

LABORATORIO DE SISTEMAS OPERATIVOS

Nombre: Fernando Huilca Villagómez

Carrera: Ingeniería en Software

Grupo: GR1SW

Fecha: 12/08/2024

Profesor: Marco Sánchez PhD.

TEMA: GESTIÓN DE MEMORIA Y PAGING

Índice de Contenidos

Objetivos.....	1
Informe.....	2
Conclusiones y Recomendaciones.....	9
Bibliografía.....	10

Índice de Figuras

Figura 1: Asignación de Memoria

Figura 2: Memoria de Carga en procesos

Figura 3: Best Fit

Figura 4: Worst Fit

Figura 5: Valores de memoria

Figura 6: Procesos de 4,5 y 3 páginas

Figura 7: Tabla de valores de memoria

Figura 8: Ejecución de memoria para cada proceso

1. OBJETIVOS

- 1.1. Demostrar el funcionamiento de la administración de memoria utilizando el simulador CPU Simulator, enfocándose en la actividad que realiza una memoria virtual y la paginación en la gestión de memoria.
- 1.2. Implementar dos estrategias de asignación utilizadas en la gestión de la memoria para comparar su eficiencia.

INFORME

Procedimiento: Se utiliza CPU-OS Simulator para implementación e identificación de asignación de memoria, preferible el uso de un sistema operativo Unix/Linux para el mejor manejo e identificación.

1. Políticas de asignación en memoria

1. Ejecutar CPU OS Simulator.
2. Abrir la ventana de OS Simulator
3. Dentro de Views seleccionar la opción View Memory.
4. En la parte inferior de la pantalla de Main Memory, parte Manual llenar la memoria con 10 frames. Asegúrese de que la lista desplegable de Frames esté establecido en 1. Haga clic en el botón ADD repetidamente hasta que se asigne toda la memoria (necesita 10 clics)
5. Remover los frames 2, 3, 5, 7, 8, 9.
6. Observar la asignación en memoria, capturar una imagen y explicar por qué se obtiene esa imagen.



Figura 1: Asignación de Memoria

La imagen muestra un simulador de gestión de memoria RAM, donde se visualizan marcos ocupados y libres. La memoria asignada es de 1024 bytes y la libre es de 1536 bytes, sin espacio de intercambio usado. Se utilizan las políticas de "First Fit" para la colocación y "FIFO" para el reemplazo. La paginación y la memoria virtual están habilitadas, pero no la compactación ni la demanda paginada. El simulador permite agregar o remover marcos y compactar la memoria si es necesario.

7. En la ventana View Memory -> Main Memory verificar que se encuentre seleccionado First Fit en Placement Policy.
8. Cargar un proceso a partir del siguiente código: (compilar, cargar en memoria y crear un proceso)

```
program LoopForeverTest
  N = 0
  while true
    N = N + 1
  wend
end
```

