PRACTICA #8

"MANEJO DEL AREA DE SWAP"

OBJETIVO: Aprender las bases del manejo del área de swap.

INTRODUCCION:

El área de swap también es conocida con el nombre de espacio de intercambio, ésta área se utiliza cuando la memoria RAM es limitada, por lo que es necesario utilizar memoria virtual o área de swap.

Linux soporta dos tipos de área de swap: particiones de swap (device swap) y archivos de swap (file system swap). Una partición de swap es una partición física de disco, la cual tiene un ID de sistema de archivos establecida con el número 82, que es el número que identifica al área de swap de Linux. Un archivo de swap es grande en un sistema de archivos, que se emplea como área de swap.

Memoria Virtual

El manejo de memoria comienza con una emisión compleja de sistemas multiusuarios debido a el total de requerimientos de los usuarios y el sistema operativo, generalmente se exceden de la memoria disponible del sistema. Incluso la rápida reducción de precios de chips de memoria y consecuentemente el crecimiento en la memoria principal no tiene la capacidad de vencer las necesidades de mayores requerimiento de memoria requerida. El tema comienza con todas las soluciones de mas memoria en el uso de una parte del disco común a una extensión de la memoria principal. Llamada memoria virtual, este almacenamiento de memoria provee espacio para todos los usuarios y puede ser expandida como la necesites si el número de usuarios del sistema este configurado para crecer. Desde entonces el CPU puede usar únicamente la memoria principal para ejecutar el código y datos de acceso. El código y datos de componentes de un proceso tiene que ser copiado en la memoria principal a ejecución. Entre los esquemas usados de manejo están los sig.:

1.- Swapping
2.- Demand Paging
3.- Demand Paging y Swapping

I. El sistema Swapping, intercambian, procesos completos (textos, datos, pilas y otras tablas de sistema necesita la administración de un proceso de tiempo que esta en la memoria principal).

Un proceso reside en disco hasta que esta listo para correr otra vez. Cuando hiciste un swap en la memoria principal y se etiqueta basado en su prioridad. Entre mas grande sea la memoria principal disponible, será mas grande el número de procesos que están residentes en memoria.

II. El sistema Demand Paging, por otro lado, no intercambian para un proceso completo. El copiado mínimo necesario para correr los procesos y el uso del concepto de fallas de paginas, para indicar la necesidad de brincar a otra página de los procesos desde memoria virtual en disco a memoria física. Una falla de página ocurre cuando la página requerida no esta físicamente en memoria, y las secciones para leerlo a memoria física desde la memoria virtual son comenzadas. Cada proceso esta parcialmente en memoria todo el tiempo.

III. El sistema Demand Paging y Swapping por otro lado, no solo usa únicamente "demanda de página" porque tiene la capacidad de intercambiar procesos completos, si es necesario para hacer espacio de swapping parcial en otros procesos. Este esquema se beneficia de ambos, tanto del primero como el segundo esquema.

Es más recomendable utilizar una partición de swap en vez de un archivo, porque todo el acceso a un archivo de este tipo se realiza por medio del sistema de archivos de Linux. Como los bloques de disco son los que conforman el archivo de swap, puede llegarse a dar que no estén juntos y, por lo tanto, el rendimiento no es tan bueno como lo sería, con una partición de swap. La E/S a una partición de swap se ejecuta directamente hacia el dispositivo, y los bloques de la partición de disco están siempre juntos, además, al mantener el área de swap fuera del sistema de archivos, reduce el riesgo de corromper el sistema de archivos, si es que algo extraño sucede con el archivo de swap.

El Buffer Cache

Al leer de un disco (excepto de un disco RAM por razones obvias) es muy lento comparado con el acceso a memoria (real) . Y además es común leer la misma parte de un disco varias veces durante un tiempo relativamente corto. Por ejemplo, uno tal vez primero lea un mensaje de un e-mail, entonces lea la carta en un editor cuando se conteste a él. entonces se hará que el programa de correo lo lea otra vez y cuando lo copie hacia un folder, o considere que tan seguido use el comando ls, tal vez sea corrido en un sistema con muchos usuarios. Leyendo la información del disco una sola vez y luego guardándola en la memoria hasta que ya no se necesite, uno puede aumentar la velocidad como en la primera leída. Esto es llamado buffering del disco y la memoria usada para el propósito es llamada el cache del buffer.

Debido a que la memoria es, desafortunadamente finita, el cache del buffer usualmente no puede ser suficiente (no puede contener todos los datos que uno quiera usar). Cuando el cache se llena, la información que no ha sido usada por el más largo tiempo es descartada y la memoria entonces es usada libremente por la nueva información.

El buffering del disco trabaja para escribir también. Por un lado, la información que es escrita es pronto vuelta a leerse (por ejemplo un archivo de código fuente es salvado en un archivo, entonces escrito por el compilador), así poniendo la información que es escrita en el caché es una buena idea. Por otra parte, solo por poner la información en el cache y no escribirla en el disco de una vez, el programa que escribe corre más rápido. Las escrituras pueden llevarse a cabo en background, haciendo más despacio los otros programas.

