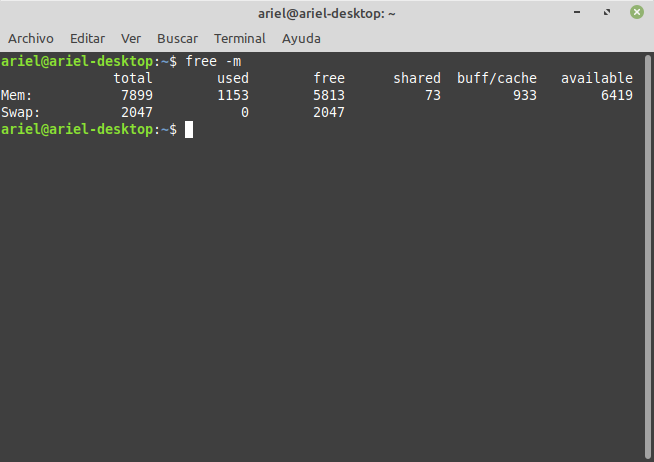
Parte II: Utilizando un equipo con Linux



***1. Memoria física (expresar los resultados en MiB).***

a. Cantidad total: **7899 Mb**

b. Cantidad usada: **1153 Mb**

c. Cantidad libre: **5813 Mb**

d. Cantidad en buff/cache: **933 M**b

e. Cantidad disponible: **6419 Mb**

***2. Memoria intercambiada a disco (expresar los resultados en MiB).***

a. Cantidad total: **2047 Mb**

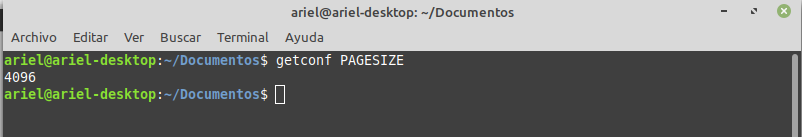
b. Cantidad usada: **0 Mb**

c. Cantidad libre: **2047 Mb**

***3. Averiguar con qué comando se puede conocer el tamaño de página. ¿Qué tamaño tiene cada página de su sistema operativo Linux?***

Se puede usar el siguiente comando:

$ getconf PAGESIZE

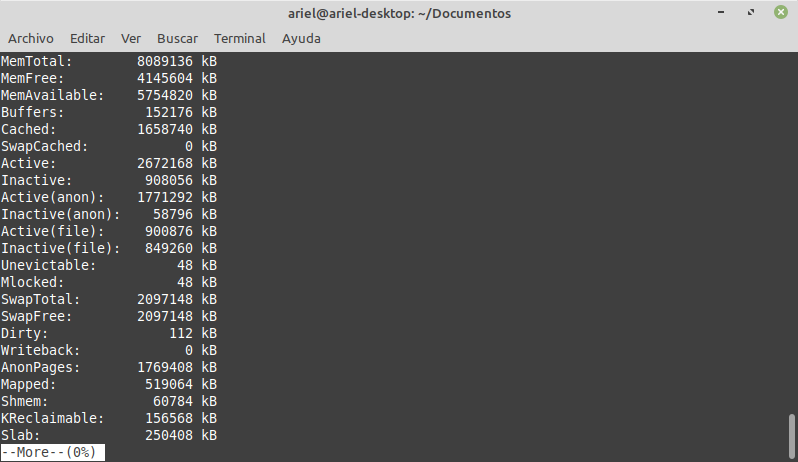


***4. ¿En qué archivo del subdirectorio proc se encuentra la información sobre la memoria? Explique 3 de sus campos***.

El archivo que posee la información sobre la memoria es el archivo **meminfo.**

Se puede acceder a el de la siguiente forma:

$ more *proc*/*meminfo*



A continuación veremos algunos de los campos principales

La información provista en este archivo se podría dividir en forma de estadísticas altas y bajas. En la primera parte del archivo, se ve un resumen de los valores más comunes que generalmente se utilizan.

