

Limits setzen/überschreiben mit systemd

Wie kann ich Limits für Services setzen, die via systemd gestartet werden? Meine Einstellungen in `/etc/security/limits.conf` oder `/etc/security/limits.d/*.conf` werden ignoriert, da diese nur von `pam_limits.so` verwendet werden, was `systemd` nicht nutzt.

Um die Limit anzupassen muss das systemd unit angepasst werden, z.B. für MySQL:

```
$ systemctl edit mysql.service
# einfügen und speichern:
[Service]
LimitNOFILE=500000

# Service neu starten
$ systemctl restart mysql.service
```

Im `systemctl status` sieht man jetzt den Override

```
$ systemctl status mysql.service
● mysql.service - Percona Server
   Loaded: loaded (/lib/systemd/system/mysql.service; enabled; vendor preset: enabled)
   Drop-In: /etc/systemd/system/mysql.service.d
            └─override.conf
```

Limits für alle Prozesse überschreiben

```
mkdir -p /etc/systemd/system.conf.d/
cat >/etc/systemd/system.conf.d/10-filelimit.conf <<EOF
[Manager]

DefaultLimitNOFILE=500000
EOF
systemctl daemon-reload
## ggf. Reboot!
```

Folgende Limits können überschrieben werden:

Directive	ulimit equivalent	Unit
<code>ulimit -u</code>	<code>ulimit -u</code>	Number of processes
<code>ulimit -m</code>	<code>ulimit -m</code>	Maximum memory size
<code>ulimit -v</code>	<code>ulimit -v</code>	Maximum virtual memory size
<code>ulimit -f</code>	<code>ulimit -f</code>	Maximum file size
<code>ulimit -s</code>	<code>ulimit -s</code>	Maximum stack size
<code>ulimit -t</code>	<code>ulimit -t</code>	Maximum time limit
<code>ulimit -l</code>	<code>ulimit -l</code>	Maximum locked memory size
<code>ulimit -p</code>	<code>ulimit -p</code>	Maximum pipe size
<code>ulimit -n</code>	<code>ulimit -n</code>	Maximum number of open files
<code>ulimit -c</code>	<code>ulimit -c</code>	Maximum core file size
<code>ulimit -x</code>	<code>ulimit -x</code>	Maximum number of shared library pages
<code>ulimit -r</code>	<code>ulimit -r</code>	Maximum number of resident pages
<code>ulimit -w</code>	<code>ulimit -w</code>	Maximum number of write pages
<code>ulimit -d</code>	<code>ulimit -d</code>	Maximum number of data pages
<code>ulimit -b</code>	<code>ulimit -b</code>	Maximum number of buffer pages
<code>ulimit -k</code>	<code>ulimit -k</code>	Maximum number of kernel pages
<code>ulimit -g</code>	<code>ulimit -g</code>	Maximum number of global pages
<code>ulimit -e</code>	<code>ulimit -e</code>	Maximum number of external pages
<code>ulimit -o</code>	<code>ulimit -o</code>	Maximum number of object pages
<code>ulimit -i</code>	<code>ulimit -i</code>	Maximum number of instruction pages
<code>ulimit -a</code>	<code>ulimit -a</code>	Maximum number of all pages

Notes

LimitCPU=	ulimit -t	Seconds	-
-----------	-----------	---------	---

LimitFSIZE=	ulimit -f	Bytes	-
-------------	-----------	-------	---

LimitDATA=	ulimit -d	Bytes	Don't use.
------------	-----------	-------	------------

This limits the allowed address range, not memory use! Defaults to unlimited and should not be lowered. To limit memory use, see MemoryMax= in systemd.resource-control(5).

LimitSTACK=	ulimit -s	Bytes	-
-------------	-----------	-------	---

LimitCORE=	ulimit -c	Bytes	-
------------	-----------	-------	---

LimitRSS=	ulimit -m	Bytes	Don't use. No effect on Linux.
-----------	-----------	-------	--------------------------------

LimitNOFILE=	ulimit -n	Number of File Descriptors	Don't use. Be careful when raising the soft limit above 1024,
--------------	-----------	----------------------------	---

			since
select(2) cannot function with file descriptors above			1023 on
Linux. Nowadays, the hard limit defaults to 524288, a			very high
value compared to historical defaults. Typically			applications
should increase their soft limit to the hard			limit on
their own, if they are OK with working with file			descriptors
above 1023, i.e. do not use select(2). Note that			file
descriptors are nowadays accounted like any other form of			memory, thus
there should not be any need to lower the hard			limit. Use
MemoryMax= to control overall service memory use,			including
file descriptor memory.			

	LimitAS=	ulimit -v	Bytes	Don't use.
This limits the allowed address range, not memory				use! Defaults
to unlimited and should not be lowered. To limit				memory use,
see MemoryMax= in systemd.resource-control(5).				

	LimitNPROC=	ulimit -u	Number of Processes	This limit is
enforced based on the number of processes				belonging to
the user. Typically it's better to track				processes per
service, i.e. use TasksMax=, see				
systemd.resource-control(5).				

LimitMEMLOCK=	ulimit -l	Bytes	-
LimitLOCKS=	ulimit -x	Number of Locks	-
LimitSIGPENDING=	ulimit -i	Number of Queued Signals	-
LimitMSGQUEUE=	ulimit -q	Bytes	-
LimitNICE=	ulimit -e	Nice Level	-
LimitRTPRIO=	ulimit -r	Realtime Priority	-
LimitRTTIME=	ulimit -R	Microseconds	-

Weitere Infos in den manpages

```
man 5 systemd.exec
```

```
man 5 systemd.resource-control
```

Revision #2

Created 17 October 2023 07:36:46 by magenbrot

Updated 17 October 2023 07:53:53 by magenbrot