

# Linux-Out-of-Memory-Killer

Aus aktuellem Anlass.

## Was ist das ?

Der „OOM Killer“ oder „Out of Memory Killer“ ist ein Prozess, den der Linux-Kernel einsetzt, wenn das System kritisch wenig Arbeitsspeicher hat. Diese Situation tritt auf, weil Prozesse auf dem Server viel Speicher verbrauchen und das System mehr Speicher für seine eigenen Prozesse und zur Zuweisung an andere Prozesse benötigt. Wenn ein Prozess startet, fordert er einen Speicherblock vom Kernel an. Bei dieser ersten Anfrage handelt es sich in der Regel um eine große Anfrage, die der Prozess nicht sofort oder überhaupt nicht vollständig nutzen wird. Der Kernel ist sich der Tendenz von Prozessen bewusst, redundanten Speicher anzufordern, und weist dem Systemspeicher zu viel zu. Das heißt, wenn das System beispielsweise über 8 GB RAM verfügt, kann der Kernel Prozessen 8,5 GB zuweisen. Dadurch wird die Nutzung des Systemspeichers maximiert, indem sichergestellt wird, dass der den Prozessen zugewiesene Speicher aktiv genutzt wird.

Normalerweise stellt diese Situation kein Problem dar. Wenn jedoch genügend Prozesse beginnen, alle angeforderten Speicherblöcke zu nutzen, ist nicht genügend physischer Speicher vorhanden, um sie alle zu unterstützen. Das bedeutet, dass die laufenden Prozesse mehr Speicher benötigen, als physisch verfügbar ist. Diese Situation ist kritisch und muss sofort gelöst werden.

Die Lösung, die der Linux-Kernel verwendet, besteht darin, den OOM Killer aufzurufen, um alle laufenden Prozesse zu überprüfen und einen oder mehrere davon zu beenden, um Systemspeicher freizugeben und das System am Laufen zu halten.

## Prozessauswahl

Immer wenn ein Speichermangel auftritt, wird die Funktion `out_of_memory()` aufgerufen. Darin wird die Funktion `select_bad_process()` verwendet, die einen Score von der Funktion `badness()` erhält. Der „schlechteste“ Prozess ist derjenige, der geopfert wird. Es gibt einige Regeln, denen die Funktion `badness()` für die Auswahl des Prozesses folgt.

1. Der Kernel muss eine Mindestmenge an Speicher für sich selbst beschaffen
2. Versuchen Sie, eine große Menge Speicher zurückzugewinnen
3. Beenden Sie einen Prozess nicht mit wenig Speicher
4. Versuchen Sie, die Mindestanzahl an Prozessen abzurechnen
5. Einige sorgfältige Algorithmen, die die Opferpriorität für Prozesse erhöhen, die der Benutzer beenden möchte

Nach all diesen Checklisten prüft der OOM-Killer den Score (`oom_score`). OOM legt den „`oom_score`“ für jeden Prozess fest und multipliziert diesen Wert dann mit der Speichernutzung. Bei Prozessen mit größeren Werten besteht eine hohe Wahrscheinlichkeit, dass sie vom OOM-Killer beendet werden. Die Prozesse, die dem privilegierten Benutzer zugeordnet sind, haben einen niedrigeren Bewertungswert und eine geringere Wahrscheinlichkeit, von OOM beendet zu werden.

```
postgres=# SELECT pg_backend_pid();
pg_backend_pid
-----
3813
(1 Zeile)
```

**Die Postgres-Prozess-ID ist 3813, daher können Sie in einer anderen Shell den Score-Wert mithilfe dieses oom\_score-Kernelparameters abrufen:**

```
$ sudo cat /proc/3813/oom_score
2
```

Wenn Sie wirklich möchten, dass Ihr Prozess nicht durch OOM-Killer getötet wird, gibt es einen weiteren Kernel-Parameter `oom_score_adj`. Sie können einen großen negativen Wert hinzufügen, um die Wahrscheinlichkeit zu verringern, dass Ihr Prozess abstürzt.

```
sudo echo -100 > /proc/3813/oom_score_adj
```

oder `oomprotect` des `rcctl`-Befehls kann verwendet werden, um dies festzulegen.

```
rcctl set <i>Dienstname</i> oomprotect -1000
```