Algunos de estos valores son:

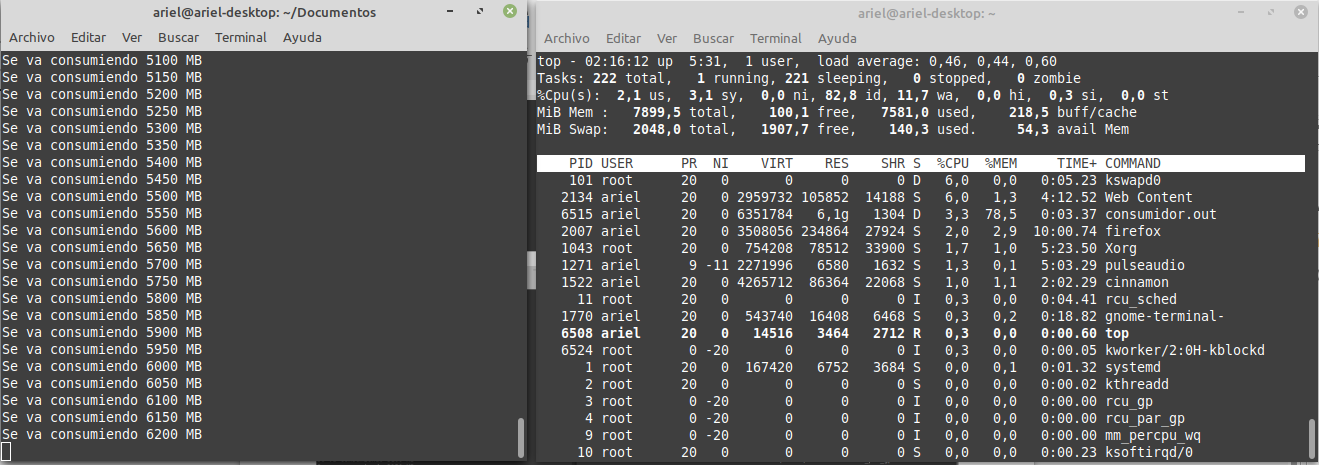
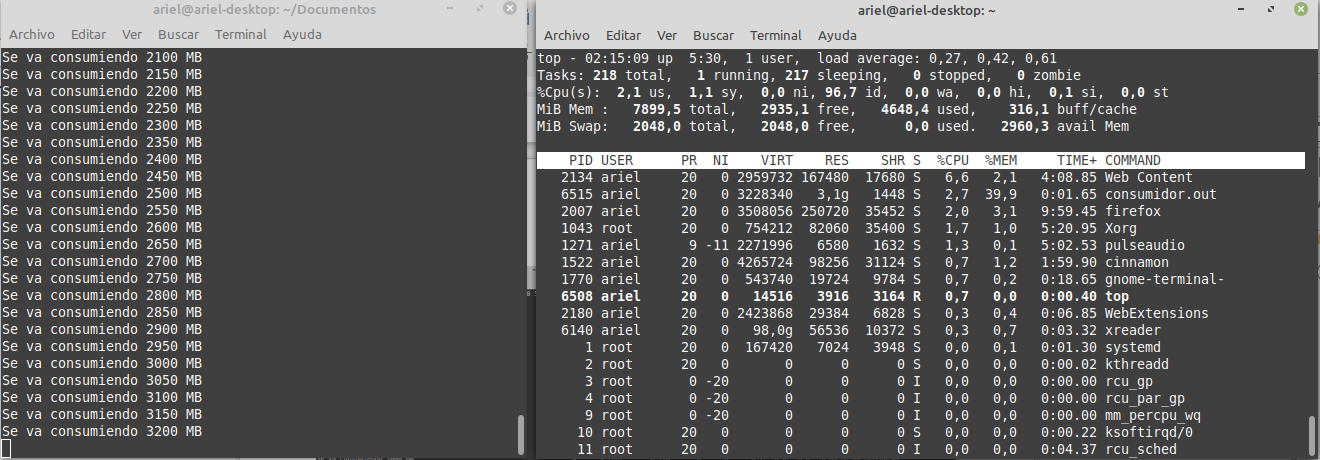
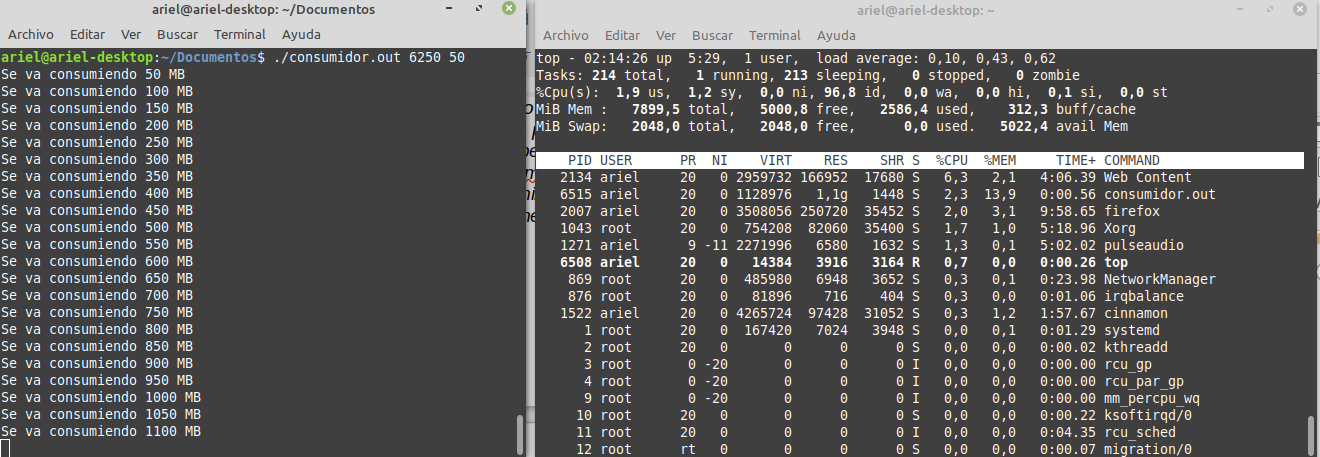
|  |  |
| --- | --- |
| MemTotal | Cantidad total de RAM física. |
| MemFree | Cantidad de RAM física que no ha sido utilizada por el sistema. |
| MemAvailable | Cantidad de memoria estimada que está disponible para iniciar nuevas aplicaciones, sin tener en cuenta el swapping. |
| Buffers | Cantidad de memoria en caché de búfer. Casi no se usa como métrica hoy en día. |
| Cached | Cantidad de RAM física usada como memoria caché. |

