

Manejo de Memoria - Fragmentación

Silvio Vileriño

09 de octubre de 2014

Contexto de la clase

Sistemas operativos - Post-Teórica Manejo Memoria

- Este ejercicio es de la practica 4 de Sistemas Operativos (Manejo de memoria)

Contexto de la clase

Sistemas operativos - Post-Teórica Manejo Memoria

- Este ejercicio es de la practica 4 de Sistemas Operativos (Manejo de memoria)
- Clase practica a darse luego de la teórica de Manejo de memoria.

Contexto de la clase

Sistemas operativos - Post-Teórica Manejo Memoria

- Este ejercicio es de la practica 4 de Sistemas Operativos (Manejo de memoria)
- Clase practica a darse luego de la teórica de Manejo de memoria.
- Repaso rápido de fragmentación y algoritmos de asignación de memoria.

Contexto de la clase

Sistemas operativos - Post-Teórica Manejo Memoria

- Este ejercicio es de la practica 4 de Sistemas Operativos (Manejo de memoria)
- Clase practica a darse luego de la teórica de Manejo de memoria.
- Repaso rápido de fragmentación y algoritmos de asignación de memoria.
- El ejercicio elegido para la PO integra los conceptos de fragmentación y algoritmos de asignación de memoria.

Contexto de la clase

Sistemas operativos - Post-Teórica Manejo Memoria

- Este ejercicio es de la practica 4 de Sistemas Operativos (Manejo de memoria)
- Clase practica a darse luego de la teórica de Manejo de memoria.
- Repaso rápido de fragmentación y algoritmos de asignación de memoria.
- El ejercicio elegido para la PO integra los conceptos de fragmentación y algoritmos de asignación de memoria.
- Dejo como tarea buscar un contraejemplo de algo que al principio puede no ser muy intuitivo y es algo que considero importante que los alumnos puedan entender por su cuenta, y de ser necesario pueden venir a consultar luego.

Uso eficiente de la memoria

Fragmentación

La fragmentación es un fenómeno en el cual el espacio de memoria es usado de forma ineficiente, reduciendo la capacidad y/o la performance del sistema.

Uso eficiente de la memoria

Fragmentación

La fragmentación es un fenómeno en el cual el espacio de memoria es usado de forma ineficiente, reduciendo la capacidad y/o la performance del sistema.

- **Fragmentación interna:** Ocurre cuando un bloque de memoria asignado no es aprovechado en su totalidad

Uso eficiente de la memoria

Fragmentación

La fragmentación es un fenómeno en el cual el espacio de memoria es usado de forma ineficiente, reduciendo la capacidad y/o la performance del sistema.

- **Fragmentación interna:** Ocurre cuando un bloque de memoria asignado no es aprovechado en su totalidad
- **Fragmentación externa:** Este fenómeno ocurre cuando la memoria libre total (suma de todos los bloques libres dispersos) es mayor o igual que la memoria requerida por un proceso pero no puede cumplirse al petición dado que no existe un bloque contiguo de memoria libre de tamaño mayor o igual al requerido.

Uso eficiente de la memoria

Fragmentación

La fragmentación es un fenómeno en el cual el espacio de memoria es usado de forma ineficiente, reduciendo la capacidad y/o la performance del sistema.

- **Fragmentación interna:** Ocurre cuando un bloque de memoria asignado no es aprovechado en su totalidad
- **Fragmentación externa:** Este fenómeno ocurre cuando la memoria libre total (suma de todos los bloques libres dispersos) es mayor o igual que la memoria requerida por un proceso pero no puede cumplirse al petición dado que no existe un bloque contiguo de memoria libre de tamaño mayor o igual al requerido.

Notar que diferentes tipos de fragmentación pueden ocurrir al mismo tiempo.

Uso eficiente de la memoria

Algoritmos de asignación de memoria

Consideremos un sistema que utiliza particiones de memoria en bloques de tamaño variable.

