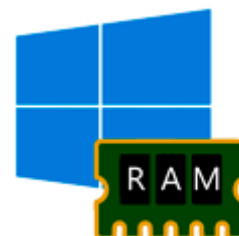


# Некоторые особенности использования оперативной памяти в виртуальных машинах Windows

 [interface31.ru/tech\\_it/2022/07/nekotorye-osobennosti-ispol-zovaniya-operativnoy-pamyati-v-virtual-nyh-mashinah-windows.html](https://interface31.ru/tech_it/2022/07/nekotorye-osobennosti-ispol-zovaniya-operativnoy-pamyati-v-virtual-nyh-mashinah-windows.html)

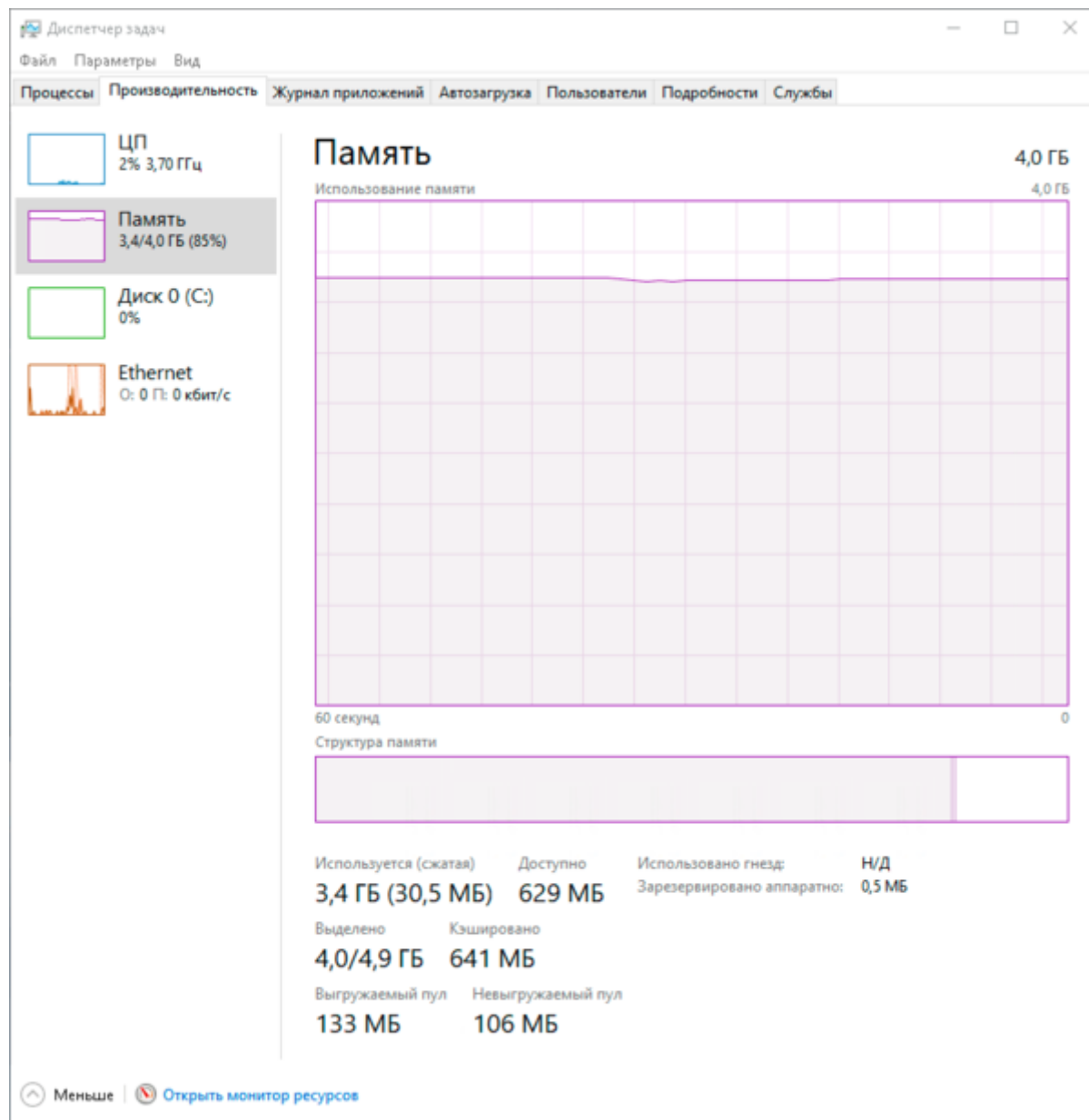
Виртуальные машины прочно вошли в нашу жизнь и принесли с собой новые подходы к управлению ресурсами, а также новые потенциальные проблемы. Это не хорошо и не плохо, так и должно быть, сильные и слабые стороны есть у любого решения, и задача хорошего специалиста знать о них и играть от сильных сторон. Но тем не менее не все обладают нужным набором знаний и даже не представляют куда смотреть, если ресурсы, выделенные виртуальной машине, ведут себя непредсказуемо, в первую очередь это касается оперативной памяти.



