Swap area (swap): application and security features

The need to use swap

The presence of the swap area is an important component of the memory management system, and is necessary for the normal functioning of the system .

Wherein

Info

n Acquiring backup RAM is not the primary purpose of paging. Its main purpose is to ensure the effective release and balancing of the use of available memory.

Warning

In fact, using the swap space as an “extra backup volume” is an abnormal situation, and requires measures to expand the hardware.

n Linux-systems, there are different types of pages of RAM, each of which has its own characteristics, but two types are essential for understanding the need for paging:

1. Pages that can be restored by re-reading this content from files (pages with commands of executable processes, caches of their file data), so-called file cache;
2. Pages of data on the distribution of memory between processes, the so-called. anonymous pages with no source files.

And the main purpose of the swap area is to free space in the memory for the file cache by unloading irrelevant anonymous pages. Thus, deleting the swap area will not prevent an increase in the total number of disk operations when the RAM is full, but simply replaces the disk operations of crowding out anonymous pages with repeated ones disk file read operations. This is not only less effective, since the choice of pages to update is less, but, in turn, also leads to even more memory overflow.

Additionally, the swap area is usually used to organize sleep modes (hibernation or suspend to disk). When entering these modes in the swap area, a full copy of the RAM is saved. (In modern systems, you can provide a sleep mode without using the paging area, while preserving the contents of RAM in files).

Select and configure paging options

Placing swap areas: disk partition or file?

Modern Linux-based systems allow you to place the swap area in a dedicated disk partition, and in files. Regardless of the method of placement, modern versions of the Linux kernel provide approximately the same speed of paging, defining paging files, and using direct access to them. When working with disk drives, the fragmentation of these files can be a significant factor affecting the paging speed when working with files, so disk drives can be preferable for disk drives. When using solid-state drives file fragmentation is indifferent. In any case, the ability to dynamically disable / connect file swap areas makes it possible to efficiently use disk space by resizing the swap area as needed without interrupting system operation.

Info

When installing RavanOS, by default, the swap area is automatically created in a separate disk partition. You can check where the swap area (s) are located (placed) with the command

```
sudo swapon
```

Info

If you need to check file fragmentation, use the filefrag command:

```
sudo filefrag -v /example.swap
```

To reduce file fragmentation in the Ext4 file system, use the e4defrag command:

```
sudo e4defrag -v /example.swap
```

Warning

When placing files on solid-state drives, defragmentation does not make any sense.

Swap Area Size

In general, for RavanOS, the correct initial choice may be a 2-3GB swap area, followed by refinement based on the results of operation. If you intend to use the sleep mode (hibernation or suspend to disk) while preserving the RAM image, then the required volume depends on the RAM size, and in this case, a good initial version of the size of the swap area is a volume equal to the RAM size plus 2-3GB.

The main parameter for swapping is the value of the kernel parameter `vm.swappiness` , which determines the amount of free memory (in percent) at which page displacement begins. You can check the value of this parameter with the command:

```
```sudo sysctl vm.swappiness```
```

Swap setup

By default, the value of `vm.swappiness` is 60, i.e. paging out of memory pages begins when free memory becomes less than 60%.

It is believed that this value works well for most systems. You can change the value of the `vm.swappiness` parameter by the command:

```
```sysctl -w vm.swappiness = &amp;lt;new_value&amp;gt;```
```

The minimum attribute value at which the swap will work is one.

> **Info**
>
> If you set the vm.swappiness parameter to zero, the swap is turned off completely.

Security data in the paging area

Memory pages that are dynamically copied from RAM to the paging space during computer operation may contain confidential information that is not covered by any protective transformations. Thus, having read access to the paging area creates the risk of leakage of confidential information.

A separate problem from the point of view of security is storage in the swap area of the contents of the RAM of computers that are turned off. A copy of the contents of RAM can remain in the swap area:

- in case of unexpected shutdown of the computer as a result of a hardware failure;
- in case of unexpected power failure;
- and always remains there in computers that are in sleep mode.

If there is physical access to the equipment, such data can be read regardless of the established discretionary and mandatory limitations.

Paging Security Guidelines

In RavanOS (Smolensk 1.6, Leningrad 8.1), the paging section created during system installation is automatically assigned a security label with maximum levels of confidentiality and integrity and the maximum set of access categories:

- Privacy Level 3;
- Integrity level 63;
- The set of access categories is complete.

Thus, when installing RavanOS (Smolensk 1.6, Leningrad 8.1), unprivileged users are automatically protected from reading the swap space.

