



Universidad Nacional Autónoma de México

Facultad de Ingeniería
Ciudad Universitaria



Sistemas operativos

Escrito/Investigación de Exposición

“Manejo de memoria virtual, espacio de paginación y área swap”

Alumnos:

- Tapia Ledesma Angel Hazel
- Echevarria Aguilar Luis Angel

Profesor: Wolf Iszaevich Gunnar Eyal

Grupo: 08

Semestre 2026-1

Fecha de entrega: 31/08/2025

Manejo de memoria virtual, espacio de paginación y área de swap

Historia de la memoria virtual

En los primeros años de la computación, la memoria era un recurso extremadamente costoso y limitado. Durante las décadas de 1940 y 1950, los equipos solo contaban con una pequeña cantidad de memoria RAM y con dispositivos de almacenamiento secundario, como tambores magnéticos o memorias de núcleos. Estas restricciones imposibilitaron la ejecución de programas de gran tamaño o de múltiples programas en paralelo.

Para enfrentar este problema, los primeros sistemas implementaron estrategias rudimentarias, como la ejecución secuencial de procesos o la limitación estricta de la memoria asignada a cada aplicación. Sin embargo, estas soluciones resultaban poco eficientes y restringían el crecimiento de las capacidades de los equipos.

En 1956, Fritz-Rudolf Guntch introdujo la idea de la memoria virtual, la cual permitiría ampliar artificialmente la memoria disponible a través del uso de almacenamiento secundario. El primer sistema en implementar esta técnica de forma práctica fue el Atlas, desarrollado en la Universidad de Mánchester, que se considera pionero en la adopción de memoria virtual como parte de su arquitectura.

Este avance marcó un hito, ya que permitió que los sistemas operativos manejaran de manera más flexible los recursos de memoria, habilitando la multiprogramación y sentando las bases de los modernos esquemas de gestión de memoria.

¿Qué es la memoria virtual?

La memoria virtual es una técnica mediante la cual un sistema operativo simula disponer de más memoria principal de la que físicamente posee. Esto se logra utilizando parte del disco duro como extensión de la RAM.

Cuando la memoria principal se llena, los datos menos utilizados se transfieren a un espacio del disco llamado archivo de paginación o espacio de intercambio. Gracias a esta estrategia, el sistema puede ejecutar programas de gran tamaño o mantener múltiples procesos en ejecución sin necesidad de una gran cantidad de RAM.

El sistema operativo gestiona automáticamente este intercambio, decidiendo qué datos permanecen en RAM y cuáles se trasladan al disco, de manera que el usuario perciba un entorno continuo de trabajo. Desde la perspectiva de la CPU, la memoria se organiza como un espacio único de direcciones virtuales que luego son traducidas a direcciones físicas en RAM o en disco.

Paginación y su funcionamiento

La paginación es el método más utilizado por los sistemas operativos modernos para implementar la memoria virtual. Su funcionamiento se basa en dividir la memoria en bloques de tamaño fijo. Por un lado, la memoria física (RAM) se organiza en marcos; por otro, la memoria virtual de cada proceso se divide en páginas del mismo tamaño.

Cuando un proceso necesita acceder a un dato o instrucción, la dirección que utiliza no es una dirección física de la RAM, sino una dirección virtual. Esa dirección se compone de dos partes: el número de página y el desplazamiento dentro de esa página. Para transformar la dirección virtual en una dirección física dentro de la RAM, el sistema operativo utiliza una tabla de páginas.

La tabla de páginas es una estructura de datos esencial que mantiene la relación entre páginas virtuales y marcos físicos.

El acceso a la memoria se realiza de la siguiente manera: cuando la CPU solicita una dirección virtual, el sistema operativo consulta la tabla de páginas del proceso en ejecución. Si la página está en RAM, se obtiene de inmediato el marco físico correspondiente y se accede al dato. Sin embargo, si la página no está presente, se produce lo que se denomina un fallo de página. En esta situación, el sistema operativo debe localizar la página requerida en el disco, copiarla a un marco de RAM y actualizar la tabla de páginas para que futuras referencias puedan acceder a ella directamente.

Dado que la memoria RAM es limitada, no siempre existe un marco libre. En ese caso, el sistema operativo debe decidir qué página residente en memoria será reemplazada. Para ello, se emplean

algoritmos de reemplazo de páginas como FIFO (primero en entrar, primero en salir) o LRU (menos recientemente utilizada), que seleccionan de manera eficiente la página menos crítica para liberar espacio.