Muchos de los sistemas operativos tienen caché del buffer (aunque se llamen de otra forma) , pero no todos trabajan de acuerdo a los principios arriba mencionado. Algunos son a través de escritura: la información es escrita en el disco una vez (se mantiene en el caché también, desde luego). El caché es llamado escritura-atrás si los escritos son hechos en un momento más tarde. escritura-atrás es más eficiente que a tráves-escritura, pero también más propensa a errores : si la maquina se colisiona, o la energía es cortada en un mal momento o el disco flexible es removido del drive del disco antes que la información de datos, los cambios en el cache se pierden. Esto tal vez aún signifique que el sistema de archivos (si existe) no este en completo orden de trabajo, tal vez por que la información no escrita contenga importantes cambios a la información contenida en libros.

Debido a esto nunca debe de apagar la energía sin usar un procedimiento de apagado apropiado (sistema de arranque y paro) o remover un floppy del drive de disco (montado y desmontado) o después de que el programa que esté usando señale que ha terminado y la luz del drive del disco flexible este apagado. LINUX tiene un demonio llamado bdflush, que forza que la información sea escrita a disco pero este es un tanto imperfecto como el sync que corre en las tradicionales UNIX, donde en estas se hace un `update` (el comando que forza la información a ser escrita en disco) cada 30 segundos, por lo que el sync ya no es necesario.

Para crear un partición de swap se requiere crear antes una partición de disco con el comando f disk (éste comando, también nos sirve para verificar el estado actual de la partición del disco) y etiquetarla con el ID 82, para que Linux lo reconozca como anteriormente se mencionó.

El tamaño de la partición depende de la necesidad, de que tanta memoria se requiera como memoria virtual, se tienen que crear múltiples particiones de swap, ahora si el sistema de LINUX lo requiere, se debe crear una partición de swap para el sistema y otra para los archivos del usuario, una sugerencia de algunos autores, es la que se llama “Regla del Pulgar”, la cual nos habla, de que para crear el área de swap, tomamos el doble de tamaño de la RAM.

Para Crear la partición formal de swap , se debe de tener la partición ordinaria del disco, después se deben seguir ciertos pasos para hacer la partición activa. El primer paso, es crear la partición en forma similar a la de un sistema de archivos, esto es por medio del comando `mkswap`, cuya sintaxis es:

```
# /sbin/mkswap [-c]  dispositivo tamañoenbloques
```

donde dispositivo es el nombre de la partición de intercambio, como /dev/hda2 , y tamañoenbloques es el tamaño del sistema de archivos de destino en bloques. Para tener una mejor idea podemos ejemplificarlo con este ejemplo:

Tenemos que el tamaño de /dev/hda2 es de 19,159 bloques, ya que Linux requiere que las particiones de área de swap sean de 9 a 65,537 bloques. Se puede obtener el tamaño en

bloques al ejecutar fdisk y observar los datos de la tabla de partición . El argumento de -c le dice a mkswap que revise el sistema de archivos buscando bloques dañados cuando crea el área de swap.

Por lo tanto quedaría:

```
# /sbin/mkswap –c /dev/hda2 19159
```

El siguiente paso después de que se haya creado y preparado la partición, es activarla para que el kernel de Linux la pueda utilizar. El comando para activarla es swapon , cuya sintaxis es:

```
# /sbin/swapon sistemaarchivo
```

donde sistemaa rchivo es el sistema de archivos que se quiere tener disponible como espacio de swap. Linux hace una llamada a swapon -a durante el arranque, lo que monta todas las particiones de swap disponibles que están listadas en /etc/fstab .

Archivo de swapestos son muy útiles si se necesita expandir el área de swap y no es posible asignar espacio en el disco duro para crear una partición mayor. La configuración de un archivo de swap es casi idéntica a la de la partición. La principal diferencia es que se tiene que crear el archivo antes de ejecutar mkswap y swapon .

Para crear un archivo de swap se utiliza el comando dd . Los principales aspectos antes de crear el archivo son el nombre del archivo y su tamaño en bloques. Para Linux un bloque tiene 1,024 bytes de tamaño. Por ejemplo, si se quisiera crear un archivo de nombre /swap se teclearía lo siguiente:

```
# dd if=/dev/zero of=/swap_archivo bs=1024 count=10240
# mkswap /swap_archivo 10240
```

donde of=/swap_archivo especifica el nombre del archivo, y count =10 240 bloques o 10 Megabytes.
Recuerde que se debe decirle a mkswap el tamaño del archivo. Antes de ejecutar swapon sobre el archivo, necesita asegurarse de que el archivo esté completamente escrito en el

disco. Use el comando sync para lograrlo. Ahora si se puede activar el archivo con el comando swapon , para activarlo, lo puede hacer como en el siguiente ejemplo:

```
# sync
# swapon /swap_archivo
```

Si se tiene que deshacerse del archivo de intercambio, asegúrese que no esté activo. Para ello use el comando swapoff , como ejemplo:

```
# swapoff /swap_archivo
```

Entonces ya puede borrar con toda seguridad el archivo.

CONCLUSIONES:

CUESTIONARIO:

1. ¿Qué es el área de swap? ¿Para qué sirve?
2. ¿Cuáles son los tipos de área de swap?
3. ¿Porqué es más recomendable usar una partición de swap?
4. ¿Cuáles son los pasos para crear una partición de swap? Descríbelos detalladamente.
5. ¿Cuáles son los pasos para crear un archivo de swap? Descríbelos detalladamente.
6. ¿Para qué se utiliza el comando `swapoff`?
7. ¿Que hace el comando `dd`?
8. ¿Que función tiene el comando `sync`?