

Taller 3 - Sistemas Operativos

Sebastián Quesada Rojas, *estudiante*, Ingeniería en Computadores, *Instituto Tecnológico de Costa Rica*



1 PREGUNTAS GUÍA

1.1 ¿Cuál sería la mayor implicación de no tener memoria virtual?

La principal consecuencia de carecer de memoria virtual sería la imposibilidad de ejecutar aplicaciones que demandan más memoria de la que el sistema tiene físicamente disponible. Sin la memoria virtual, los programas deberían adaptarse al tamaño restringido de la RAM, lo que podría provocar su cierre o no inicio. Esto también restringiría la multitarea, dado que solo se podrían realizar un número limitado de procesos simultáneamente, lo cual impactaría en la eficiencia y la experiencia del usuario.

1.2 ¿Por qué la memoria necesita reloj?

La memoria requiere un cronómetro para coordinar las actividades de lectura y escritura. El reloj ofrece un ritmo invariable que garantiza que los datos y las señales se gestionen en el instante correcto, lo cual es crucial para preservar la consistencia e integridad de los datos en los circuitos. Sin un reloj, las operaciones podrían llevarse a cabo de forma desorganizada, provocando fallos y pérdida de información.

1.3 ¿Espacio de direccionamiento es lo mismo que memoria? Justifique

No, el espacio de orientación no coincide con la memoria. El espacio de direccionamiento hace referencia al espectro de direcciones que puede emplear un procesador para entrar a la memoria. En otras palabras, es el número total de ubicaciones que un sistema puede señalar, sin importar la cantidad de memoria física que tenga instalada. Por ejemplo, un sistema podría poseer un espacio de direccionamiento de 32 bits, lo que implica que tiene la capacidad de gestionar 4 GB de direcciones, a pesar de que la memoria física instalada sea inferior.

1.4 Explique cuatro tecnologías de fabricación de la memoria RAM

- DRAM: Emplea celdas de memoria que guardan datos en forma de carga eléctrica en un condensador. Dado que la carga se filtra con el paso del tiempo, requiere una revisión periódica. Es habitual en

ordenadores personales.

- SRAM: Emplea celdas de memoria más avanzadas que no necesitan refresco, dado que los datos se conservan aunque exista energía. Se compara con la DRAM en rapidez y precio y se emplea en cachés de procesadores.
- DDR SDRAM: Optimiza el desempeño de la DRAM mediante el traspaso de datos dos veces por ciclo de reloj, lo que incrementa el ancho de banda. Hay diversas generaciones (DDR, DDR2, DDR3, DDR4, DDR5), todas ellas con avances en rapidez y eficacia.
- MRAM: Emplea características magnéticas para guardar información, proporcionando no solo una velocidad de acceso elevada, sino también una conservación de datos sin necesidad de energía, lo que la convierte en una memoria más eficaz en términos energéticos y de larga duración. Es una tecnología en ascenso con la capacidad de sustituir a las tecnologías de RAM convencionales.

1.5 Cuales son las características de una memoria ideal para un ingeniero en computadores. Explique cada una de ellas

La memoria debe poder acceder y mover datos de manera rápida para cumplir con las exigencias de procesamiento de datos en tiempo real. Esto disminuye el periodo de espera del procesador y optimiza el desempeño global. La memoria perfecta debe tener la habilidad de guardar una gran cantidad de información en un espacio físico limitado. Es crucial para maximizar la utilización del espacio en aparatos pequeños como smartphones y tabletas. En aparatos móviles y sistemas integrados, el uso de energía es esencial. Una memoria que gasta menos energía contribuye a extender la durabilidad de la batería y disminuye los gastos de operación. La memoria debe tener la capacidad de soportar situaciones difíciles, tales como variaciones de temperatura y vibraciones, y también debe asegurar la integridad de los datos a través del tiempo, reduciendo al mínimo la pérdida de información debido a errores.

1.6 ¿Qué es una partición swap? ¿Para que se utiliza? ¿Es preferible tener un swap de gran tamaño?

Una partición swap es una zona en el disco duro que se emplea como adicional de memoria virtual cuando la RAM física se encuentra saturada. Cuando el sistema requiere más memoria de la que dispone, transfiere de manera temporal datos menos empleados desde la RAM a la partición de intercambio, liberando espacio en la memoria física para otros procedimientos.

Poseer un swap de gran envergadura puede ser ventajoso, dado que posibilita que el sistema gestione múltiples aplicaciones al mismo tiempo y previene fallos o cierres imprevistos debido a la escasez de memoria. No obstante, un swap amplio no debe sustituir la demanda de suficiente memoria RAM física, dado que el acceso a la información en el disco es considerablemente más pausado que en la RAM.