Entonces decimos que el proceso de intercambio sigue estos pasos:

1. El agotamiento de la capacidad de RAM.
2. La identificación de datos a los que no se ha accedido recientemente.
3. Transferencia de datos desde la RAM a un archivo de intercambio o una partición de memoria de intercambio dedicada.
4. Carga de datos nuevos o de datos desde la memoria de intercambio a la RAM.

Beneficios y desventajas del uso de memoria de intercambio (swap)

El uso de memoria de intercambio o swap trae consigo varias ventajas:

- Optimización de RAM: libera memoria principal al mover datos inactivos al disco.
- Mayor disponibilidad: permite trabajar con procesos más grandes que la memoria física disponible.
- Continuidad operativa: evita bloqueos cuando la RAM se satura.
- Multiprogramación real: habilita la ejecución simultánea de múltiples procesos.
- La probabilidad del uso de esta partición es muy baja, gracias a los algoritmos que se implementan, ya que el uso de memoria secundaria es muy lenta.

No obstante, también presenta limitaciones:

- Menor velocidad: acceder al disco es mucho más lento que a la RAM.
- Cuellos de botella de E/S: el uso intensivo de swap puede ralentizar el sistema.
- Fragmentación del área de swap: reduce la eficiencia del intercambio.
- Dependencia del disco: un fallo en el dispositivo de almacenamiento puede provocar pérdida de datos.
- Puede haber no determinismo, a pesar de los algoritmos, lo cual hace que si se entra en este caso, haya un gran cuello de botella.
- Actualmente, la fragmentación no es gran problema, pero en dispositivos SSD, la sobrescritura es un gran problema, ya que la partición swap puede generar muchas escrituras en una celda, lo que acorta significativamente la vida útil.

El área de swap: aplicación práctica de la memoria virtual

El área de swap es la extensión de la memoria RAM en el disco, destinada a almacenar temporalmente datos que no caben en memoria principal. Su función principal es garantizar que el sistema operativo mantenga estabilidad y continuidad operativa, incluso cuando la RAM se encuentra saturada.

Existen dos formas principales de implementar el área de swap: mediante particiones dedicadas de intercambio o a través de archivos de intercambio dentro del sistema de archivos. Ambas cumplen el mismo propósito, pero presentan diferencias significativas en aspectos como permanencia, rendimiento, seguridad y manejabilidad.

Diferencias entre particiones de intercambio y archivos de intercambio

Las particiones de swap y los archivos de intercambio cumplen la misma función, pero se distinguen principalmente en su permanencia, facilidad de gestión y seguridad.

Las particiones se crean en una zona fija del disco, lo que les da un tamaño estable y un rendimiento más predecible a lo largo del tiempo. Suelen usarse en servidores o sistemas críticos porque resultan más seguras y confiables, aunque carecen de flexibilidad: una vez definidas, no es sencillo modificar su tamaño o eliminarlas sin reconfigurar el disco.

Por otro lado, los archivos de intercambio se gestionan dentro del propio sistema de archivos, lo que facilita su creación, eliminación o redimensionamiento según las necesidades del sistema. Esto los hace ideales en equipos personales o entornos de nube, donde la demanda de memoria puede variar constantemente. Sin embargo, al estar más expuestos dentro del sistema de archivos, presentan una

mayor vulnerabilidad a accesos no autorizados y su rendimiento puede verse afectado en discos fragmentados.

- particiones = estabilidad y seguridad
- archivos = flexibilidad y facilidad de gestión.

Esquemas Alternativos de Memoria de Intercambio

Swap Comprimido (zRAM/zswap)

El swap comprimido representa una evolución significativa en la gestión de memoria virtual. zRAM crea un dispositivo de bloque comprimido en RAM que actúa como área de intercambio, utilizando algoritmos como LZ4, ZSTD o LZO. Teóricamente, este esquema puede almacenar 2-3 veces más datos en el mismo espacio de memoria mediante compresión, reduciendo significativamente las operaciones de E/S a disco. zswap funciona como una caché comprimida transparente entre la memoria principal y el swap tradicional.

Swap Distribuido

Este esquema utiliza múltiples dispositivos de almacenamiento con diferentes prioridades, permitiendo al sistema operativo seleccionar el dispositivo más apropiado según criterios de velocidad, disponibilidad y carga. La teoría detrás de esta aproximación se basa en la distribución de carga de E/S y la redundancia, mejorando tanto el rendimiento como la confiabilidad del sistema.

