

Universidad Nacional Autónoma de Honduras
en el Valle de Sula



Facultad de Ingeniería

Departamento de Ingeniería en Sistemas

Informe proyecto final simulación de memoria

Asignatura: Sistemas operativos II 1800

Catedrático: Ing. Mario Pon

Alumno: José Daniel Sánchez Madrid

Cuenta: 20212020720

25 de abril 2024

Marco teórico

Particionamiento fijo de memoria: para este proyecto se permite al usuario especificar la cantidad de particiones en que se segmentará la memoria y el tamaño de cada partición, se llama particionamiento fijo ya que el tamaño especificado a cada partición no cambiará, una ventaja de este enfoque es que reduce la fragmentación externa.

Políticas

- Primer ajuste: Es la política mas sencilla, toma el proceso a ubicar y recorre la memoria en busca de una partición en la cual el proceso se pueda alojar (de mayor o igual tamaño que el proceso), sin importar si esta partición es demasiado grande para este proceso lo cual a veces mala utilización de particiones grandes de la memoria.
- Mejor ajuste: Tomando en cuenta la memoria que requiere un proceso, con esta política se analizan todas las particiones y se decide asignar el proceso en la partición más pequeña en la cual este quepa, en caso de haber dos o más particiones iguales, se usa la política de primer ajuste

Programa

Llenando información inicial.

1. Digitar una cantidad de memoria total.
2. Seleccionar la política a utilizar.
3. Digitar un intervalo de tiempo
4. Escribe la cantidad de particiones y presiona el botón “Crear”
5. Luego especifica el tamaño de cada partición, por defecto estas tienen el mismo tamaño.
6. Presiona “Siguiente”

Registrando procesos e iniciando simulación.

1. Llena los campos de memoria requerida y nombre del primer proceso, luego.

2. Presiona “Agregar Proceso” para un nuevo proceso o “Iniciar Simulación”.

Simbología:

- ParticiónID: Es una referencia hacia las particiones que cada proceso guarda será -1 cuando este proceso está en espera de memoria, de lo contrario tendrá el valor del número de la partición a la cual está asignado.
- Estado: será de valor 1 cuando el proceso está en ejecución, 2 cuando está en espera de memoria, 3 cuando ha terminado.

Errores:

- El programa mostrará errores con mensajes de advertencias significativas para el usuario como la suma de las particiones no coincide con el total de memoria (especificando la diferencia).
- La cantidad máxima de particiones en cuanto a la memoria no debe ser mayor al 10% y no mayor a 10 particiones.
- No se admiten tamaños negativos de particiones
- Antes de agregar otra fila para otro proceso o para iniciar simulación se debe completar los datos de todas las filas que se hayan creado hasta el momento.
- El tamaño del proceso es demasiado grande y no podrá ser ejecutado.
- Ingrese solo números en la columna “Memoria requerida”.

Repositorio del proyecto

Para su registro en este repositorio se ha llevado el control del proyecto mediante un repositorio remoto de GitHub

<https://github.com/DMadrid03/ProyectoSO2.git>

Conclusiones

- Tras la implementación y análisis de las políticas de ubicación (primer ajuste y mejor ajuste), se observa que la elección de la política afecta significativamente el rendimiento del sistema.
- La implementación de una cola de procesos "en espera de memoria" junto con la política de primer en entrar, primer en ser servido (PEPS) permite una gestión dinámica de la asignación de memoria. Esta funcionalidad garantiza que los procesos no sean bloqueados indefinidamente, incluso si no pueden asignarse inmediatamente a una partición de memoria, lo que mejora la eficiencia y la fluidez del sistema.
- La política de mejor ajuste muestra una mejor utilización de la memoria al reducir la fragmentación interna, lo que se traduce en una mayor capacidad para alojar procesos en la memoria disponible.
- La representación gráfica de la memoria y los procesos proporciona una experiencia de usuario intuitiva y facilita la comprensión del estado del sistema en tiempo real.

Bibliografía

- Deitel, P. J., & Deitel, H. M. (2005). Cómo Programar en C# (2da ed.). Pearson Educación.
- Stallings, W. (2005). Sistemas operativos: aspectos internos y principios de diseño (7ma ed.). Pearson Educación.