Swap in Linux: Swap Space, Swap Partitions, Swap Files, Swapiness and more

Introduction:

The concept starts with **Swap Space** in Linux and Windows operating systems. **Swap space** in Linux is an extension of physical RAM, offering virtual memory that helps maintain system stability and performance. It allows **processes to continue running** when **RAM is fully** used and prevents memory errors. Swap space also enables hibernation and safeguards critical processes by temporarily offloading data.

> **ChatGPT definition:** Linux, swap space is a form of virtual memory that acts as an extension of RAM. When the physical RAM is fully utilized, Linux moves inactive pages (data) from RAM to swap to free up space for active processes.

However keep this in mind that if we use the secondary storage as RAM alternatives then we suffer from **performance degradation**. This feature allows an operating system to temporarily move **inactive or less frequently used memory pages** from RAM to a designated area on the hard drive. So we can think of swapping as the **opposite of Caching**.

Operating systems like Windows or Linux provide a certain amount of swap space by default, which users can later change in accordance with their requirements. Users can also disable swap space, but that means that the kernel must kill some processes to create enough free RAM for new processes.

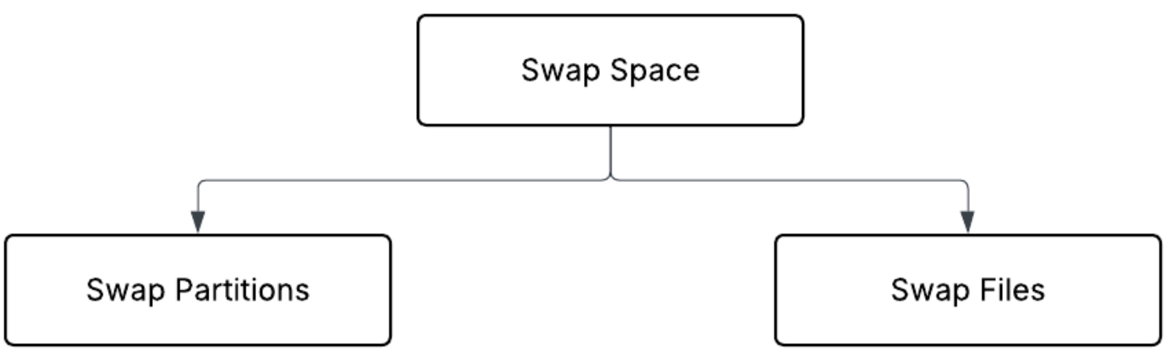

Figure: Swap Space types

Swap Space Overview:

SN	Category	Swap Partitions	Swap Files
1	Location	A swap partition is a dedicated section of a storage device that is reserved solely for use as virtual memory.	Located on an existing filesystem (***often within the root directory or a designated swap directory***) that serves as virtual memory.
2	Creation	Created during the initial partitioning and formatting of the storage device, typically during the operating system installation process.	Created as a regular file on an existing partition. It can be resized or even deleted if required.
3	Advantages	Better performance, stability, and resilience.	Flexibility, dynamic allocation, easy management.
4	Disadvantages	Inflexible size, potential wasted space.	Slightly lower performance, susceptible to file system issues.
5	Management	Requires repartitioning to change size.	Easily resizable and manageable through file operations.
6	Usecase	Commonly used in servers and **high-performance systems.**	Preferred for desktops, laptops, and **cloud environments**.