***5. Si su máquina tiene 800 MiB de memoria disponible y 300 MiB de memoria libre, de repente necesita ejecutar un programa que ya se sabe que en promedio ocupa 600 MiB al iniciar. ¿El sistema operativo empezará a swappear?***

El swap es un espacio de intercambio, que bien puede ser una partición lógica en el disco o simplemente un archivo. En lugar de utilizar espacio en memoria RAM, el swap utiliza espacio en disco duro para almacenar datos temporales, reduciendo así el uso de la RAM. El conjunto combinado de memoria RAM y swap crean una memoria virtual mayor a la que trae el equipo por defecto. Así, el Kernel de Linux puede ejecutar procesos que requieren de más memoria de la que se encuentra físicamente disponible.

Esto se daría en el caso del ejemplo propuesto, ya que la memoria libre al momento de ejecutar un programa es menor a la la cantidad de memoria que necesita el programa para iniciar. Esa falta de memoria, se soluciona usando el espacio destinado al swap en el disco duro.

***6. Para comprobar lo anterior, en el anexo 4 se deja un programa cuyo único objetivo es solicitar memoria y a esa memoria la establece en cero. El programa cada 1 segundo va solicitando pedazos de memoria. Entonces, llegará un punto que este proceso haya solicitado más memoria que la disponible y el sistema operativo empezará a intercambiar páginas al disco. Utilice las herramientas vmstat y top para monitorear esta situación. Se recomienda utilizar 2 terminales para que en una se vaya ejecutando el proceso consumidor de memoria y en la otra el programa vmstat o top.***

**

Como podemos apreciar en las capturas anteriores, cuando ejecutamos el comando *top,* y la memoria consumida por el script está en valores alrededor de los 1100 Mb, la memoria libre es de 5000,8 y el uso de la memoria Swap es de 0.  
Cuando la memoria usada por el script está alrededor de los 3200 Mb, la memoria libre bajó a 2935,1 y la memoria Swap todavía se mantiene en 0.

En la última captura, cuando el script está por alcanzar su final, la memoria libre está en solo 100,1 y ya podemos observar que la memoria Swap ha tomado un valor de 140,3 Mb.