Swap en Red (Network Block Device)

La implementación de swap sobre dispositivos de red permite externalizar la memoria de intercambio a servidores remotos. Aunque introduce latencia de red, este esquema es útil en entornos de computación distribuida donde los nodos pueden compartir recursos de almacenamiento especializados o cuando se requiere swap en sistemas con almacenamiento local limitado.

Swap Híbrido

Los sistemas híbridos combinan múltiples tecnologías de swap en capas jerárquicas. Típicamente utilizan zRAM como primera capa (más rápida), seguida de SSD como segunda capa y discos mecánicos como última opción. Esta aproximación optimiza el balance entre velocidad, capacidad y costo.

Configuraciones Avanzadas y Optimizaciones

Encriptación de Swap

La implementación de swap encriptado protege datos sensibles que pueden ser intercambiados al almacenamiento. Utiliza sistemas como LUKS en Linux para crear dispositivos encriptados transparentes, añadiendo una capa de seguridad con mínimo impacto en rendimiento mediante aceleración por hardware cuando está disponible.

Fundamentos Técnicos de la Encriptación

El cifrado de swap surge de la necesidad de proteger información sensible que puede ser temporalmente almacenada en el área de intercambio. Cuando el sistema operativo mueve páginas de memoria desde la RAM al disco, datos como credenciales de usuario, tokens de autenticación, claves criptográficas temporales, y contenido de aplicaciones quedan expuestos en texto plano si el swap no está cifrado. Esta vulnerabilidad es particularmente crítica en sistemas que manejan información confidencial o que cumplen con regulaciones de protección de datos.

Implementación con LUKS (Linux Unified Key Setup)

LUKS representa el estándar de facto para cifrado de dispositivos de bloque en sistemas Linux. Su implementación en el contexto de swap involucra la creación de un contenedor cifrado que actúa como intermediario transparente entre el sistema operativo y el dispositivo de almacenamiento físico. El proceso utiliza algoritmos criptográficos robustos como AES-XTS-PLAIN64, que está específicamente diseñado para el cifrado de almacenamiento, proporcionando protección contra ataques de análisis diferencial y manipulación de datos.

La configuración típica emplea claves efímeras generadas aleatoriamente en cada arranque del sistema, lo que garantiza que los datos del swap anterior sean irrecuperables incluso si se compromete el dispositivo físico. Alternativamente, pueden utilizarse claves persistentes cuando se requiere compatibilidad con funciones de hibernación del sistema.

Gestión de Claves y Consideraciones de Seguridad

La gestión segura de claves criptográficas en el contexto de swap presenta desafíos únicos. Las implementaciones modernas utilizan derivación de claves basada en funciones hash criptográficas como SHA-256, que permiten generar claves determinísticas a partir de entropía del sistema o contraseñas de usuario. La integración con sistemas de gestión de claves del sistema operativo, como el kernel keyring de Linux, proporciona un almacenamiento seguro temporal de material criptográfico.

Control de Ancho de Banda

Los cgroups (control groups) permiten limitar el ancho de banda utilizado por operaciones de swap, evitando que procesos específicos saturen el subsistema de E/S. Esta funcionalidad es especialmente importante en sistemas multiusuario o contenedorizados donde la equidad de recursos es crucial.

Arquitectura de Control Groups y Limitación de Recursos

Los control groups representan una infraestructura del kernel Linux que permite la organización jerárquica de procesos y la aplicación de límites de recursos por grupo. En el contexto específico del swap, el subsistema blkio (block I/O) de cgroups proporciona mecanismos granulares para controlar el ancho de banda de dispositivos de almacenamiento utilizados como área de intercambio.

La implementación utiliza algoritmos de limitación de tasa como token bucket y leaky bucket, que permiten controlar tanto el throughput promedio como las ráfagas de actividad. El token bucket permite ráfagas controladas llenando un "depósito" de tokens a una tasa constante, donde cada operación de E/S consume tokens. Cuando se agotan los tokens, las operaciones se bloquean hasta que se generen nuevos tokens, efectivamente limitando la tasa de operaciones.

Aplicaciones en Entornos Virtualizados y Contenedorizados

En entornos de contenedores, el control de ancho de banda de swap previene que contenedores individuales monopolicen los recursos de E/S del host, manteniendo el rendimiento predecible para otros contenedores. Las tecnologías como Docker y Kubernetes integran estos controles a través de especificaciones de recursos en sus definiciones de contenedores.

Los sistemas de orquestación pueden implementar políticas dinámicas de asignación de ancho de banda basadas en prioridades de aplicaciones, patrones de uso históricos, o métricas de rendimiento en tiempo real. Esta gestión inteligente de recursos es fundamental en entornos de nube donde múltiples inquilinos comparten infraestructura física.