However, the system administrator has the option to add his own swap areas after installation `(see the mkswap / swapon / swapoff commands, as well as the swapon () / swapoff () system calls)` . At the same time, the operating system does not check the access rights set for the paging area being added, and the administrator should independently specify the necessary restrictions on discretionary and mandatory access rights. For example, the following command sequence can be used to place the paging space in the / swap_area file:

Info

```
#create a 1Gb file  sudo fallocate -l 1G /swap_area

#restrict discretionary access rights (for RavanOS SE /RavanOS CE)  sudo chown root: root /swap_area
sudo chmod 600 /swap_area

#Limit mandated access rights (only for the Linux operating system CH SE the RavanOS)  #-privacy 3
#-integrity level 63  #-open all cat e gorii access

sudo pdpl-file 3: 63: -1 /swap_area

#mark up the swap area sudo mkswap / swap_area

#include swap space in

sudo swapon /swap_area
```

To ensure the removal of information from the paging area when taking this paging area out of work, you can use:

for magnetic disk drives (applicable only in OS RavanOS):

- to file s in disk partitions with a file system `Ext2 / Ext3 / Ext4` - mount option file system secdel in / etc / fstab;

- for section s paging - tool command line swap-wiper with the name of the swap partition as an argument . For details, see “Administrator's Guide a ”, clause 8.1).

for solid-state drives (see Solid State Drives (SSD): application features) (in OS RavanOS CE and in OS CH RavanOS SE):

- to file s in disk partitions with a file system Ext4 - mount option file system discard in the / etc/fstab;
- for swap partitions - blkdiscard command:

  sudo blkdiscard /dev/sda5

The swap-wiper tool included in OS CH RavanOS SE (Smolensk 1.6, Smolensk 1.5, Leningrad 8.1) is designed to erase data contained in swap disk partitions when the system is turned off.

This tool is available in the / usr / lib / parsec / bin / swap-wiper file and is designed to automatically erase paging areas when the computer is turned off. The swap-wiper tool is blocked by default in its settings file.

To use the tool, in the /etc/parsec/swap_wiper.conf file, set the value of the ENABLED = Y attribute, after which the tool will be automatically applied when the system is turned off.

ou can set the ENABLED = Y attribute using a text editor, or fly-admin-smc, a standard graphical security management tool, accessible via the graphical menu:

Info

`"Start" - "Control Panel" - "Security" - "Security Policy" - "Security Settings" - "Memory Cleanup Policy", "Purge Swap Partitions")`

Warning

Erasing of information on solid-state drives is not guaranteed, see details. Solid-state drives (SSD): application features

To ensure reliable protection against unauthorized data access in the paging areas, it is recommended to use the automatic protection of data from disks and paging areas provided in RavanOS SE / CE.

This solution somewhat increases the load on the processor, but significantly increases the safety of confidential information both when using solid-state drives , and when a computer is suddenly turned off or when the computer is in sleep mode.

> **Attention**
>
> As an additional security measure, it is possible to recommend the use of backup power supply systems, which provide automatic shutdown of computers in case of power failures, including cleaning of pumping areas.

In extreme cases, if it is impossible or, based on the specific model of the offender, insufficient use of the protection measures described above, you should completely disable the paging mechanisms, for which you can either disable or unload the pages (value 0 of the kernel parameter vm.swappiness):

```
```sysctl -w vm.swappiness = 0```
```

either completely shutting down all swap areas (the -a or --all parameter of the swapoff command):

```
```swapoff -a```
```

followed by cleaning up the free swap areas.

Software Security Recommendations

When developing your own software

When developing your own software to protect memory containing unprotected sensitive data, you should use the `mlock ()` and `mlockall ()` system calls :

> **Info**
>
> `#include <sys / mman.h> int mlock (const void * addr , size_t len ); int munlock (const void * addr , size_t len ); int mlockall (int flags ); int munlockall (void);`

The `mlock ()` and `mlockall ()` calls block some or all of the virtual address space of the process in RAM , preventing pages from being paged into the paging area. The `munlock ()` and `munlockall ()` calls do the opposite, allowing the kernel memory manager to unload pages into the swap area if necessary . Application details see

```
  man mlock
```

When using third-party software

When using third-party programs, you can use the cgroups mechanism to reduce the likelihood of a process memory being dumped into the paging area, which allows you to assign a memory.swappiness group attribute with a zero value to a separate process (process group). This attribute is generally similar to the vm.swappiness kernel attribute described above, but its zero value does not guarantee complete unloading , since the kernel settings have a higher priority than the group settings, and the kernel can unload such processes in the event of a memory deficit.

Kernel software and paging security

The memory of the kernel (kernel modules) is not subject to unloading into the swap area, respectively, is not subject to risks associated with being in the swap area.

However, the privilege level of the kernel allows access to any system resources; accordingly, the kernel modules themselves are a potential threat to the confidentiality of the paging space.

Generally accepted security measures should be applied to kernel modules: use only signed kernel modules obtained from trusted sources, and if possible, control the source texts for unusual system calls. An additional measure of protection is also the use of protective data conversion.