9. Observar en qué parte de memoria se carga el proceso, capturar una imagen y explicar por qué se obtiene esa imagen.

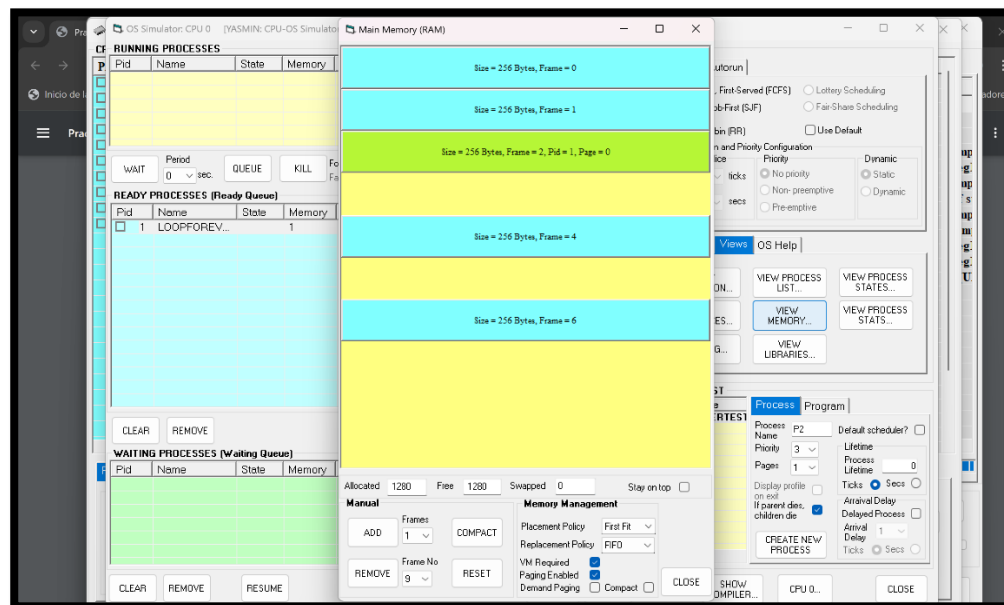


Figura 2: Memoria de Carga en procesos

La imagen muestra un simulador de sistemas operativos con una ventana que visualiza la memoria RAM. Aquí, se observa que un proceso (Pid = 1) ocupa un marco de 256 bytes (Frame = 2), mientras que otros marcos están libres. La memoria asignada es de 1280 bytes, con 1280 bytes libres, y se están utilizando políticas de colocación "First Fit" y reemplazo "FIFO", con la paginación y memoria virtual activadas.

10. Remover el proceso del estado Listo (ready processes).

11. Repetir el paso 7, 8, 9 y 10 con Best Fit y luego Worst Fit en Placement Policy.

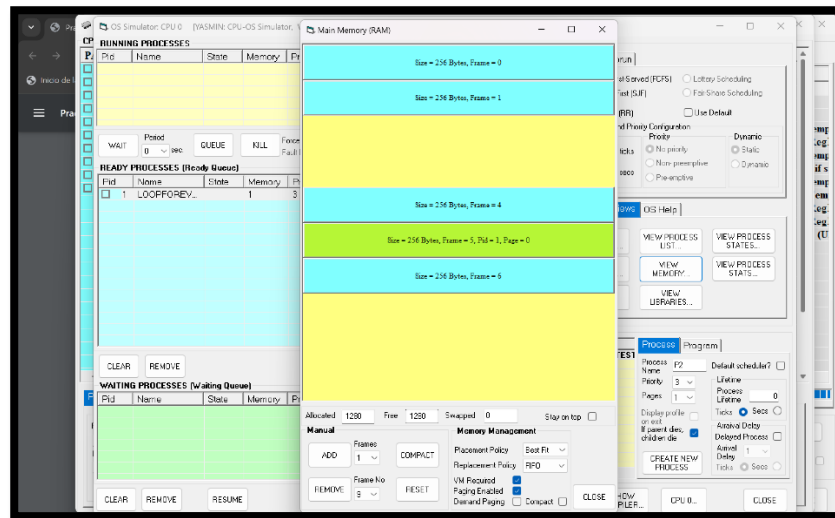


Figura 3: Best Fit

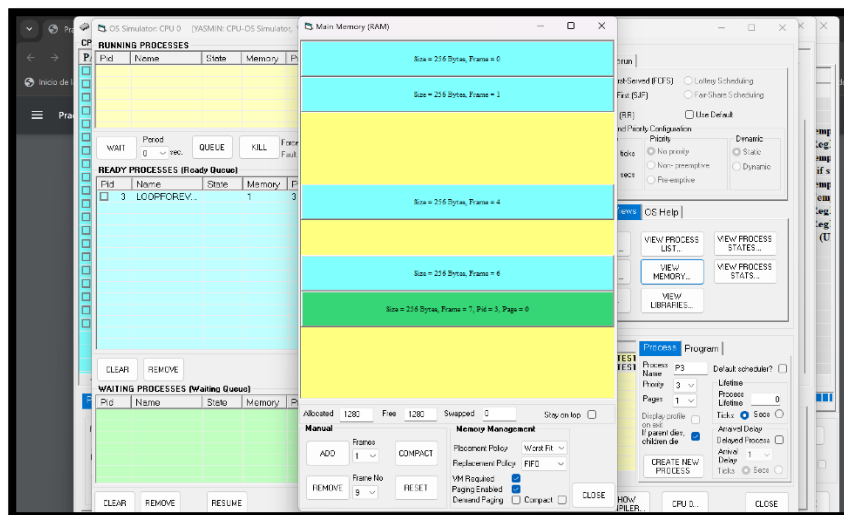


Figura 4: Worst Fit

2. Memoria virtual y el swapping

1. Compilar y cargar en memoria el siguiente código:

```
program OSQueuesDemo %Start of program
  while true %Forever loop
    for n = 1 to 15 %Repeat 15 times
      i = 1 %Just something to do!
    next %End of repeat loop
  wait(3) %Suspend program for 3 secs
wend %End of forever loop
end %End of program
```

2. Ingresar a la opción de OS Simulator, en la pestaña Views seleccionar View Utilization y View Memory en donde se deshabilitará la paginación.
3. Capturar las imágenes de estas dos ventanas y documentar los valores de memoria Allocated, Free y Swapped.