1.7 ¿A que se refiere el término de swappiness? ¿Que significa si el valor es cercano en 100?

El concepto de "swappiness" alude a un indicador en sistemas Linux que regula la inclinación del sistema a trasladar procesos de RAM a la partición de intercambio. El coeficiente de swappiness puede oscilar entre 0 y 100.

Un valor bajo (alrededor de 0) señala que, siempre que sea viable, el sistema evitará el uso del swap y optará por conservar los procesos en la RAM, lo que resulta ventajoso para aplicaciones que necesitan un acceso inmediato a la información. Un valor próximo a 100 indica que el sistema tiene una mayor tendencia a trasladar procesos a la swap, lo que puede resultar beneficioso en circunstancias donde la memoria RAM es escasa, pero puede provocar un desempeño más pausado, dado que el acceso a los datos en el disco es más complicado, dado que el acceso a los datos en el disco es más difícil.

2 MEMORIA PRINCIPAL EN LINUX

Siguiendo la lista de instrucciones dadas se tienen las pruebas del proceso:

```

sebas@instance-20241015-213826 ~$ free -h
total        used        free      shared  buff/cache   available
Mem:          3.6Gi       430Mi       3.1Gi       8.0Mi       277Mi       3.1Gi
Swap:          0Mi         0Mi         0Mi         0Mi       277Mi       3.1Gi
sebas@instance-20241015-213826 ~$

```

Fig. 1. Ejecución Primer Comando

Para esta parte de la ejecución, se había obtenido este *screenshot* en caso de presentarse el error de permiso

```

sebas@instance-20241015-213826 ~$ partprobe
sebas@instance-20241015-213826 ~$ sudo fdisk -l
Disk /dev/sda: 50 GiB, 53687091200 bytes, 104857600 sectors
Disk model: PersistentDisk
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 4096 bytes
Disklabel type: gpt
Disk identifier: 0CBF3783-8157-4352-9536-349180A48B98

Device      Start      End  Sectors  Size Type
/dev/sda1    2048    411647    409600    200M EFI System
/dev/sda2    411648  104855647 104444000    49.8G Linux filesystem
/dev/sda3    104855648 104857566    1919    959.5K Linux swap
sebas@instance-20241015-213826 ~$ sudo mkswap /dev/sda3
Setting up swappspace version 1, size = 952 KiB (974848 bytes)
no label, UUID=6ad14ac-f861-40b7-9dd1-35d0b0c9b10
sebas@instance-20241015-213826 ~$ swapon /dev/sda3
swapon: cannot open /dev/sda3: Permission denied
sebas@instance-20241015-213826 ~$ sudo swapon /dev/sda3

```

Fig. 2. Implementación de la Partición

denegado, sin embargo, luego se pudo solucionar el error con el comando *sudo* pero se olvidó realizar el nuevo *screenshot*.

```

sebas@instance-20241015-213826 ~$ echo UUID=9de0f321-ccbe-48cd-a4d3-242b60383edb swap swap de
fault 0 /etc/fstab
sebas@instance-20241015-213826 ~$

```

Fig. 3. Comando 10

Con el proceso mostrado a continuación, se puede observar que se logró crear con éxito la partición en CentOS. La única diferencia fué que se tuvo que utilizar como partición 3 y no como partición 2, como se describía en la especificación. La implementación se realizó en una máquina virtual hecha en Google Cloud.

3 PAGINACIÓN DE MEMORIA EN LINUX

Para crear el archivo swap se ejecutaron, en otra MV nueva, los siguientes comandos:

```

sudo fallocate -l 1G /swapfile
sudo chmod 600 /swapfile
sudo mkswap /swapfile
sudo swapon /swapfile
free -m

```

El último comando se ejecuta para saber que se generó el archivo correctamente. Posteriormente, se generó una carpeta para la ejecución del código. Con la ayuda de ChatGPT, se hizo un código genérico para generar una imagen cualquiera con Python, esto para no tener que descargar el SDK de Google para cargar una imagen a la MV. El código generado es:

```

from PIL import Image, ImageDraw

# Crear una imagen en blanco (color de fondo blanco)
ancho, alto = 400, 400
imagen = Image.new('RGB', (ancho, alto),
color='white')

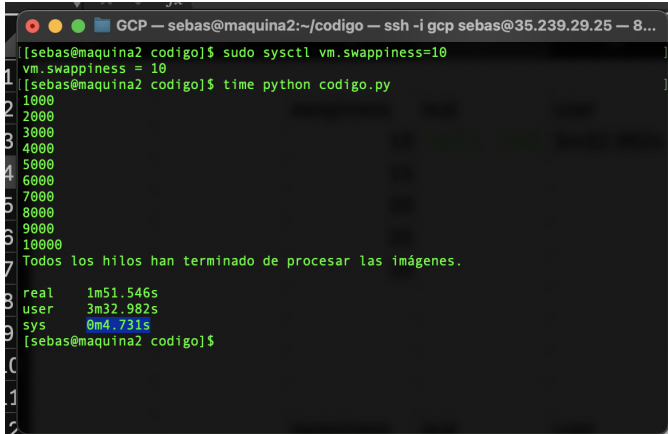
# Dibujar un rectángulo
dibujar = ImageDraw.Draw(imagen)
dibujar.rectangle([50, 50, 350, 350],
outline="black", fill="blue")