Swap frees up RAM for more important tasks that require more processing power by transferring data to and from a designated disk space. The data interchange is called swapping, while the designated space is called **swap space**. The swapping rate and assertiveness are determined by a parameter called **Swappiness**. **Swappiness does not control the swap space size but instead influences the swapping limits and frequency.**

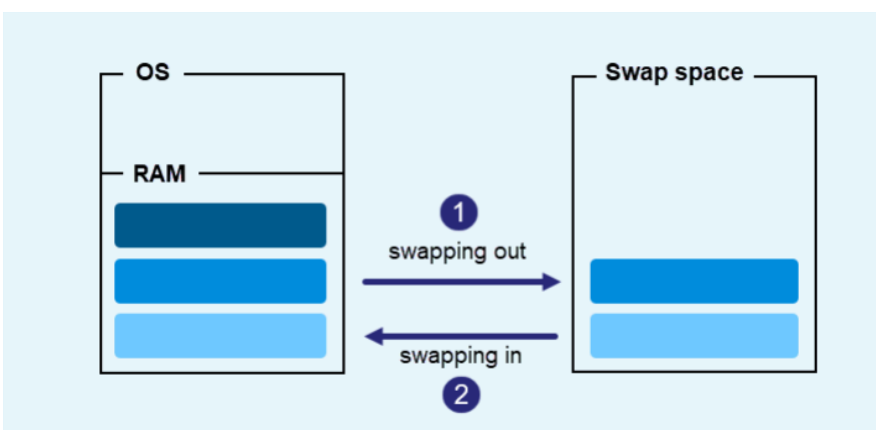

Fig: Swap space as an extension of RAM

Swappiness:

Swappiness is a **system-wide setting**, not specific to individual applications. It applies to how the entire operating system handles memory management and swap usage.

Swapiness is a value that is defined internally. Lower swappiness values prioritize keeping data in RAM. On the other hand, higher swappiness values result in assertive swapping. Summary is below:

Swappiness Value	Swappiness Effect
0	The value instructs the kernel to avoid swappiness as much as possible.
10-50	The value tells the kernel to be slightly aggressive in swapping out memory pages.
50-100	The value instructs the kernel to be moderately aggressive in swapping out memory pages.
>100	The higher values instruct the kernel to be very aggressive in swapping out memory pages.

Swappiness Value depends upon the following factors:

- System configuration.
- Workload.
- Memory.
- Performance requirements.

Heres a command to check for Swapiness: cat /proc/sys/vm/swappiness

```
manojgautam@gen-np-td-mg:~$ cat /proc/sys/vm/swappiness
60
```

Alternative command: sysctl vm.swappiness

```
manojgautam@gen-np-td-mg:~$ sysctl vm.swappiness
vm.swappiness = 60
```

Changing Swapiness Values:

```
sudo sysctl vm.swappiness=[value]
OR
sudo nano /etc/sysctl.conf
OR
vm.swappiness = [value]
```

Fine-tuning swappiness in Linux can improve system performance. However, the task is not straightforward and requires trial and error.

Some Low level Details:

As this parameter sets the kernel's balance between reclaiming pages from the page cache and reclaiming pages by swapping out process memory, a discussion of page reclaim overall is warranted. The reclaim code works (in a very simplified way) by calculating a few numbers:

The distress value is a measure of how much trouble the kernel is having freeing memory. The first time the kernel decides it needs to start reclaiming pages, distress will be zero; if more attempts are required, that value goes up, approaching a max value of 100.

The mapped_ratio value is an approximate percentage of how much of the system's total memory is mapped (i.e. is part of a process's address space) within a given memory zone.

And vm_swappiness is the swappiness parameter, set to 60 by default.

We calculate the parameter named swap tendency from the above values.

```
swap_tendency = mapped_ratio/2 + distress + vm_swappiness;
```

If swap_tendency is below 100, the kernel will only reclaim page cache pages. Once it goes above that value, however, pages which are part of some process's address space will also be considered for reclaim. So, if the system's distress value is low and swappiness is set to 60, the system will not swap process memory until 80% of the total RAM in the system is allocated. Users who would like to never see application memory swapped out can set swappiness to a low value, say 5 or 10, causing the kernel to ignore process memory until the distress value gets quite high.

Overall, increasing this value will make the system more inclined to utilize swap space, leaving more memory free for caches. Decreasing this value will make the system less inclined to swap, and may improve application responsiveness.