## Einen Prozess beenden

Wenn ein oder mehrere Prozesse ausgewählt sind, ruft OOM-Killer die Funktion `oom_kill_task()` auf. Diese Funktion ist dafür verantwortlich, das Beendigungs-/Kill-Signal an den Prozess zu senden. Im Falle von nicht genügend Speicher `oom_kill()` rufen Sie diese Funktion auf, damit sie das SIGKILL-Signal an den Prozess senden kann. Es wird eine Kernel-Protokollmeldung generiert.

Nicht genügend Speicher: Prozess [pid] [name] abgebrochen.

## So steuern Sie OOM-Killer

Linux bietet eine Möglichkeit, den OOM-Killer zu aktivieren und zu deaktivieren, es wird jedoch nicht empfohlen, den OOM-Killer zu deaktivieren. Der Kernel-Parameter `vm.oom-kill` wird zum Aktivieren und Deaktivieren des OOM-Killers verwendet. Wenn Sie die OOM-Killer-Laufzeit aktivieren möchten, verwenden Sie den Befehl `sysctl`, um dies zu aktivieren.

```
sudo -s sysctl -w vm.oom-kill = 1
```

**Um den OOM-Killer zu deaktivieren, verwenden Sie denselben Befehl mit dem Wert 0:**

```
sudo -s sysctl -w vm.oom-kill = 0
```

Dieser Befehl stellt dies nicht dauerhaft ein und wird durch einen Neustart des Computers zurückgesetzt. Um es dauerhaft festzulegen, fügen Sie diese Zeile in die Datei `/etc/sysctl.conf` ein:

```
echo vm.oom-kill = 1 >>/etc/sysctl.conf
```

Die andere Möglichkeit zum Aktivieren oder Deaktivieren besteht darin, die Variable `panic_on_oom` zu schreiben. Sie können den Wert jederzeit in `/proc` überprüfen.

```
# cat /proc/sys/vm/panic_on_oom  
0
```

Wenn Sie den Wert auf 0 setzen, bedeutet dies, dass der Kernel nicht in Panik gerät, wenn ein Fehler wegen unzureichendem Arbeitsspeicher auftritt.

```
$ echo 0 > /proc/sys/vm/panic_on_oom
```

Wenn Sie diesen Wert auf 1 setzen, bedeutet dies, dass der Kernel bei einem Fehler wegen nicht genügend Arbeitsspeicher in Panik gerät.

```
echo 1 > /proc/sys/vm/panic_on_oom
```

Es gibt neben der Aktivierung und Deaktivierung noch einige weitere Einstellungen für den OOM-Killer. Da wir bereits erwähnt haben, dass Linux den Speicher durch die Zuweisung von Prozessen zu stark beanspruchen kann, kann dieses Verhalten durch die Linux-Kernel-Einstellung gesteuert werden. Der `vm.overcommit_memory` wird variabel verwendet, um dieses Verhalten zu steuern.

Der Variablenspeicher `vm_overcommit_memory` kann mit den folgenden Einstellungen gesteuert werden:

0: Setzen Sie die Variable auf 0, wobei der Kernel entscheidet, ob ein Overcommit durchgeführt wird oder nicht. Dies ist der Standardwert für die meisten Linux-Versionen.

1: Wenn Sie die Variable auf 1 setzen, bedeutet dies, dass der Kernel immer überlastet ist. Dies ist eine riskante Einstellung, da der Kernel den Speicher immer für Prozesse überbelegt. Dies kann dazu führen, dass dem Kernel nicht mehr genügend Speicher zur Verfügung steht, da die Wahrscheinlichkeit groß ist, dass Prozesse am Ende den vom Kernel zugewiesenen Speicher nutzen.