***a. Utilizando top, responda:***

***i) ¿Qué sucede con la columna VIRT y RES de este proceso? ¿Qué representa cada una?***

En el caso del ejemplo planteado, los valores relevados son los siguientes:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Memoria script | VIRT(Mb) | RES(Gb) | %MEM | %CPU | PR | NI |
| 1100 | 1128976 | 1,1 | 13,9 | 2,3 | 20 | 0 |
| 3200 | 3228340 | 3,1 | 39,9 | 2,7 | 20 | 0 |
| 6200 | 6351784 | 6,1 | 78,5 | 3,3 | 20 | 0 |

Pudimos observar que los valores de las columnas VIRT y RES aumentaban de forma proporcional al consumo de memoria del proceso *consumidor.out*

La columna **VIRT** (**Virtual Memory Size** / Tamaño de Memoria Virtual) muestra la cantidad total de memoria virtual utilizada por la tarea. Incluye todo el código, los datos y las bibliotecas compartidas, además de las páginas que se han intercambiado y las páginas que se han mapeado pero no se han utilizado.

La columna **RES** (**Resident Memory Size /** Tamaño de la memoria residente) Es un subconjunto del espacio de direcciones virtuales (VIRT) que representa a la memoria física no “swapeada” que una tarea está usando actualmente.

***ii) ¿A cuánto llegó la columna %MEM?***

El valor de esta columna fué aumentando a medida que el script iba corriendo, llegando a un valor cercano al 80%

***iii) ¿La columna %CPU se vio muy afectada?***

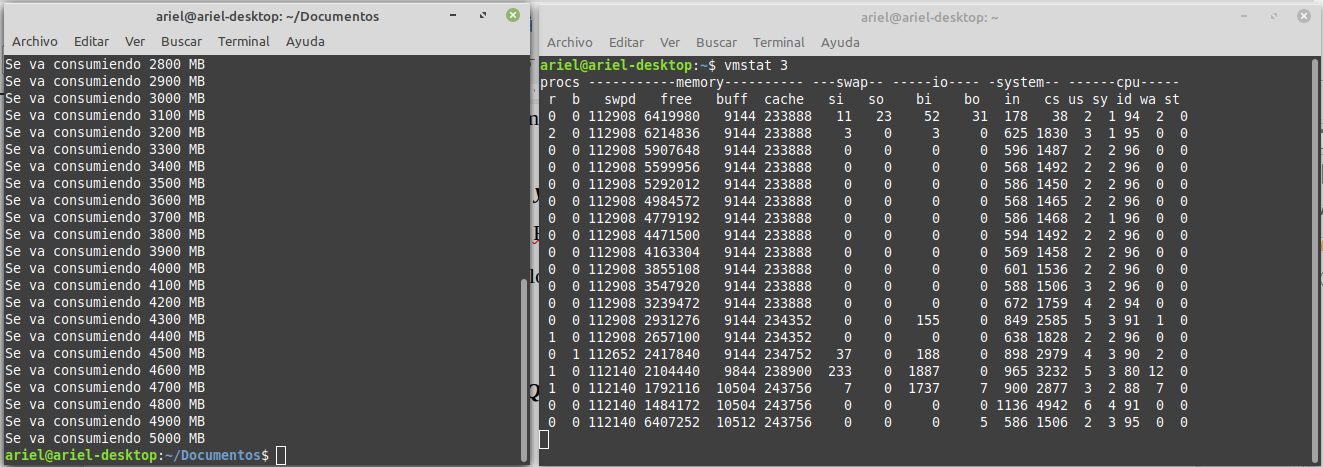
Observando la variación de esta columna, podemos llegar a la conclusión que la misma no se vió afectada por este proceso.

***iv) ¿Qué significan las columnas PR y NI?***

PR: prioridad del proceso. Si aparece RT es que se está ejecutando en tiempo real.

NI: asigna la prioridad. Si tiene un valor bajo (hasta -20) quiere decir que tiene más prioridad que otro con valor alto (hasta 19)

***b. Utilizando vmstat, responda:***

******

***i) ¿Qué sucede con la columna si? ¿Qué representa esta columna?***

Observamos que a medida que transcurre la ejecución del script, los valores de esta columna se mantienen casi constantes (valor 0), salvo en algún momento en donde los mismo aumentan considerablemente (valores de 37 y 233),

Esta columna **SI** indica el número medio de datos de procesos ligeros cargados en la memoria swap por segundo,

***ii) ¿Qué significan las columnas BI y BO?***

**BI** : Bloques recibidos de un dispositivo de bloque. El número de bloques de datos utilizados para intercambiar la memoria virtual de nuevo a la RAM.

**BO** : Bloques enviados a un dispositivo de bloque. El número de bloques de datos que se utilizan para intercambiar la memoria virtual de la RAM al espacio de intercambio.