Implementación en Sistemas Multiusuario

Los sistemas multiusuario se benefician particularmente del control de ancho de banda de swap para garantizar equidad entre usuarios. La implementación típica crea cgroups específicos por usuario o por clase de servicio, aplicando límites apropiados basados en políticas administrativas. Los

algoritmos de scheduling pueden priorizar usuarios interactivos sobre procesos batch, o proporcionar garantías de ancho de banda mínimo para usuarios premium.

La monitorización continua del uso de ancho de banda por grupo permite ajustes dinámicos de las políticas, detectando y mitigando comportamientos anómalos que podrían impactar el rendimiento general del sistema. Las herramientas de análisis pueden identificar patrones de uso y recomendar optimizaciones en la asignación de recursos.

Algoritmos de Compresión Adaptativos

La selección adaptativa de algoritmos de compresión requiere el análisis en tiempo real de las características de los datos que se van a comprimir. Los sistemas implementan heurísticas basadas en métricas como la entropía de Shannon, la detección de patrones repetidos, y el análisis de la distribución de bytes para clasificar el contenido y predecir la efectividad de diferentes algoritmos de compresión.

La entropía de Shannon proporciona una medida de la aleatoriedad de los datos, donde valores bajos indican datos altamente estructurados (como texto o datos tabulares) que se benefician de algoritmos de compresión más agresivos, mientras que valores altos sugieren datos ya comprimidos o aleatorios donde algoritmos rápidos con menor overhead son más apropiados.

Características y Optimizaciones de Algoritmos Específicos

LZ4: Se posiciona como el algoritmo de elección para sistemas con recursos de CPU limitados o requisitos de latencia estrictos. Su diseño prioriza la velocidad de compresión y descompresión sobre la eficiencia de compresión, logrando tasas de descompresión que pueden exceder los 2GB/s en hardware moderno. La implementación utiliza técnicas de búsqueda de coincidencias optimizadas y estructuras de datos diseñadas para minimizar el acceso a memoria.

ZSTD (Zstandard): Representa un balance óptimo entre velocidad y eficiencia de compresión. Su arquitectura permite ajuste dinámico del nivel de compresión, desde configuraciones ultrarrápidas comparables a LZ4 hasta configuraciones de alta compresión que rivalizan con algoritmos más lentos. La capacidad de entrenamiento con diccionarios específicos del dominio puede mejorar significativamente la eficiencia de compresión en datos con patrones recurrentes.

LZO: Mantiene relevancia por su amplia compatibilidad y consistencia de rendimiento en diferentes arquitecturas de hardware. Su diseño maduro y estable lo hace apropiado para sistemas embebidos o entornos donde la predictibilidad del comportamiento es más importante que el rendimiento máximo.

Mitigación de Desventajas Mediante Mecanismos Especializados

Optimización para SSD

Los SSD presentan desafíos únicos debido a su limitado número de ciclos de escritura. Las estrategias de mitigación incluyen el uso de zRAM para reducir escrituras físicas, configuración de swappiness muy bajo (1-10), e implementación de wear leveling a nivel de sistema operativo. Algunos sistemas utilizan tmpfs para directorios temporales, reduciendo la presión sobre el swap.

Monitoreo y Alertas Proactivas

Los sistemas de producción implementan monitoreo continuo del uso de swap mediante scripts automatizados y herramientas de sistema. Estos mecanismos pueden disparar alertas cuando el uso excede umbrales predefinidos, permitiendo intervención antes de que el rendimiento se vea comprometido significativamente.

Gestión Térmica y de Energía

En dispositivos móviles y sistemas embebidos, el swap puede configurarse para considerar restricciones térmicas y de batería. Los algoritmos adaptativos pueden reducir la actividad de swap

durante condiciones de alta temperatura o baja batería, priorizando la estabilidad del sistema sobre el rendimiento máximo.

Tendencias y Mejores Prácticas Actuales

La evolución hacia sistemas con mayor cantidad de RAM ha modificado las estrategias de swap. Las mejores prácticas actuales favorecen configuraciones híbridas que combinan pequeñas cantidades de zRAM para casos comunes con swap tradicional como red de seguridad. En entornos de contenedores, se prefiere la gestión de swap a nivel de host en lugar de swap individual por contenedor, optimizando la utilización de recursos y simplificando la administración.

Los sistemas de nube implementan swap dinámico que se ajusta automáticamente según la demanda de carga de trabajo, mientras que los servidores de alto rendimiento utilizan configuraciones estáticas optimizadas para cargas de trabajo específicas. Esta diversificación refleja la madurez del campo y la necesidad de soluciones especializadas para diferentes casos de uso.