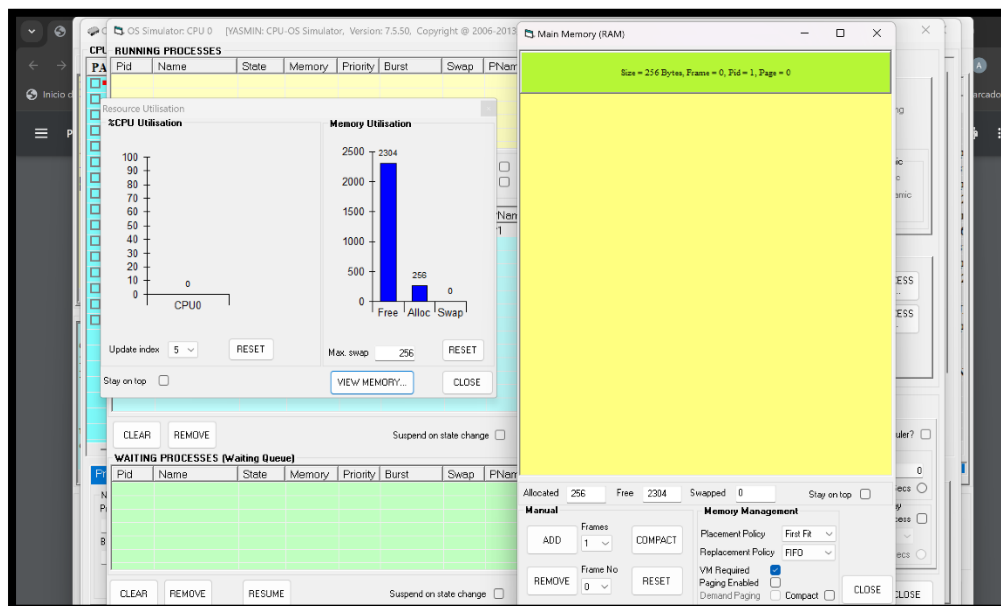


Figura 5: Valores de memoria

4. Crear procesos de tal manera que sobrepasen la memoria disponible, en este caso tres procesos de 4, 5 y 3 páginas respectivamente.