Recordemos rápidamente los algoritmos de asignación de memoria.

Uso eficiente de la memoria

Algoritmos de asignación de memoria

Consideremos un sistema que utiliza particiones de memoria en bloques de tamaño variable.

Recordemos rápidamente los algoritmos de asignación de memoria.

- **First fit:** Elegir el primer bloque disponible para responder la petición de memoria.

Uso eficiente de la memoria

Algoritmos de asignación de memoria

Consideremos un sistema que utiliza particiones de memoria en bloques de tamaño variable.

Recordemos rápidamente los algoritmos de asignación de memoria.

- **First fit:** Elegir el primer bloque disponible para responder la petición de memoria.
- **Best fit:** Recorrer toda la lista de bloques disponibles y elegir el que **mas** se ajuste a la memoria requerida en la petición **minimizando la fragmentación interna**.

Uso eficiente de la memoria

Algoritmos de asignación de memoria

Consideremos un sistema que utiliza particiones de memoria en bloques de tamaño variable.

Recordemos rápidamente los algoritmos de asignación de memoria.

- **First fit:** Elegir el primer bloque disponible para responder la petición de memoria.
- **Best fit:** Recorrer toda la lista de bloques disponibles y elegir el que **mas** se ajuste a la memoria requerida en la petición **minimizando la fragmentación interna**.
- **Worst fit:** Recorrer toda la lista de bloques disponibles y elegir el que **menos** se ajuste a la memoria requerida en la petición **maximizando la fragmentación interna**.

Ejercicio 4.3

Enunciado

Se tiene un sistema con 16 MB de RAM que utiliza particiones fijas para ubicar a los programas en memoria.

- Cuenta con particiones de los siguientes tamaños: 8 MB, 1 MB, 4 MB, 512 KB, 512 KB, 2 MB, en ese orden.
- Se desean ejecutar 5 programas de los siguientes tamaños: 500 KB, 6 MB, 3 MB, 20 KB, 2 MB, en ese orden.
- 1. Indique como asignaría las particiones utilizando best-fit. ¿Cual es la cantidad de bytes de memoria desperdiciados?
- 2. ¿Alguna de las estrategias de asignación vistas en clase (peor-ajuste, primer-ajuste) produce como resultado la imposibilidad de ejecutar los 5 programas a la vez?

Ejercicio 4.3

Best fit

Veamos que ocurre con la fragmentación en esta configuración de la memoria utilizando el algoritmo best fit

Ejercicio 4.3

Best fit

Veamos que ocurre con la fragmentación en esta configuración de la memoria utilizando el algoritmo best fit

- 1 Ubicamos el programa de 500Kb en un bloque de 512Kb, pues es el mas indicado según el algoritmo best-fit. Aquí tenemos 12 bytes desperdiciados, **produciendo fragmentación interna**.

Ejercicio 4.3

Best fit

Veamos que ocurre con la fragmentación en esta configuración de la memoria utilizando el algoritmo best fit

- 1 Ubicamos el programa de 500Kb en un bloque de 512Kb, pues es el mas indicado según el algoritmo best-fit. Aquí tenemos 12 bytes desperdiciados, **produciendo fragmentación interna.**
- 2 Ubicamos el programa de 6Mb en un bloque de 8Mb. **2 Mb se desperdician debido a la fragmentación interna.**

Ejercicio 4.3

Best fit

Veamos que ocurre con la fragmentación en esta configuración de la memoria utilizando el algoritmo best fit

- ① Ubicamos el programa de 500Kb en un bloque de 512Kb, pues es el mas indicado según el algoritmo best-fit. Aquí tenemos 12 bytes desperdiciados, **produciendo fragmentación interna.**
- ② Ubicamos el programa de 6Mb en un bloque de 8Mb. **2 Mb se desperdician debido a la fragmentación interna.**
- ③ El programa de 3Mb va al bloque de 4Mb libre. **Nuevamente perdemos 1Mb.**

