Laboratorio 2: Gestión de Memoria

En este laboratorio veremos como actúa el sistema operativo cuando está utilizando memoria física y memoria virtual, como también el rendimiento con caché y sin caché. Primero empezaremos viendo la memoria física vs la memoria virtual.

Para saber cuando se empieza a usar la memoria virtual utilicé dos bash, uno que te permite llenar la RAM y otro que te permite monitorear el uso de la memoria

```
lucas@lucas-VirtualBox:~$ watch -n 1 free -h
```

Para monitorear el uso de memoria

```
lucas@lucas-VirtualBox:~$ stress --vm 2 --vm-bytes 1G --timeout 100s
```

Para utilizar 1GB de RAM

Con estos dos comandos ya podremos ver cuando el sistema operativo empieza a utilizar la memoria virtual, primero veremos como se ve cuando no se está utilizando la memoria virtual en el ordenador.

```
Every 1.0s: free -h                                                    lucas-V

               total        used        free      shared  buff/cache   available
Mem:           2.9Gi       1.0Gi       1.8Gi        39Mi       375Mi       1.9Gi
Swap:          2.9Gi          0B       2.9Gi
```

Podemos apreciar que hay una fila llamada “Mem” y otra llamada “Swap”, la primera es la RAM y la segunda es la memoria virtual que en ese momento no se esta utilizando y por eso aparece 0B, mientras que en la RAM se está utilizando 1GB. Ahora utilizaremos el bash stress para forzar a que el sistema operativo utilice la memoria virtual disponible.

```
Every 1.0s: free -h                                                    lucas-Vi

               total        used        free      shared  buff/cache   available
Mem:           2.9Gi       1.8Gi       1.1Gi        27Mi       243Mi       1.1Gi
Swap:          2.9Gi       244Mi       2.7Gi
```

Como podemos ver el sistema operativo empezó a utilizar la memoria virtual, ya que paso de 0B a 244Mi en ese momento.

Ahora mediremos el rendimiento del sistema operativo sin la utilización de la memoria virtual y mientras se está utilizando la memoria virtual. Para esto utilice el programa hecho en el laboratorio anterior. Pero antes de hacer eso voy a deshabilitar la memoria virtual y volver a habilitarla para restablecerlo a 0B, utilizando “sudo swapoff -a” para deshabilitarla y “sudo swapon -a” para volver a habilitarla, ya que incluso luego de terminar el bash stress el sistema operativo sigue utilizando una pequeña cantidad de memoria virtual.

Cuando ejecuto el programa sin que el sistema operativo esté utilizando memoria virtual tarda 18,48 segundos en realizar el proceso.

```
lucas@lucas-VirtualBox:~$ python3 miprograma.py
Ingrese su nombre: lucas
Hola  lucas  este es un simulador de estados hecho por Lucas Velazquez favor agu
arde

Iniciando...
Ejecutando
Terminado
El tiempo del estado ´Ejecutando´ a ´Terminado´ fue de:  18.48 segundos
```

Mientras que cuando se utiliza la memoria virtual tarda 23,58 segundos en ejecutarse.

```
lucas@lucas-VirtualBox:~$ python3 miprograma.py
Ingrese su nombre: lucas
Hola  lucas  este es un simulador de estados hecho por Lucas Velazquez favor agu
arde

Iniciando...
Ejecutando
Terminado
El tiempo del estado ´Ejecutando´ a ´Terminado´ fue de:  23.58 segundos
```

Ahora compararemos el rendimiento del sistema operativo cuando la lectura de un archivo grande se realiza sin caché y con caché, primeramente hay que vaciar todo el caché con el siguiente bash.

```
lucas@lucas-VirtualBox:~$ sync
lucas@lucas-VirtualBox:~$ echo 3 | sudo tee /proc/sys/vm/drop_caches
```

Luego realizamos la lectura del archivo grande que ya tengo creado, por medio de este bash que también nos muestra el tiempo de lectura.

```
lucas@lucas-VirtualBox:~$ /usr/bin/time -f "%e" dd if=archivo_cache of=/dev/null
 bs=1M
500+0 records in
500+0 records out
524288000 bytes (524 MB, 500 MiB) copied, 0.751336 s, 698 MB/s
0.76
```

Sin caché, tardó 0,75 segundos, ahora vamos a volver a ingresar el bash para que lo realice con el caché

```
lucas@lucas-VirtualBox:~$ /usr/bin/time -f "%e" dd if=archivo_cache of=/dev/null
 bs=1M
500+0 records in
500+0 records out
524288000 bytes (524 MB, 500 MiB) copied, 0.104643 s, 5.0 GB/s
0.10
```

Podemos apreciar que el sistema operativo lo realizó más rápido, ya que tardo únicamente 0,10 segundos.