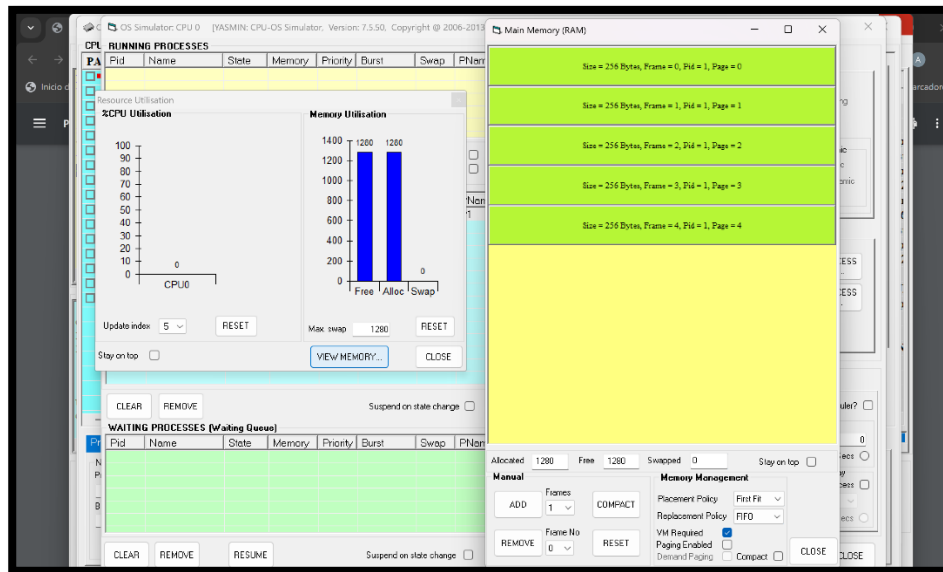
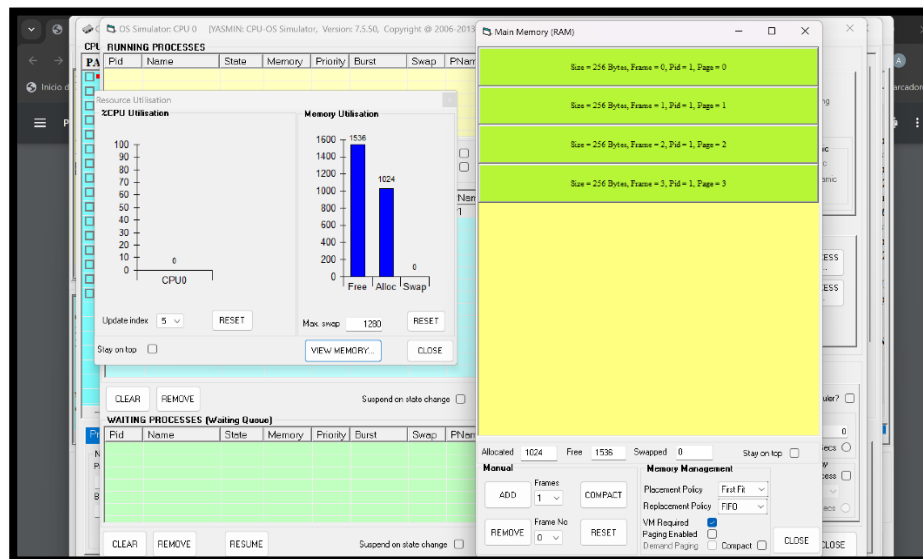
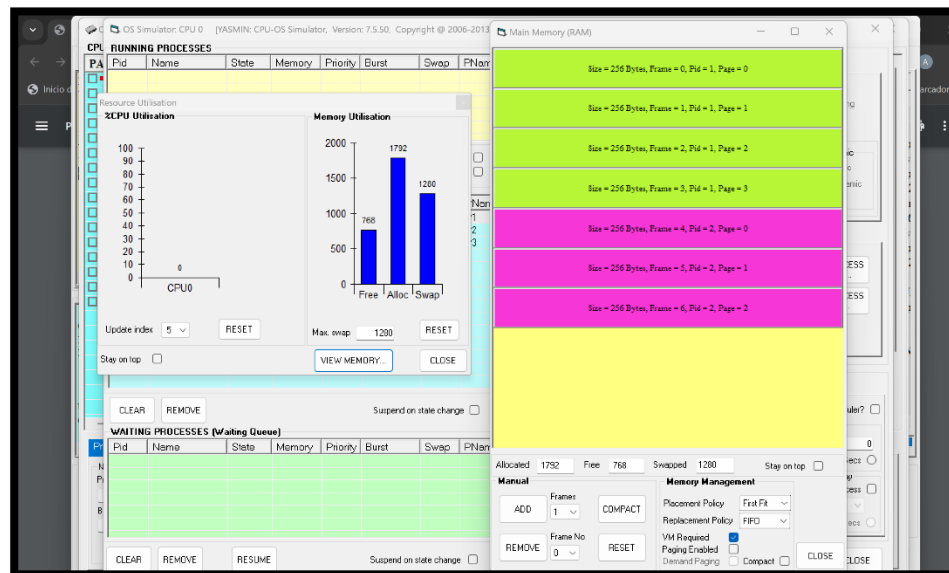
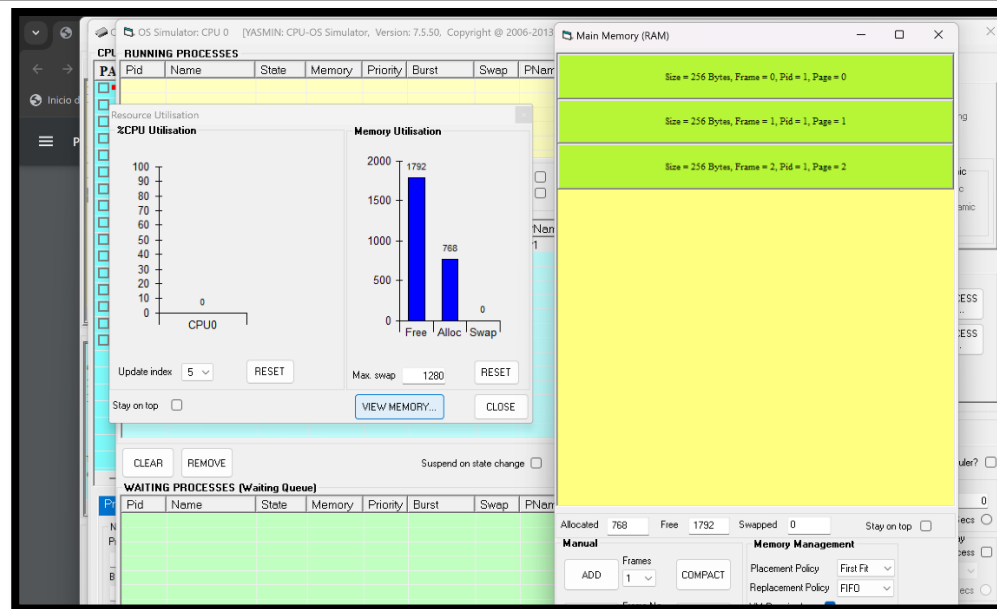


