



Proyecto Final
“Simulador de Procesos”

Materia: Sistemas Operativos
Profesor: Francisco Gutierrez

Lic. en Informática y Tecnologías Computacionales
6° A

Integrantes:
Elías de Jesús Alba Avalos
Sebastián De Lara Del Hoyo
José Manuel Carrillo Torres
Alma Guadalupe González Gutiérrez

INDICE

Contenido

INTRODUCCION:	3
CONTENIDO:	3
Mejor Ajuste:	3
Peor Ajuste:	4
Primer Ajuste:	4
Manual de usuario	5
CONCLUSIONES:	7
BIBLIOGRAFIA:	8

INTRODUCCION:

El siguiente trabajo pretende informar sobre la realización del proyecto final para la materia de sistemas operativos, el cual consta de un simulador de procesos desarrollado en lenguaje C# en base a 3 tipos de asignación vistos en clase; Mejor Ajuste, Peor Ajuste y Primer Ajuste.

Se incluye una breve explicación de los tipos de asignación, imágenes del sistema realizado y las conclusiones del equipo e individuales.

CONTENIDO:

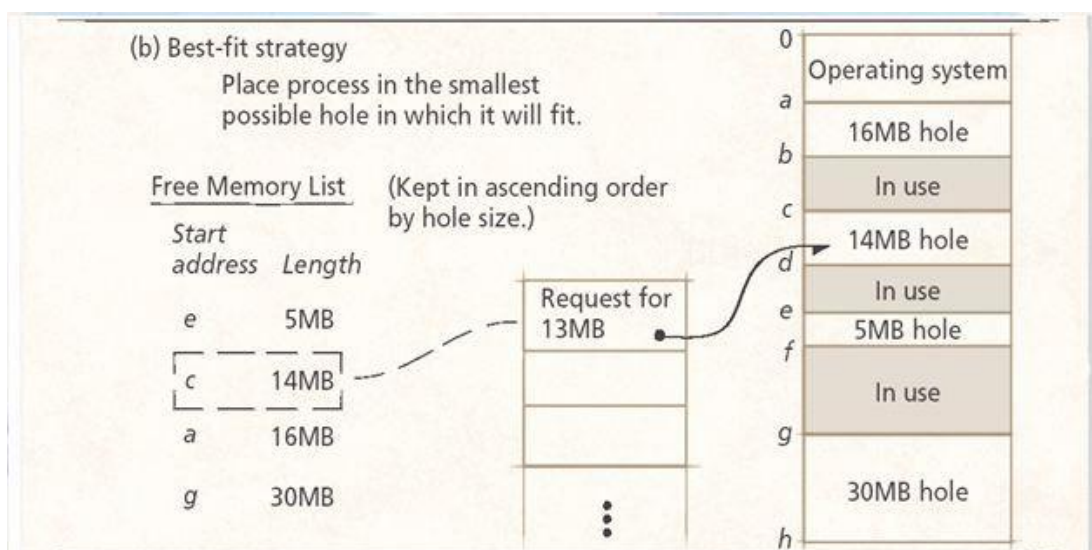
En el Particionamiento Dinámico:

- El tamaño y el número de particiones es variable
- Al proceso se le asigna exactamente la cantidad de memoria que necesita
- Aparecen huecos en la memoria. Esto se conoce como fragmentación externa
- Se debe realizar una compactación para desplazar a los procesos de forma que estén juntos y todo el espacio libre esté en un solo bloque

En base a lo anterior el sistema operativo debe decidir qué bloque libre asignar a un proceso. Los cuales se muestran a continuación.

Mejor Ajuste:

Elige el bloque que tiene el tamaño más cercano al solicitado, tiene el peor rendimiento de todos. Como se busca el bloque más pequeño por proceso, se



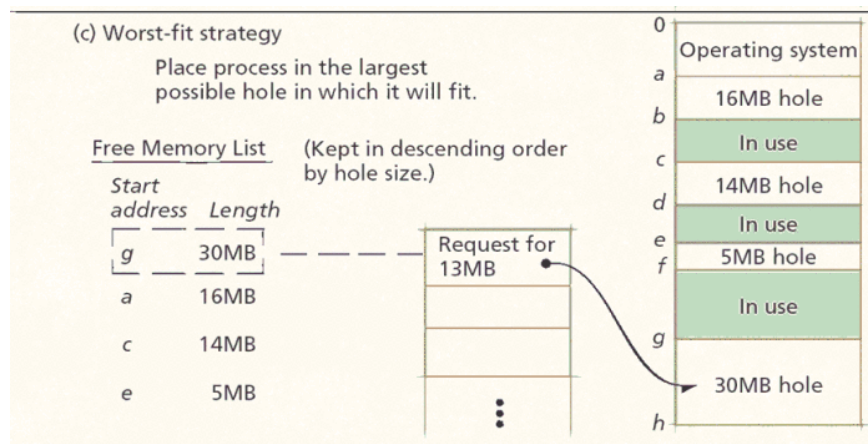
produce el menor volumen de fragmentación, pero hay que compactar más a menudo.