Ejercicio 4.3

Best fit

Veamos que ocurre con la fragmentación en esta configuración de la memoria utilizando el algoritmo best fit

- 1 Ubicamos el programa de 500Kb en un bloque de 512Kb, pues es el mas indicado según el algoritmo best-fit. Aquí tenemos 12 bytes desperdiciados, **produciendo fragmentación interna.**
- 2 Ubicamos el programa de 6Mb en un bloque de 8Mb. **2 Mb se desperdician debido a la fragmentación interna.**
- 3 El programa de 3Mb va al bloque de 4Mb libre. **Nuevamente perdemos 1Mb.**
- 4 El programa de 20Kb va al bloque de 512Kb libre. **Desperdiciamos 492Kb.**

Ejercicio 4.3

Best fit

Veamos que ocurre con la fragmentación en esta configuración de la memoria utilizando el algoritmo best fit

- ① Ubicamos el programa de 500Kb en un bloque de 512Kb, pues es el mas indicado según el algoritmo best-fit. Aquí tenemos 12 bytes desperdiciados, **produciendo fragmentación interna.**
- ② Ubicamos el programa de 6Mb en un bloque de 8Mb. **2 Mb se desperdician debido a la fragmentación interna.**
- ③ El programa de 3Mb va al bloque de 4Mb libre. **Nuevamente perdemos 1Mb.**
- ④ El programa de 20Kb va al bloque de 512Kb libre. **Desperdiciamos 492Kb.**
- ⑤ El programa de 2Mb va al bloque de 2Mb libre. **Aquí no tenemos fragmentación interna.**

Ejercicio 4.3

Best fit

Veamos que ocurre con la fragmentación en esta configuración de la memoria utilizando el algoritmo best fit

- ① Ubicamos el programa de 500Kb en un bloque de 512Kb, pues es el mas indicado según el algoritmo best-fit. Aquí tenemos 12 bytes desperdiciados, **produciendo fragmentación interna.**
- ② Ubicamos el programa de 6Mb en un bloque de 8Mb. **2 Mb se desperdician debido a la fragmentación interna.**
- ③ El programa de 3Mb va al bloque de 4Mb libre. **Nuevamente perdemos 1Mb.**
- ④ El programa de 20Kb va al bloque de 512Kb libre. **Desperdiciamos 492Kb.**
- ⑤ El programa de 2Mb va al bloque de 2Mb libre. **Aquí no tenemos fragmentación interna.**

Tenemos en total 3576 bytes desperdiciados.

Ejercicio 4.3

Seguimiento Best fit

Best Fit

8Mb	4Mb	2Mb	1	5	5
			1	1	1
			2	2	2
			M	K	K
			b	b	b

8Mb	4Mb	2Mb	1	5	5
			1	0	1
			2	0	2
			M	K	K
			b	b	b

6Mb	4Mb	2Mb	1	3	5
			1	0	1
			2	0	2
			M	K	K
			b	b	b

6Mb	3Mb	2Mb	1	3	5
			1	0	1
			2	0	2
			M	K	K
			b	b	b

6Mb	3Mb	2Mb	1	5	2
			1	0	0
			2	0	0
			M	K	K
			b	b	b

6Mb	3Mb	2Mb	1	3	2
			1	0	0
			2	0	0
			M	K	K
			b	b	b

Ejercicio 4.3

Enunciado

Se tiene un sistema con 16 MB de RAM que utiliza particiones fijas para ubicar a los programas en memoria.

- Cuenta con particiones de los siguientes tamaños: 8 MB, 1 MB, 4 MB, 512 KB, 512 KB, 2 MB, en ese orden.
- Se desean ejecutar 5 programas de los siguientes tamaños: 500 KB, 6 MB, 3 MB, 20 KB, 2 MB, en ese orden.
- 1. Indique como asignaría las particiones utilizando best-fit. ¿Cual es la cantidad de bytes de memoria desperdiciados?
- 2. ¿Alguna de las estrategias de asignación vistas en clase (peor-ajuste, primer-ajuste) produce como resultado la imposibilidad de ejecutar los 5 programas a la vez?