Funcionamiento del swap

El funcionamiento del área de swap está íntimamente relacionado con el principio de la memoria virtual y la paginación. Cuando un sistema operativo ejecuta programas, cada uno de ellos requiere una porción de memoria RAM para poder llevar a cabo sus operaciones. Sin embargo, debido a que la RAM es limitada, llega un momento en que la memoria disponible no es suficiente para mantener todos los procesos activos al mismo tiempo. Es en este punto cuando entra en juego el área de swap.

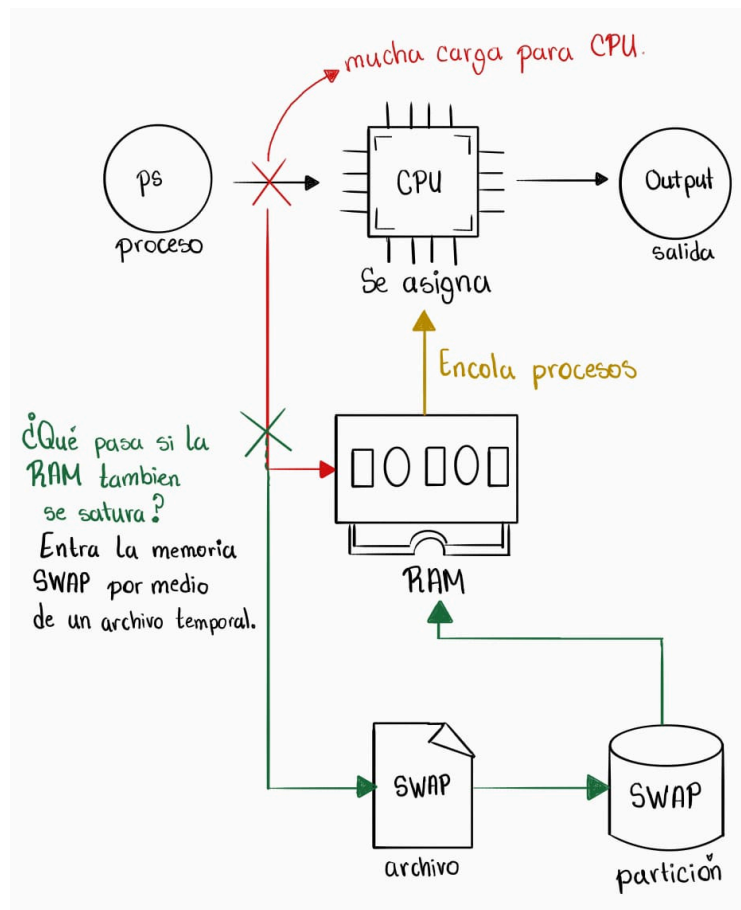
El sistema operativo monitoriza constantemente el uso de la memoria y decide qué información es más importante mantener en RAM y qué datos pueden trasladarse de manera temporal al disco. Por lo general, se prioriza mantener en memoria principal las páginas de datos o instrucciones que se utilizan con mayor frecuencia, mientras que aquellas que no han sido consultadas recientemente son transferidas al área de swap. Este proceso se conoce como swapping out, es decir, sacar temporalmente información de la RAM hacia el disco.

Cuando posteriormente una aplicación necesita acceder a datos que han sido desplazados al área de swap, ocurre un fallo de página. En ese momento, el sistema operativo interrumpe la ejecución, localiza la información en la zona de intercambio, la trae de vuelta a la RAM y actualiza la tabla de páginas del proceso. A este proceso de recuperación se le denomina swapping in. Aunque este intercambio garantiza que el sistema no se quede sin memoria disponible, implica un tiempo de acceso mayor, ya que la lectura y escritura en disco son más lentas que en la RAM.

El swap puede funcionar bajo distintos esquemas. En el modelo de swapping completo de procesos, el sistema puede mover procesos enteros entre la RAM y el área de intercambio, liberando espacio de manera inmediata, pero con un mayor coste de tiempo. En contraste, la paginación bajo demanda se centra únicamente en mover las páginas necesarias en el momento en que son requeridas, lo que resulta más eficiente en la mayoría de los casos. Algunos sistemas combinan ambas técnicas, permitiendo que, en condiciones normales, solo se realicen intercambios de páginas, pero recurriendo al intercambio completo cuando la memoria está críticamente saturada.