Figura 6: Procesos de 4,5 y 3 páginas





5. Capturar nuevamente las imágenes de estas dos ventanas y documentar los valores de memoria Allocated, Free y Swapped.

Process	Pages	Free	Alloc	Swap
P1	4	1536	1024	0
P2	5	1200	1200	0
P3	3	1792	768	0

Figura 7: Tabla de valores de memoria

6. Explicar los valores obtenidos.

Los valores reflejan la asignación de memoria para tres procesos en un sistema operativo: P1 tiene 4 páginas y se le asignaron 1024 bytes, quedando 1536 bytes libres; P2 tiene 5 páginas con 1200 bytes asignados, dejando 1200 bytes libres; y P3 tiene 3 páginas con 768 bytes asignados, quedando 1792 bytes libres. No se ha utilizado memoria de intercambio (swap) en ninguno de los procesos.

7. Seleccionar en la ventana de procesos en ejecución la opción(running processes) Suspend on state change, asegurese de que la velocidad de la CPU está en la posición más rápida(en la pestaña OS Control) y ejecutar los procesos creados (click en el botón START en la pestaña OS Control). Cuando aparece la ventana del mensaje indicando un cambio en el estado, tome nota de los tres valores de utilización de la memoria en la siguiente tabla que se muestra a continuación considerando el proceso actualmente en ejecución.

Reinicie el SO haciendo clic en el botón RESUME en la pestaña OS Control

Cuando termine, desmarque la casilla de verificación e Suspend on state change y haga clic en el botón RESUME. A medida que los procesos comienzan a ejecutarse, haga clic en el botón KILL para detenerlos uno por uno.