Ejercicio 4.3

Seguimiento First Fit

La idea es que acá en el pizarron interactue con los alumnos, para resolver el ejercicio entre todos y luego vemos la resolución.

First Fit

8Mb	4Mb	2Mb	1	5	5
			1	1	1
			2	2	2
			M	K	K
			b	b	b

500 Kb	4Mb	2Mb	1	5	5
			1	1	1
			2	2	2
			M	K	K
			b	b	b

El programa de 6Mb no puede asignarse debido a fragmentacion externa(asumiendo no merge entre 2 bloques contiguos)

500 Kb	3Mb	2Mb	1	5	5
			1	1	1
			2	2	2
			M	K	K
			b	b	b

500 Kb	3Mb	20 Kb	1	5	5
			1	1	1
			2	2	2
			M	K	K
			b	b	b

El programa de 2Mb no puede asignarse pues no hay espacio suficiente disponible(suma de todos los bloques)

Ejercicio 4.3

Seguimiento Worst Fit

La idea es que acá en el pizarron interactue con los alumnos, para resolver el ejercicio entre todos y luego vemos la resolución.

Worst Fit

8Mb	4Mb	2Mb	1	5	1
			1	1	1
			2	2	2
			M	K	K
			b	b	b

500 Kb	4Mb	2Mb	1	5	1
			1	1	1
			2	2	2
			M	K	K
			b	b	b

El programa de 6Mb no puede asignarse debido a fragmentacion externa(asumiendo no merge entre 2 bloques contiguos)

500 Kb	3 Mb	2Mb	1	5	1
			1	1	1
			2	2	2
			M	K	K
			b	b	b

500 Kb	3 Mb	20 Kb	1	5	1
			1	1	1
			2	2	2
			M	K	K
			b	b	b

El programa de 2Mb no puede asignarse pues no hay espacio suficiente disponible(suma de todos los bloques)

Ejercicio 4.3

Continuación - Conclusión

- 1 Ya vimos que ocurre con los algoritmos worst-fit y first-fit en las slides anteriores.

Ejercicio 4.3

Continuación - Conclusión

- ① Ya vimos que ocurre con los algoritmos worst-fit y first-fit en las slides anteriores.
- ② A diferencia del algoritmo best fit **en este ejemplo particular**, con los otros algoritmos hay programas que no pueden ser cargados en memoria debido a pérdida de memoria por problemas de fragmentación.

Ejercicio 4.3

Continuación - Conclusión

- 1 Ya vimos que ocurre con los algoritmos worst-fit y first-fit en las slides anteriores.
- 2 A diferencia del algoritmo best fit **en este ejemplo particular**, con los otros algoritmos hay programas que no pueden ser cargados en memoria debido a pérdida de memoria por problemas de fragmentación.
- 3 Pensar en los tiempos de ejecución de best-fit/worst-fit versus first-fit. (A menos que los bloques estén ordenados por tamaño)

Ejercicio 4.3

Continuación - Conclusión

- 1 Ya vimos que ocurre con los algoritmos worst-fit y first-fit en las slides anteriores.
- 2 A diferencia del algoritmo best fit **en este ejemplo particular**, con los otros algoritmos hay programas que no pueden ser cargados en memoria debido a pérdida de memoria por problemas de fragmentación.
- 3 Pensar en los tiempos de ejecución de best-fit/worst-fit versus first-fit. (A menos que los bloques estén ordenados por tamaño)
- 4 **Notar que no siempre best fit es el algoritmo que mejor resuelve el problema en todos los casos.** Buscar como tarea un ejemplo de esto.

¿Preguntas?