

# Содержание

**OOM killer** ..... 3



# OOM killer

Любой запущенный процесс получает собственный выделенный участок памяти. Но этот участок не отображается на доступные физические адреса, а использует **виртуальное адресное пространство**, которое является дополнительным слоем абстракции между процессом и физической памятью. Для сопоставления адресов между виртуальной и физической памятью используются специальные структуры - **таблицы страниц**.

Что это дает? Во-первых, изоляцию процессов друг от друга, каждый из них работает в собственном виртуальном адресном пространстве и не имеет доступа к памяти других процессов. Во-вторых, позволяет системе гибко управлять физической памятью, например, для неактивного процесса часть страниц может быть перенесено из оперативной памяти в пространства подкачки, но это никак не повлияет на работу самого процесса, его адресное пространство не изменится, поменяется лишь трансляция в таблице страниц. И, наконец, это позволяет экономить физическую память, так если несколько процессов используют одни и те же данные, то система может транслировать адреса виртуальных пространств на один и тот же участок физической памяти.

Поэтому современные Linux-системы выделяют память процессам с превышением имеющейся в наличии физической памяти (RAM + Swap), по умолчанию используя эвристические механизмы. Но случается, что процессы съедают всю доступную физическую память и вот тут системе надо выбрать что-то одно. А выбор, скажем честно, между плохим или очень плохим. Можно убить один из процессов и продолжить работать дальше, а можно вызвать Kernel panic. Понятно, что последнее - это совсем ни в какие ворота, поэтому на сцену выходит **OOM Killer**, это поведение используется в современных Linux системах по умолчанию.

На самом деле OOM Killer использует достаточно сложный анализ и присваивает каждому процессу **очки негодности (badness)** и при исчерпании памяти будет убит не кто-то попало, а самый негодный процесс. Очки негодности могут принимать значения от -1000 до 1000, чем выше значение, тем более высока вероятность что OOM Killer убьет процесс. Процесс с -1000 очков никогда не будет убит OOM Killer.

Как вычисляются очки негодности? За основу берется процент физической памяти используемый процессом и умножается на 10, таким образом получаем базовое значение. Нетрудно заметить, что 1000 очков - это 100% использования памяти. Затем к данному значению начинают применяться различные модификаторы:

- Прибавляется половина очков всех дочерних процессов, имеющих собственную виртуальную память.
- Если приоритет процесса больше нуля очки умножаются на два. Приоритет может иметь значение от -20 (наивысший приоритет) до 20 (самый низкий).
- Очки делятся на коэффициент, связанный с процессорным временем, чем более активен процесс и чем больше процессорного времени он использует, тем больше будет этот коэффициент.
- Очки делятся на коэффициент, связанный с временем жизни процесса, чем больше времени прошло с момента запуска, тем выше будет это значение.
- Для процессов, запущенных от имени root, обслуживающих систему или процессов ввода-вывода очки делятся на 4.
- Для процесса, при выделении памяти которому произошла ошибка out of memory и процессам имеющим с ним общую память, очки делятся на 8.

- Очки умножаются на  $2^{\text{oom\_adj}}$ , где `oom_adj` - специальный коэффициент, имеющий значения от -17 до 15, при -17 процесс никогда не будет убит OOM Killer.

Если внимательно изучить эти критерии, то становится ясно, что OOM Killer убьет самый «толстый» процесс, который наименее активен в системе и имеет самое короткое время жизни

Количество очков находится в `/proc/PID/oom_score`. Соответственно, присвоить значение вручную можно так: `echo -17 > /proc/PID/oom_adj`. Заранее определить коэффициент `oom_adj` для еще не запущенного процесса можно через систему инициализации.

From:

<https://wiki.radi0.cc/> - radi0wiki

Permanent link:

[https://wiki.radi0.cc/glossary:gnu\\_linux:oom\\_killer](https://wiki.radi0.cc/glossary:gnu_linux:oom_killer)

Last update: **2026/03/03 11:56**