## Куда пропала моя память?

Разберем один недавний случай, с которым к нам обратился один коллега. После перевода всей инфраструктуры на виртуальные машины он задумался отказаться от стационарного рабочего места в пользу нескольких виртуальных машин, каждая из которых предполагалась к управлению своей частью инфраструктуры.

И вроде бы все шло хорошо, пока он не начал выполнять на этой виртуалке некоторые тяжелые задачи, после чего был выявлен недостаток свободной оперативной памяти. И это при том, что на машине стоял практически чистый Windows с набором инструментов администрирования, а в автозагрузку добавился только Яндекс.Диск. Так куда же делась оперативная память?



Обычно в таких случаях начинают искать процессы с непомерным потреблением, но в нашем случае и **диспетчер задач** и **монитор ресурсов** никакой аномалии не видели, только вот согласно их данным оперативной памяти должно быть занято значительно меньше, всего-то около 1,5 ГБ. Ладно, перейдем к тяжелой артиллерии, следующая команда PowerShell покажет число запущенных в системе процессов и общий объем занимаемой ими памяти:

```
Get-Process | measure PM -sum
```

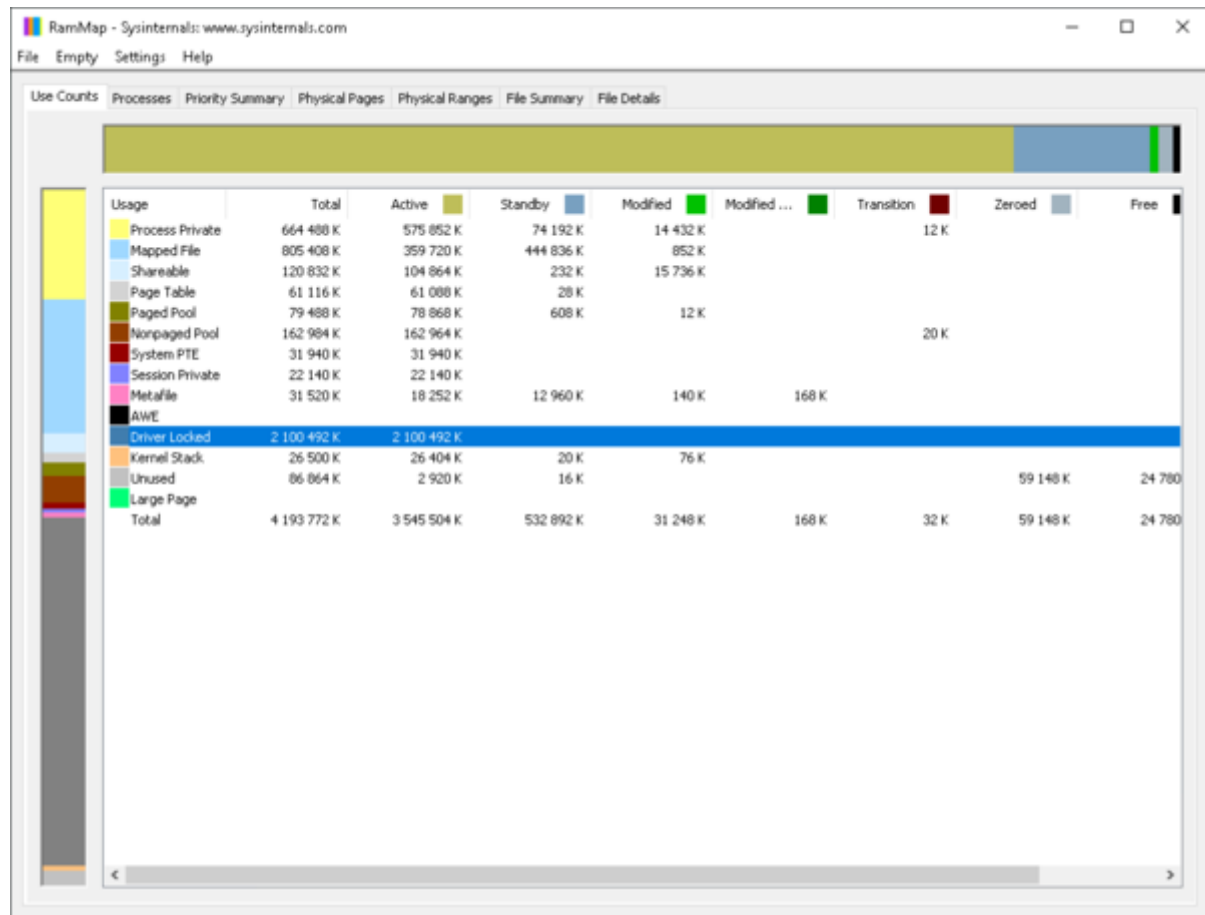
Если вы не хотите пересчитывать память из байтов в мега и гигабайты, то можно воспользоваться следующим "заклинанием", которое просто покажет объем занятой всеми процессами памяти в МБ:

```