2: Wenn Sie die Variable auf 2 setzen, bedeutet dies, dass der Kernel keinen Speicher über das `overcommit_ratio` hinaus überbelegen soll. Dieses `overcommit_ratio` ist eine weitere Kernel-Einstellung, mit der Sie den Prozentsatz des Speichers angeben, den der Kernel überbelegen kann. Wenn kein Platz für eine Überbelegung vorhanden ist, schlägt die Speicherzuweisungsfunktion fehl und die Überbelegung wird verweigert. Dies ist die sicherste Option und der empfohlene Wert für PostgreSQL.

Die zweite Sache, die den OOM-Killer beeinflussen kann, ist das Verhalten von swappiness. Dieses Verhalten kann durch die Variable `cat /proc/sys/vm/swappiness` gesteuert werden. Diese Werte geben die Kernel-Einstellung für die Handhabung des Seitenaustauschs an. Je größer der Wert, desto geringer ist die Wahrscheinlichkeit, dass OOM den Prozess abbricht, aber es wirkt sich aufgrund der E/A auf die Datenbankeffizienz aus. Ein kleinerer Wert für die Variable, die die Swappiness steuert, bedeutet, dass die Wahrscheinlichkeit, dass OOM-Killer eingreift, höher ist, aber es verbessert auch die Datenbankleistung. Der Standardwert ist 60, aber wenn Ihre gesamte Datenbank in den Speicher passt, wird empfohlen, diesen Wert auf 1 zu setzen.

# Warum wird Apache/MySQL/Postgres immer getötet?

Die oben aufgeführten Kriterien bedeuten, dass der OOM Killer bei der Auswahl eines Prozesses zum Beenden einen Prozess auswählt, der viel Speicher beansprucht und über viele untergeordnete Prozesse verfügt, die keine Systemprozesse sind. Eine Anwendung wie Apache, MySQL, Nginx, Clam (ClamAV) oder ein Mailserver sind ein guter Kandidat. Da diese Situation jedoch normalerweise auf ausgelasteten Webservern auftritt, sind Apache oder MySQL die größten systeminternen Prozesse im Arbeitsspeicher und werden folglich abgebrochen.

Denken Sie daran, dass der Webserver oder der DB-Server zwar für Sie sehr wichtig sind, die Situation jedoch kritisch ist, wenn der Kernel den OOM-Killer aufruft. Wenn durch das Beenden eines Prozesses kein Speicher freigegeben wird, stürzt der Server kurz darauf ab. Eine Fortsetzung des Normalbetriebs ist zu diesem Zeitpunkt nicht mehr möglich.

## Häufige Ursache

Eine der häufigsten Ursachen dafür, dass Apache/Nginx/MySQL durch den OOM Process Killer getötet wird, besteht darin, dass die Website eine große Menge an Datenverkehr empfängt. Hierbei kann es sich um echten Datenverkehr von einer neuen Werbeaktion, einer Medienaufmerksamkeit oder ähnlichem handeln ein Bot, der die Website crawlt, oder in manchen Fällen kann es sich um Botnetze handeln, die versuchen, Ihre Website mit Brute-Force anzugreifen. Die Überprüfung der Apache-/Nginx-Protokolle ist ein guter Ausgangspunkt, um festzustellen, ob dies der Fall ist.

## Zusammenfassung

Der Name Killer (OOM-Killer) muss Sie nicht verwirren. Der Mörder ist nicht immer schädlich; Es ist ein Retter für Ihr System. Es tötet den schlimmsten Prozess ab und bewahrt Ihr System vor einem Absturz. Um zu vermeiden, dass OOM-Killer zum Beenden von PostgreSQL verwendet werden muss, wird empfohlen, den `vm.overcommit_memory`-Wert auf 2 zu setzen. Dadurch wird der OOM-Killer nicht zu 100 % vermieden, aber die Wahrscheinlichkeit verringert, dass der PostgreSQL-Prozess beendet wird.

From:

<https://www.cooltux.net/> - TuxNet DokuWiki

Permanent link:

<https://www.cooltux.net/doku.php?id=blog:linux-out-of-memory-killer>

Last update: **2024/04/03 05:27**

