

Manejo de Memoria - Fragmentacion

Silvio Vileriño

09 de octubre de 2014

Contexto de la clase

Sistemas operativos - Post-Teorica Manejo Memoria

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

Contexto de la clase

Sistemas operativos - Post-Teorica Manejo Memoria

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

Contexto de la clase

Sistemas operativos - Post-Teorica Manejo Memoria

- Este ejercicio es de la practica 4 de Sistemas Operativos (Manejo de memoria)
- Clase practica a darse luego de la teorica de Manejo de memoria.
- Repaso rapido de fragmentacion y algoritmos de asignacion de memoria.

Contexto de la clase

Sistemas operativos - Post-Teorica Manejo Memoria

- Este ejercicio es de la practica 4 de Sistemas Operativos (Manejo de memoria)
- Clase practica a darse luego de la teorica de Manejo de memoria.
- Repaso rapido de fragmentacion y algoritmos de asignacion de memoria.
- El ejercicio elegido para la PO integra los conceptos de fragmentacion y algoritmos de asignacion de memoria.

Contexto de la clase

Sistemas operativos - Post-Teorica Manejo Memoria

- Este ejercicio es de la practica 4 de Sistemas Operativos (Manejo de memoria)
- Clase practica a darse luego de la teorica de Manejo de memoria.
- Repaso rapido de fragmentacion y algoritmos de asignacion de memoria.
- El ejercicio elegido para la PO integra los conceptos de fragmentacion y algoritmos de asignacion 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

Fragmentacion

La fragmentacion es un fenomeno 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

Fragmentacion

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

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

Uso eficiente de la memoria

Fragmentacion

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

- **Fragmentacion interna:** Ocurre cuando un bloque de memoria asignado no es aprovechado en su totalidad
- **Fragmentacion externa:** Este fenomeno 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 peticion dado que no existe un bloque contiguo de memoria libre de tamaño mayor o igual al requerido.

Uso eficiente de la memoria

Fragmentacion

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

- **Fragmentacion interna:** Ocurre cuando un bloque de memoria asignado no es aprovechado en su totalidad
- **Fragmentacion externa:** Este fenomeno 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 peticion dado que no existe un bloque contiguo de memoria libre de tamaño mayor o igual al requerido.

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

Uso eficiente de la memoria

Algoritmos de asignacion de memoria

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

Recordemos rapidamente los algoritmos de asignacion de memoria.

Uso eficiente de la memoria

Algoritmos de asignacion de memoria

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

Recordemos rapidamente los algoritmos de asignacion de memoria.

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

Uso eficiente de la memoria

Algoritmos de asignacion de memoria

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

Recordemos rapidamente los algoritmos de asignacion de memoria.

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

Uso eficiente de la memoria

Algoritmos de asignacion de memoria

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

Recordemos rapidamente los algoritmos de asignacion de memoria.

- **First fit:** Elegir el primer bloque disponible para responder la peticion de memoria.
- **Best fit:** Recorrer toda la lista de bloques disponibles y elegir el que **mas** se ajuste a la memoria requerida en la peticion **minimizando la fragmentacion interna**.
- **Worst fit:** Recorrer toda la lista de bloques disponibles y elegir el que **menos** se ajuste a la memoria requerida en la peticion **maximizando la fragmentacion 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 asignaria las particiones utilizando best-fit. ¿Cual es la cantidad de bytes de memoria desperdiciados?
- 2. ¿Alguna de las estrategias de asignacion 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 fragmentacion en esta configuracion de la memoria utilizando el algoritmo best fit

Ejercicio 4.3

Best fit

Veamos que ocurre con la fragmentacion en esta configuracion de la memoria utilizando el algoritmo best fit

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

Ejercicio 4.3

Best fit

Veamos que ocurre con la fragmentacion en esta configuracion de la memoria utilizando el algoritmo best fit

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

Ejercicio 4.3

Best fit

Veamos que ocurre con la fragmentacion en esta configuracion de la memoria utilizando el algoritmo best fit

- ① Ubicamos el programa de 500Kb en un bloque de 512Kb, pues es el mas indicado segun el algoritmo best-fit. Aqui tenemos 12 bytes desperdiciados, **produciendo fragmentacion interna.**
- ② Ubicamos el programa de 6Mb en un bloque de 8Mb. **2 Mb se desperdician debido a la fragmentacion 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 segun el algoritmo best-fit. Aqui 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 segun 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 fragmentacion en esta configuracion de la memoria utilizando el algoritmo best fit

- ① Ubicamos el programa de 500Kb en un bloque de 512Kb, pues es el mas indicado segun el algoritmo best-fit. Aqui tenemos 12 bytes desperdiciados, **produciendo fragmentacion interna.**
- ② Ubicamos el programa de 6Mb en un bloque de 8Mb. **2 Mb se desperdician debido a la fragmentacion 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. **Aqui no tenemos fragmentacion 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

Seguimiento First Fit

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

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 aca en el pizarron interactue con los alumnos, para resolver el ejercicio entre todos y luego vemos la resolucion.

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

Continuacion - Conclusion

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

Ejercicio 4.3

Continuacion - Conclusion

- 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 perdida de memoria por problemas de fragmentacion.

Ejercicio 4.3

Continuacion - Conclusion

- 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 perdida de memoria por problemas de fragmentacion.
- 3 Pensar en los tiempos de ejecucion de best-fit/worst-fit versus first-fit.(A menos que los bloques esten ordenados por tamaño)

Ejercicio 4.3

Continuacion - Conclusion

- 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 perdida de memoria por problemas de fragmentacion.
- 3 Pensar en los tiempos de ejecucion de best-fit/worst-fit versus first-fit.(A menos que los bloques esten 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?