

Recreación de administración de memoria particionada fija con particiones de distinto tamaño

Para resolver el siguiente ejercicio, como primera medida se deberá leer la teoría del tema “Administración de memoria particionada fija con particiones de distinto tamaño” de la bibliografía de la materia.

El objetivo del siguiente ejercicio es recrear los conceptos teóricos del tema propuesto, creando en lenguaje C bajo Linux una pequeña aplicación que permita administrar espacio de memoria compartida como si fuera una memoria particionada fija con particiones de distinto tamaño.

Hay que tener en cuenta distintos aspectos, por un lado las estructuras de datos que mantendrán la información de las particiones, por ejemplo si la partición esta libre u ocupada, su tamaño, la dirección de memoria base (adecuarlo a la teoría).

Otro aspecto a tener en cuenta son los procesos de asignación de partición, primer ajuste, mejor ajuste, peor ajuste.

Las funciones que nuestra pequeña aplicación debería implementar son:

- 1) Crear memoria compartida de un tamaño, este valor de tamaño se puede ingresar por teclado.
- 2) Crear un archivo de configuración en el file system, el cual debería tener mínimamente el número de la partición y su tamaño.
- 3) Leer el archivo de configuración y crear en memoria estática o dinámica una estructura de datos que puede ser una matriz, un vector, una lista, una pila, o una cola. Esta estructura estática o dinámica será contendrá la información de administración de la memoria compartida, esta estructura no solo contiene el número de partición y su tamaño, también contiene un indicador de partición libre u ocupada, la dirección de memoria base de la partición y una relación al proceso que la ocupa.
- 4) Cargar procesos en memoria, esta función permitirá ingresar el nombre de un proceso representado por una letra mayúscula y su tamaño en bytes (adecuar los tamaños de los procesos a la realidad de este ejercicio, por ejemplo proceso A 30 Bytes).
- 5) Terminar un proceso, se indica el nombre del proceso que debe terminar.
- 6) Mostrar la estructura de datos de administración por pantalla.

- 7) Mostrar la fragmentación interna de cada partición por pantalla.
- 8) Mostrar el contenido de la memoria compartida por pantalla.

Ideas que pueden ayudar:

- 1) La memoria compartida se puede inicializar con el carácter * (asterisco).
- 2) Cuando se carga un proceso, este se representa en la partición de memoria seleccionada dentro de la memoria compartida donde se ubicará, completando con la letra nombre del proceso tantas veces como su tamaño lo indica. Por ejemplo si el proceso B demanda 20 Bytes y suponiendo que el tamaño de la partición que lo contiene es de 30 Bytes, entonces su contenido es
BBBBBBBBBBBBBBBBBBBB*****
- 3) Deberíamos tener una estructura que contenga los datos de los procesos, por ejemplo `typedef struct task_struct { char nombre; int tamaño;};` y luego un vector de `task_struct` de la forma `task_struct procesos[P];` podríamos pensar la estructura de administración de memoria de la misma manera.
- 4) Usar todas las funciones utilizando un menú de opciones.
- 5) Elegir una política de asignación de partición (primer ajuste, mejor ajuste o peor ajuste);

Consultas a

juancarlosjromer@gmail.com