El proceso de swap no solo asegura la continuidad de la ejecución, sino que también protege la estabilidad del sistema. Sin esta técnica, un exceso de procesos en ejecución llevaría al colapso inmediato, ya que la RAM sería insuficiente para sostener la carga de trabajo. Aun así, es importante subrayar que el intercambio entre RAM y disco no sustituye la necesidad de contar con memoria física adecuada, pues abusar del swap degrada el rendimiento. Cuanto más frecuentes sean los intercambios, más lento se percibirá el sistema, generando lo que se conoce como thrashing, una situación en la que el equipo pasa más tiempo trasladando páginas entre la RAM y el disco que ejecutando procesos.

En síntesis, el funcionamiento del swap se basa en un equilibrio dinámico: libera memoria principal al mover datos inactivos al disco, recupera páginas cuando son necesarias y, mediante algoritmos de reemplazo, decide qué información mantener en RAM para garantizar que los procesos críticos puedan seguir ejecutándose. De esta forma, el área de intercambio actúa como un colchón de seguridad que, aunque más lento que la memoria física, resulta indispensable para la estabilidad y la multitarea en los sistemas operativos modernos.



Relevancia en la práctica

En Linux, la partición de swap se identifica con el ID de sistema 82. Su tamaño depende de la RAM y de la carga de trabajo; una recomendación clásica es “a rule of thumb” (regla en general), que sugiere asignar el doble de la RAM, aunque en sistemas modernos esta regla se adapta según necesidades reales. Dicha regla se refiere a un principio o método práctico para la toma de decisiones que se basa en la experiencia, se toma por simplicidad como si se tratara de un atajo sin caer en teoría previa o cálculos precisos.

Tamaño del área de swap

La cantidad de swap recomendada varía según la RAM:

Amount of RAM installed in system	Recommended swap space	Recommended swap space with hibernation
≤ 2GB	2X RAM	3X RAM
2GB – 8GB	= RAM	2X RAM
8GB – 64GB	4G to 0.5X RAM	1.5X RAM
>64GB	Minimum 4GB	Hibernation not recommended

En sistemas antiguos se aplicaba la regla de asignar el doble de la RAM como swap, pero en equipos modernos con decenas o cientos de GB, esta recomendación resulta obsoleta.

Herramientas y comandos en Linux

Usaremos:

Comandos del sistema:

- `free -h` - Muestra uso de memoria RAM y swap en formato legible

- parted - Editor de particiones más avanzado que fdisk
- fdisk - Editor básico de particiones de discos
- partprobe -s - Actualiza la tabla de particiones en el kernel sin reiniciar
- mkswap - Formatea una partición como área de intercambio (swap)
- swapon - Activa particiones o archivos swap
- swapon -s - Lista los dispositivos swap activos
- swapon -a - Activa todos los swaps definidos en /etc/fstab
- vim - Editor de texto en terminal
- cat - Muestra el contenido de archivos
- systemctl reboot - Reinicia el sistema usando systemd
- reboot - Reinicia el sistema (comando tradicional).

Opciones del menú fdisk:

- p - Print (muestra la tabla de particiones)
- n - New (crea nueva partición)
- t - Type (cambia el tipo de partición)
- w - Write (guarda los cambios al disco)

Archivos de configuración:

- /etc/fstab - Archivo que define qué dispositivos montar automáticamente al arranque

Tipos de partición:

- 82 - Código hexadecimal para partición Linux swap
- Partición primaria - Partición básica del disco (máximo 4 en MBR)

Ejemplo para crear una partición swap:

Se busca crear una partición swap de 300 MB y que arranque automáticamente (Sabemos que es un espacio asignado pequeño, pero es para entender su configuración).

Procedimiento en línea de comandos linux:

//Primero debes asegurarte de ser usuario root.

#free -h // Muestra el uso actual de memoria y swap en formato legible.

#parted /dev/vdb unit MB print free //Muestra la tabla de particiones y espacio libre en MB.

#fdisk /dev/vdb //Abre el editor de particiones para el disco /dev/vdb.

//Ya dentro del menú interactivo (editor de particiones) seleccionar lo siguiente:

- p //Imprime/muestra la tabla de particiones actual.
- n //Crea una nueva partición.
- p //Selecciona partición primaria.
- enter //Número de partición dejarlo por default
- enter //Sector inicial dejarlo por default
- +300M //Se define el tamaño de la partición
- p //Muestra la nueva partición creada
- //Buscaremos cambiar el tipo de partición que se creó inicialmente.
- t //Cambia el tipo de partición.
- 2 //Selecciona la partición 2 (default), ya que es sobre la que queremos trabajar.
- 82 //Define el tipo como Linux swap
- p //Verifica que se haya modificado el tipo de partición a swap
- w //Escribe/guarda los cambios en el disco.

