simulación anterior la posibilidad de cargar procesos en las particiones libres. El programa debe permitir ingresar procesos mientras haya memoria libre para asignar, por cada proceso ingresar (Id de proceso y tamaño del proceso). La tabla de particiones deberá contener (Id de partición, dirección de comienzo de partición, tamaño de la partición, estado de la partición, id de proceso asignado a la partición, fragmentación interna). Mostrar por cada proceso ingresado la tabla de particiones.

Ejercicio 3 Memoria Particionada Fija con particiones de tamaño variable

3.1.- Simular la administración de la asignación de memoria particionada fija con particiones de tamaño variable. Pedir memoria contigua que simule la memoria RAM de la máquina. El 10 % de dicha memoria será destina al núcleo del Sistema Operativo. Ingresar la cantidad de particiones fijas en la cuál se dividirá la memoria de Usuario. Ingresar el Id y el tamaño de cada partición. Actualizar las la tabla de particiones. La tabla de particiones contendrá la siguiente información (id partición, dirección de comienzo de la partición, tamaño de la partición, estado de la partición). Mostrar la información contenida en la tabla de particiones.

3.2.- Agregar a la simulación anterior la posibilidad de cargar procesos en las particiones libres. El programa debe permitir ingresar procesos mientras haya memoria libre para asignar, por cada proceso ingresar (Id de proceso y tamaño del proceso). La tabla de particiones deberá contener (Id de partición, dirección de comienzo de partición, tamaño de la partición, estado de la partición, id de proceso asignado a la partición, fragmentación interna). Mostrar por cada proceso ingresado la tabla de particiones. Utilizando la técnica del mejor ajuste.

Ejercicio 4 Memoria Particionada Reubicable

4.1.- Simular la administración de la asignación de memoria particionada variable reubicable. Pedir memoria contigua que simule la memoria RAM de la máquina. El 10 % de dicha memoria será destina al núcleo del Sistema Operativo. Ingresar procesos indicando el identificador y el tamaño del proceso, a medida que se ingresan los procesos actualizar la tabla de particiones libres y la tabla de particiones ocupadas. La tabla de particiones libres contendrá la siguiente información (id partición, dirección de comienzo de la partición, tamaño de la partición). La tabla de particiones ocupadas contendrá la siguiente información (id partición, dirección de comienzo de la partición, tamaño de la partición, id del proceso).Mostrar la información contenida en la tabla de particiones y la tabla de particiones ocupadas.

Ejercicio 5 Memoria Paginada Simple (Un Proceso)

5.1.- Simular la administración de la asignación de memoria paginada simple para un proceso.

Algunos pasos a tener en cuenta para la construcción de la simulación.
a) Definir una tabla de páginas para un proceso utilizando una estructura que contenga (Número de página y Número de Marco).
b) Definir un vector de marcos donde la posición del elemento coincide con un número de marco y el elemento indica si ese marco está asignado o no (1 o 0).
c) Ingresar cantidad de memoria RAM disponible.
d) Pedir memoria utilizando malloc.
e) Ingresar el tamaño del proceso.
f) Definir páginas y marcos de 32 bytes.
g) Definir un vector de punteros al comienzo de cada marco de memoria
h) Asignar cada página del proceso a marcos de memoria y actualizar la tabla de páginas y el vector de información de asignación de marcos.
i) Mostrar el contenido de todas las estructuras de información.

Ejercicio 6 Memoria Paginada Simple (N Procesos)

6.1.- Simular la administración de la asignación de memoria paginada simple para N procesos.

Algunos pasos a tener en cuenta para la construcción del programa.

a) Definir una tabla de páginas para cada uno de los procesos utilizando estructuras, donde cada una de ellas contiene (Número de página y Número de marco).
b) Definir un vector de marcos donde la posición del elemento coincide con un número de marco y el elemento indica si ese marco está asignado o no (1 o 0).
c) Ingresar una cantidad de memoria RAM disponible.
d) Pedir memoria utilizando malloc para simular la memoria RAM.
e) Ingresar N que es la cantidad de procesos. Por cada proceso ingresar el identificador de proceso y el tamaño del proceso.
f) Definir páginas y marcos de 32 bytes.
g) Definir un vector de punteros al comienzo de cada marco de memoria
h) Asignar cada página de cada proceso a marcos de memoria y actualizar las tablas de páginas y el vector de marcos asignados.
i) Mostrar el contenido de todas las estructuras de información.