```

```
# Guardar la imagen
imagen.save('imagen_creada.png')
```

Luego se procedió a cargar el código realizado con objetos a la máquina virtual. El código se realizó utilizando Python y se puede observar [aquí](#). Se procedió a ejecutar el código y cambiar el parámetro de swappiness según se pedía en la especificación con el comando:

```
sudo sysctl vm.swappiness=10
```



```
GCP — sebas@maquina2:~/codigo — ssh -i gcp sebas@35.239.29.25 — 8...
[sebas@maquina2 codigo]$ sudo sysctl vm.swappiness=10
vm.swappiness = 10
[sebas@maquina2 codigo]$ time python codigo.py
1000
2000
3000
4000
5000
6000
7000
8000
9000
10000
Todos los hilos han terminado de procesar las imágenes.
real    1m51.546s
user    3m32.982s
sys     0m4.731s
[sebas@maquina2 codigo]$
```

Fig. 4. Ejecución del código

Por último, se procedió a guardar los resultados en 2 txt disponibles con los nombres **p1.txt** y **p2.txt**. Con el código disponible [aquí](#), se procedió a generar las gráficas.

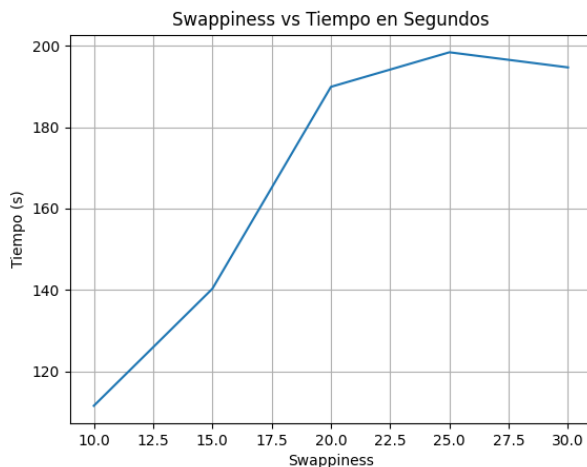


Fig. 5. Primera gráfica obtenida

Conforme se incrementa el valor de swappiness de 10 a 25, se nota un aumento considerable en el tiempo de ejecución. Esto indica que el sistema está empleando más intercambio, lo que disminuye el desempeño debido al incremento de las operaciones de intercambio entre la memoria física y el espacio de intercambio. Desde un valor de 25, el tiempo se estabiliza, lo que señala que el sistema ha llegado a un punto donde el uso de swap no incrementa ni disminuye significativamente el rendimiento, solamente disminuye muy poco.

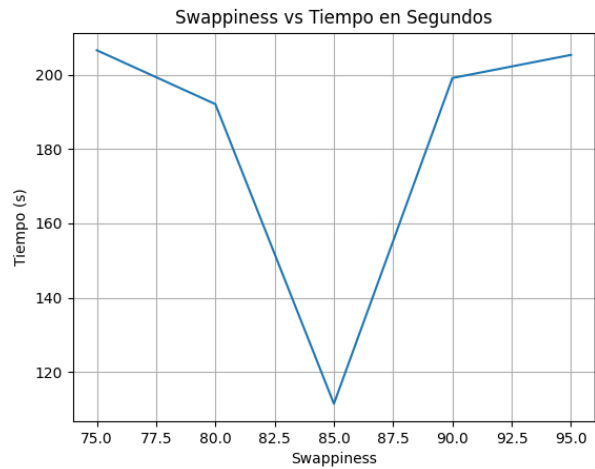


Fig. 6. Segunda gráfica obtenida

En este caso, los valores de swappiness más elevados inicialmente se mantienen por un largo periodo (parecido al valor más alto en el primer gráfico), pero se nota un tiempo mínimo en el valor de 85. Este comportamiento podría ser resultado de que en esta etapa, el sistema mantiene un mejor equilibrio entre el uso de memoria física y swap, consiguiendo un desempeño más eficiente. Cuando se ajusta el swappiness a niveles extremadamente elevados, como 90 y 95, el tiempo vuelve a incrementarse, lo que señala que el sistema se apoya demasiado en el espacio de intercambio, impactando de nuevo en el rendimiento.