8. Documentar qué sucede en memoria al ejecutarse cada proceso.

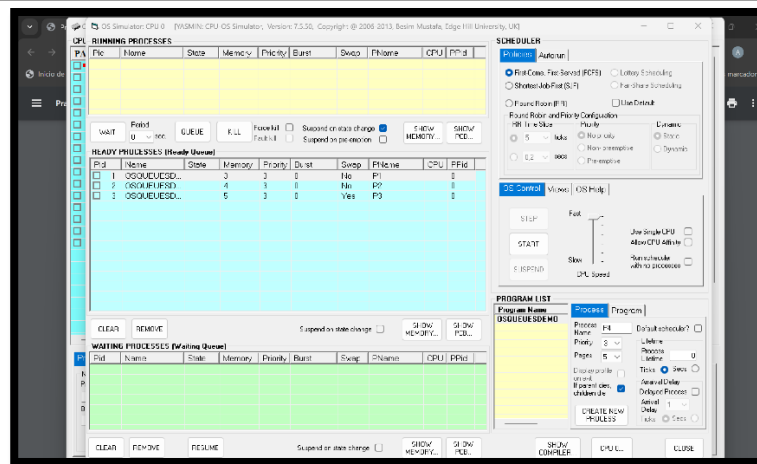
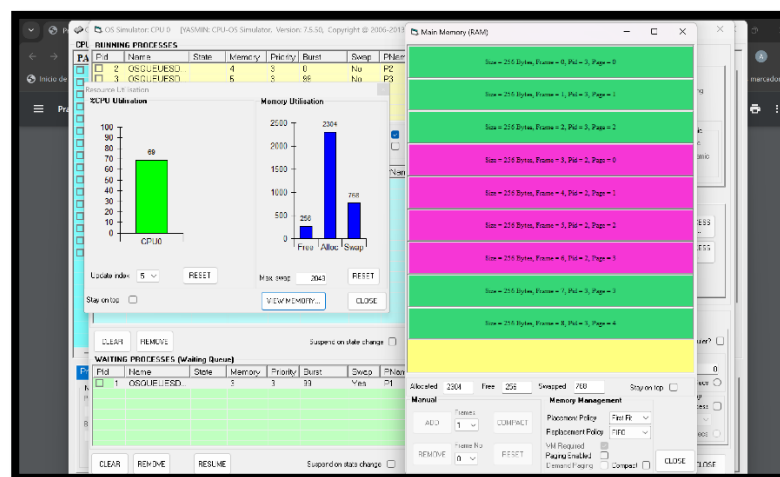
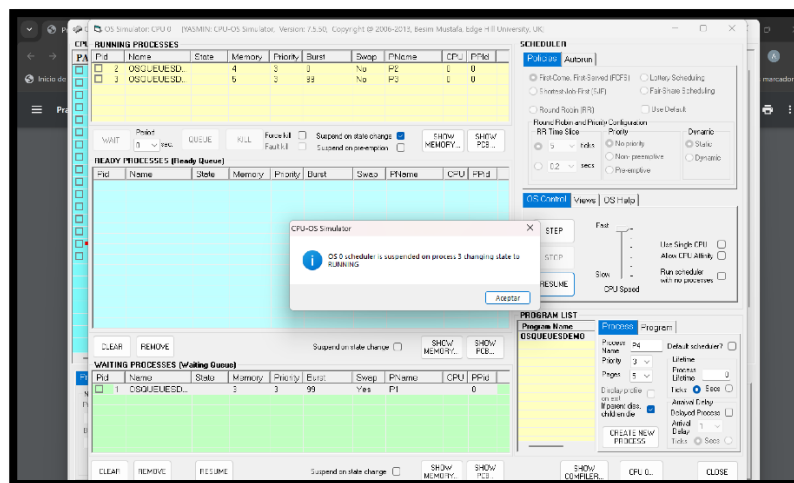


Figura 8: Ejecución de memoria para cada proceso



9. Habilitar paging y repetir los pasos 3 al 8.

10. Habilitar también Demand paging y repetir los pasos 3 al 8.

2. CONCLUSIONES Y RECOMENDACIONES

Conclusiones:

Eficiencia en la gestión de memoria: La gestión de memoria realizada a través del simulador demuestra que el sistema puede manejar múltiples procesos sin necesidad de recurrir al swapping. Las políticas de asignación como "First Fit", "Best Fit" y "Worst Fit" se evaluaron, y se pudo observar cómo estas afectan la utilización de la memoria.

Importancia de la paginación: La paginación y la memoria virtual se activaron en varios experimentos, mostrando que estas técnicas son efectivas para optimizar el uso de la memoria física. A pesar de que no se necesitó el uso de swapping, la habilitación de la paginación aseguró una distribución eficiente de los marcos de memoria entre los procesos.

Recomendaciones:

Monitoreo continuo de la memoria: Es fundamental monitorear constantemente la memoria libre para prever la necesidad de activar el swapping o ajustar las políticas de asignación. Esto evitará posibles cuellos de botella en situaciones de alta carga de procesos.

Optimización de la asignación de páginas: Para maximizar la eficiencia del sistema, es recomendable ajustar las políticas de asignación de páginas. Esto puede incluir cambiar dinámicamente entre "First Fit", "Best Fit" y "Worst Fit" según la carga de trabajo, o considerar algoritmos más avanzados como "Next Fit" o "Buddy System".

3. BIBLIOGRAFÍA

[1] Silberschatz, Galvin, Gagne *Operating System Concepts*, 8ª ed. United States of America: Carolyn Weisman, 2009.