Peor Ajuste:

Asigna a un proceso el hueco más grande, de la lista de huecos, siempre que sea lo suficientemente grande para poder ubicarlo en él.

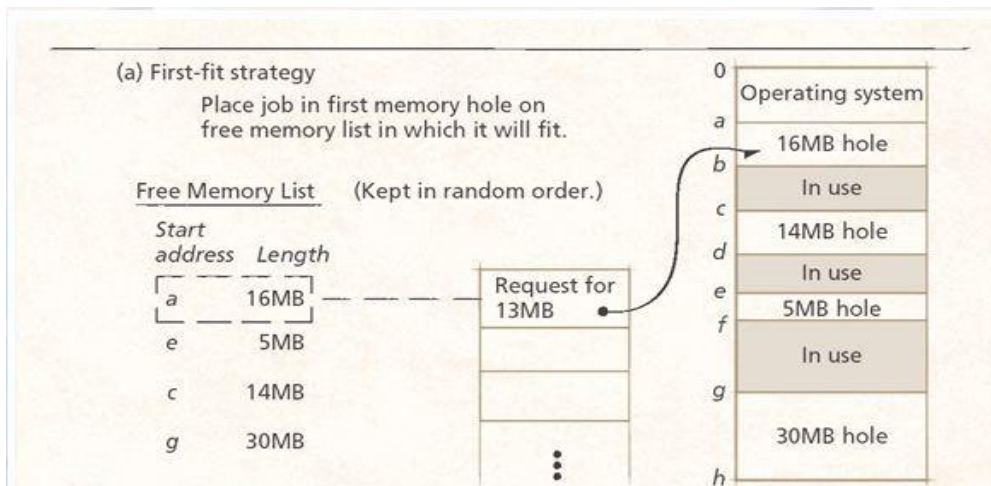
La búsqueda de hueco, puede comenzar desde el principio de la lista de huecos o desde donde finalizó la búsqueda anterior. Para encontrar el hueco adecuado es necesario explorar la lista completa, a menos que ésta esté ordenada según tamaño de hueco.

Hueco más grande: Pretende conseguir que los huecos que queden sean grandes (requiere ver toda la lista si no ordenada).



Primer Ajuste:

Escoge el primer hueco libre de tamaño suficiente. Búsqueda al principio o a partir de este punto.



Manual de usuario

1.- Cuando se ejecuta el simulador aparece esta pantalla donde el usuario mueve los parámetros según los requerimientos.

Form1

Tiempo total de simulación: Seg

Tiempo por proceso: Seg

Tamaño de la memoria: Kb

Tamaño máximo de un proceso: Kb

2.- En la siguiente imagen se muestran los parámetros establecidos y se selecciona el tipo de asignación deseado. En este ejemplo se selecciono el mejor Ajuste

Si se da click en Nueva aparece la pantalla de la imagen 1 donde se pide volver a introducir los parámetros para la simulación

Form2

Nueva

Memoria

Procesos

35 Kb

41 Kb

142 Kb

18 Kb

37 Kb

77 Kb

106 Kb

44 Kb

34 Kb

Seleccione tipo de asignación:

☒ Mejor Ajuste

☐ Peor Ajuste

☐ Primer Ajuste

groupBox2

Segmento actual:

Segmentos de memoria: 8

Numero de posiciones: 8

<-Estado de los procesos: -

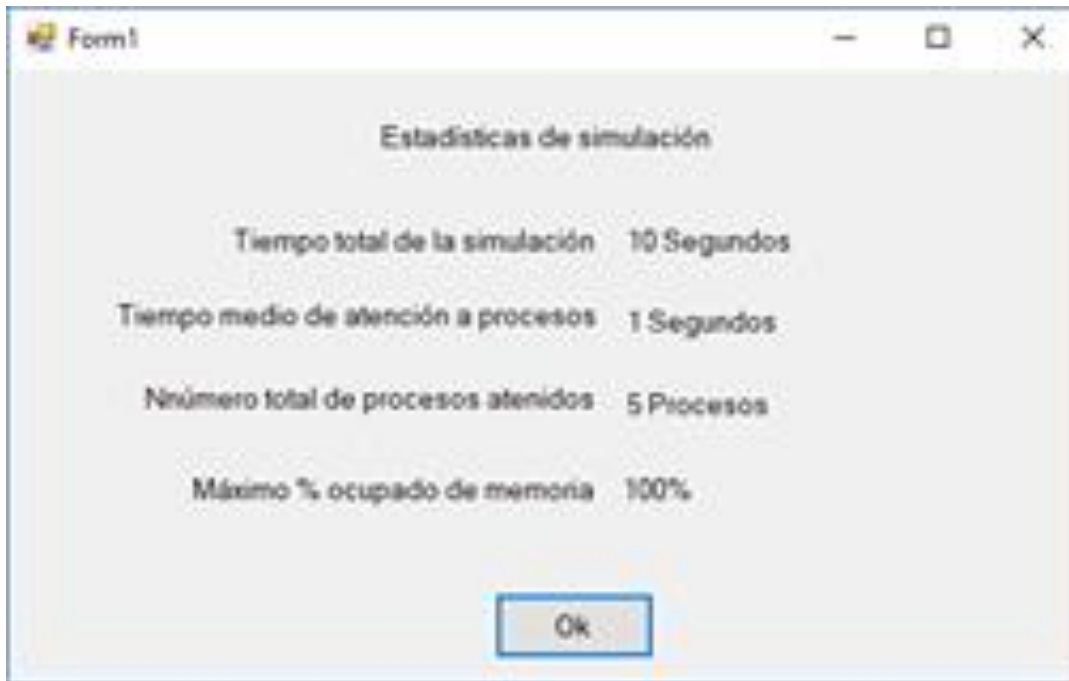
Proceso en espera

Total de memoria: 500 kb

Iniciar simulación

Detener

3.- Al terminar el tiempo de simulación aparece una pantalla que muestra las estadísticas solicitadas.



CONCLUSIONES:

En general, aprendimos como son las asignaciones de memoria, ya que cada uno investigo un algoritmo y lo programó. Al principio nos pareció un poco difícil pues no entendíamos como funcionaban las técnicas, hasta que las investigamos e hicimos ejercicios a mano para comprenderlos en su totalidad

No siempre la asignación más rápida es la mas efectiva .

Elías:

Al cargar procesos en memoria el primer espacio disponible no siempre dispondrá de un tamaño de memoria adecuado para cada proceso y al buscar un lugar adecuado puede que se asignen en espacios que desperdician más o menos memoria, es por este motivo que se requieren métodos de asignación. El método más óptimo de asignación es "el mejor método", ya que éste desperdicia menos memoria al asignar los procesos en los espacios de memoria.

José Manuel:

Los algoritmos de asignación de memoria son variados y útiles para poder utilizarlos en la situación correcta. El modo en que operan lo aprendí cuando programamos el simulador. El mejor para una computadora supondría que es el de mejor asignación el cual busca el sector de memoria donde hay menos desperdicio.

Alma:

La memoria tiene dos tipos de segmentación , todos los segmentos del mismo tamaño o los segmentos de diferentes tamaños. El problema se crea cuando en los segmentos de diferentes tamaños quedan huecos libres que deben de ser usados por los procesos para evitar el desperdicio de memoria.

En el proyecto usamos tres técnicas:

- Mejor ajuste que es cuando un proceso busca en todos los “huecos” para encontrar en donde mejor quepa y no desperdicie memoria
- Peor Ajuste: Los procesos se acomodan en el “hueco” mas grande que se tiene independiente si se desperdicia o no espacio de memoria
- Primer Ajuste: El proceso se queda en el primer hueco donde cabe

Considero que es mas rápido la técnica de primer ajuste pero no es la mejor pues desperdicia memoria,

Sebastián:

Para crear un proceso, este debe cargarse en memoria. Para que esto ocurra, el Sistema Operativo selecciona un proceso de la cola de entrada para cargar en memoria. Por lo anterior, y tras usar los tres métodos de asignación, se concluye que el mejor método para realizar la creación de procesos es, como su nombre lo indica, el de Mejor Ajuste, puesto a que con él se busca obtener el menor desperdicio de memoria principal, mientras que con el método de Primer Ajuste y Peor Ajuste son los que pueden ocupar segmentos muy grandes de memoria y desperdiciar casi todo el espacio en ellos.

BIBLIOGRAFIA:

http://www3.uji.es/~redondo/so/capitulo4_IS11.pdf

<https://www.ac.uma.es/~sromero/so/Capitulo7.pdf>

http://labvirtual.webs.upv.es/Worst_Fit.htm

<https://chsos20152912029.wordpress.com/tag/primer-ajuste/>