#partprobe -s //Actualiza la tabla de particiones en el kernel.

#mkswap /dev/vdb2 //Formatea la partición como swap

#swapon -s //Muestra los dispositivos swap activos

#swapon /dev/vdb2 //Activa el dispositivo swap específico.

#swapon -s //Muestra los dispositivos swap activos

#free -h //Verifica que el swap esté funcionando.

#vim /etc/fstab

//Modificar el archivo de config. manualmente para que se mantenga agregando:.

- /dev/vdb2 none swap defaults 0 0 //hasta abajo
- //Guardar y salir de vim

```
#swapon -a //activa todos los swaps del fstab para probar el dispositivo swap
#cat /etc/fstab //verificamos que se haya guardado bien la modificación en el fstab.
#systemctl reboot o #reboot //Reiniciar el sistema.
//Volverse a conectar al servidor o iniciar sistema.
```

¿Se puede activar / desactivar esta partición?

La respuesta es si, podemos eliminar la partición de swap, tanto de linux como de windows, aunque en windows esta partición no se llama “swap”.

1. Linux: En linux podemos “desactivarlo” de una forma muy sencilla, ya que solo tenemos que ir a los puntos de montaje, en el archivo /etc/fstab y comentar la línea que tenga como sistema de archivos “swap”, aunque esto no eliminará la partición, simplemente la “desactiva”, para eliminar y liberar el espacio, deberemos borrar la partición, ya sea con entornos gráficos como gparted o con línea de comandos con herramientas como fdisk
2. En windows, el equivalente a la partición swap es el “archivo de paginación”, para esto debemos buscar en el buscador de windows “panel de control”, después ir a “sistema y seguridad”, una vez dentro, buscamos “sistema” y ahí “ver la cantidad de memoria RAM y velocidad del procesador”, una vez en esta ventana, seleccionaremos “Configuración avanzada del sistema” y veremos una nueva ventana, en esta ventana seleccionaremos la pestaña de “opciones avanzadas” y buscaremos “rendimiento”, una vez hallado, presionaremos en “configuración”, iremos a la pestaña de “opciones avanzadas” y aquí es donde encontraremos el apartado de “memoria virtual” una vez dentro de esta ventana, podemos hacer que el Sistema elija el tamaño o hacerlo de forma manual, incluso desactivar este archivo de paginación.
3. En mac, el equivalente es el “swapfile”, para eliminarlo debemos abrir la terminal y ejecutar el comando:

```
sudo launchctl unload -w /System/Library/LaunchDaemons/com.apple.dynamic_pager.plist
```

Manejo de swap sobre un dispositivo cifrado

Cómo vimos anteriormente, la partición swap es una especie de “extensión” de nuestra memoria RAM, y al pertenecer al disco duro, esta puede ser candidata a el cifrado. Pero ahora es importante pensar ¿Es recomendable? y la respuesta es sí, ya que, cuando nosotros usamos como una extensión de la RAM, nuestro disco duro, constantemente se está escribiendo información ahí de forma no cifrada, por lo que si los algoritmos deciden enviar a esta partición datos sensibles como credenciales web o contraseñas, esta partición no tendrá una protección de cifrado como el resto del disco, por lo que es importante y recomendable que esta partición esté cifrada. Actualmente, la “pérdida” de rendimiento por cifrar los datos es totalmente insignificante, por lo que no es un factor importante o limitante a la hora de tomar una decisión acerca del cifrado de datos

Conclusión

La memoria virtual y el área de swap representan soluciones ingeniosas para superar las limitaciones físicas de la memoria RAM. Gracias a estos mecanismos, los sistemas operativos modernos pueden ejecutar múltiples programas de forma simultánea y manejar procesos mucho más grandes de lo que permitiría la memoria principal por sí sola.

La paginación se encarga de organizar y traducir direcciones, permitiendo que cada proceso trabaje como si tuviera su propio espacio de memoria independiente, mientras que el swap funciona como un respaldo en disco que almacena temporalmente la información menos usada. Aunque este proceso es más lento que el acceso directo a la RAM, garantiza que el sistema no se detenga por falta de memoria, manteniendo la estabilidad y continuidad de la operación.

En términos prácticos, la elección entre particiones o archivos de swap depende del entorno. Las particiones ofrecen mayor seguridad y rendimiento en servidores y sistemas críticos, mientras que los archivos brindan flexibilidad en equipos personales y entornos de nube. Sin importar el caso, el área de intercambio es una pieza clave dentro de la administración de memoria.

