Swap

Christian Külker

2023-05-25

Contents

Temporary memory (also known as swap) is not really needed by a Linux system. However, some Linux distributions refuse to install if it is not provided. In some cases, changing or adding swap space on the fly (when a system has no swap space) is an infrequent task, and sometimes the correct commands need to be remembered or looked up (in this document).

Linux provides two types of swap space. By default, most Linux installations use or create a swap partition, but it is also possible to use a specially configured file as a swap file.

1   Swap Games With Files

See how full the disk is

```
df -h
```

Prints the summary (if any) of swap spaces

```
swapon -s
```

See how many main memory and swap is available

```
free
free -m
cat /proc/swaps
top
```

See the swap in and out

```
vmstat
```

See the swap usage of single applications

```
smem -s swap
```

In case there is no swap and we want to create 2G of swap space

```
fallocate -l 2G /swapfile
```

See the result

```
ls -lh /swapfile
```

Swap space should not be readable by world

```
chmod 600 /swapfile
```

Create a swap signature

```
mkswap /swapfile
```

Use the swap file

```
swapon /swapfile
```

Show a summary

```
swapon -s
```

This should be visible in the memory overview

```
free -m
```

Disable the swap partition with the command that disables all swap space

```
swapoff -a
```

For some non Debian systems: To mount the swap file on boot, add an entry to `/etc/fstab`

```
echo "/swapfile none swap sw 0 0" >>/etc/fstab
```

For Debian systems: To mount the swap file on boot, add an entry to `/etc/fstab`

```
echo "/swapfile swap swap defaults 0 0" >>/etc/fstab
```

2   Swap Games with systemd

The `smem -s swap` command can list the swap space used by processes. Sometimes, if only `root` is logged in, you may see a line like this:

```
1  16940 USER      /lib/systemd/systemd --user      240     1376     2444
      8088
```

The **240** indicates that the process with PID **16940** is using swap for the user `USER`. Usually this means the user is logged in. In some cases this is not the case and the entry is hanging. For example, if the user logs in again using ssh, the ssh process will usually "inherit" the swappiness.

To understand which service is using this entry from `systemd`, remember the PID and run the following command:

```
systemd-cgls -u user.slice --no-pager
```

It will print something like:

```
1  Unit user.slice (/user.slice):├
2  user-0.slice│
3   ├─session-34535.scope│
4   │ ├─  619 /bin/login -p --│
5   │ └─28392 systemd-cgls -u user.slice --no-pager│
6   └─user@0.service│
7     └─init.scope│
8       └─16940 /lib/systemd/systemd --user└
9  user-1000.slice
10    └─user@1000.service
11      └─init.scope
12        └─11826 /lib/systemd/systemd --user
```

After executing a restart and `systemd-cgls`

```
systemctl restart systemd-user-sessions.service
systemd-cgls -u user.slice --no-pager
```

The entry gets shorter

```
1  Unit user.slice (/user.slice):└─
2  user-0.slice
3    ├─session-34535.scope
4    │ ├─  619 /bin/login -p --
5    │ └─31767 systemd-cgls -u user.slice --no-pager
6    └─user@0.service
7      └─init.scope
8        └─16940 /lib/systemd/systemd --user
```

And `smem -s swap` will not show the line for the user `USER` any more.

3   History

Version	Date	Notes
0.1.2	2023-05-25	Improve writing, spelling
0.1.1	2022-05-30	Change shell to bash, +history, +systemd –user
0.1.0	2022-05-27	Initial release

4   Disclaimer of Warranty

5   Limitation of Liability