En conclusión, el swap no es un sustituto de la memoria RAM, pero sí un aliado fundamental que actúa como “válvula de escape” cuando los recursos físicos no son suficientes. Su correcta configuración permite aprovechar al máximo el hardware disponible y asegura un equilibrio entre

rendimiento, seguridad y disponibilidad, lo que convierte a la memoria virtual en uno de los pilares más importantes del funcionamiento de los sistemas operativos actuales.

Referencias

- Alonso. R. (Agosto, 2025). “¿Qué es el archivo de paginación? ¿Necesitas tenerlo en tu PC?”. Recuperado el 31 de agosto de 2025 de: <https://hardzone.es/noticias/juegos/que-pasaria-gamers-jugando-simultaneos/>
- AIX. (2024, November 5). Ibm.com. <https://www.ibm.com/docs/es/aix/7.2.0?topic=management-paging-space-virtual-memory>
- ¿Cómo se administra la memoria física limitada mediante la memoria virtual y la paginación? (2023). LinkedIn.com. <https://es.linkedin.com/advice/0/how-do-you-manage-limited-physical-memory-using?s&lang=es>
- GeeksforGeeks. (2017, February 10). Virtual Memory in Operating System. GeeksforGeeks. <https://www.geeksforgeeks.org/operating-systems/virtual-memory-in-operating-system/>
- 9.15.5. Esquema de particionamiento recomendado | Guía de Instalación | Red Hat Enterprise Linux | 6 | Red Hat Documentation. (2025). Redhat.com. https://docs.redhat.com/es/documentation/red_hat_enterprise_linux/6/html/installation_guide/s2-diskpartrecommend-x86
- Archivo swap: ¿qué es y por qué es necesario? | Lenovo México. (2021). Lenovo.com. https://www.lenovo.com/mx/es/glosario/archivo-swap/?orgRef=https%253A%252F%252Fwww.google.com%252F&srsltid=AfmBOoppkQvJpL0hZ5kl_FGEDRAwDDR7YhdSqvgz3KLGBa2MVETFK-h
- Zivanov, S. (2023, August 30). Swap Memory: What It Is & How It Works. Knowledge Base by PhoenixNAP. <https://phoenixnap.com/kb/swap-memory>
- de swap., O. A. las B. del M. del Á. (s/f). “MANEJO DEL AREA DE SWAP”. Unam.mx. Recuperado el 1 de septiembre de 2025, de <https://estigia.fi-b.unam.mx/Linux/archive/pdf/prared08.pdf>
- Orovengua J. (Mayo, 2023). “¿Cómo deshabilitar la partición SWAP permanentemente en linux?”. Recuperado el 7 de septiembre de 2025 de: <https://www.linuxparty.es/75-hardware/11564-como-deshabilitar-la-particion-swap-permanentemente-en-linux.html>
- Valero C. (Agsoto, 2025). “Cómo gestionar la memoria virtual en windows 10 y 11”. Recuperado el 7 de septiembre de 2025 de: <https://www.adslzone.net/esenciales/windows-10/archivo-paginacion-windows-10/>
- NotFromBrooklyn. (Mayo, 2015). “cifrar swap”. Recuperado el 7 de septiembre de 2025 de: <https://notfrom.wordpress.com/2015/05/26/cifrar-swap/>
- Nelson J. (Marzo, 2016). “What is the mac swapfile? Can you delete it?”. Recuperado el 8 de septiembre de 2025 de: <https://notebooks.com/2016/03/09/what-is-the-mac-swapfile-can-you-delete-it/>
- zswap — The Linux Kernel documentation. (s/f). Kernel.org. Recuperado el 7 de septiembre de 2025, de <https://docs.kernel.org/admin-guide/mm/zswap.html>
- Physical Memory — The Linux Kernel documentation. (s/f). Kernel.org. Recuperado el 6 de septiembre de 2025, de https://docs.kernel.org/mm/physical_memory.html
- Pons, J. (2023, julio 6). La memoria swap en Linux: esencial para un rendimiento óptimo. Aprendo Linux. <https://aprendolinux.com/la-memoria-swap-en-linux/>
- Introducción, 1. (s/f). 9. Administración de Memoria: Memoria Virtual. Edu.uy. Recuperado el 9 de septiembre de 2025, de https://www.fing.edu.uy/tecnoinf/mvd/cursos/so/material/teo/so09-memoria_virt_uai.pdf