$mem = 0; Get-Process | %{$mem += $_.pm}; "{0:N2}MB " -f ($mem/1mb)
```

Но и тут мы ничего нового не узнали, запущенные процессы занимают всего около 1,5 ГБ:

```
PowerShell 7.3.0-preview.2
PS C:\Windows\System32> $mem = 0; Get-Process | %{$mem += $_.pm}; "{0:N2}MB " -f ($mem/1mb)
1 531,43MB
PS C:\Windows\System32> _
```

А где же тогда остальная память? Самое время поверить в мистику и всякие теории заговора о скрытой жизни под капотом Windows, но не будем спешить с выводами, а возьмем в руки утилиту RAMMap от Sysinternals. После чего начинает приходить первое понимание:



Обратить внимание следует на строку **Driver Locked** - это память заблокированная под собственные нужды драйвером операционной системы, в нашем случае это как раз и есть те "пропавшие" 2 ГБ. Почему так происходит? Давайте разбираться.

## Динамическая память

Практически все современные гипервизоры предоставляют возможность **динамического выделения памяти** виртуальным машинам, это **Memory Ballooning** в KVM (Proxmox и VMWare) или **Dynamic Memory** в Hyper-V. Но не все имеют четкое понимание того, как работает эта технология. На первый взгляд все просто, мы запустили виртуалку с некоторым минимальным объемом памяти, но если ей в процессе работы понадобится больше - то гипервизор выделит ей в пределах указанных лимитов.

Да, все это так, но в условиях конечного количества ресурса - оперативной памяти, чтобы кому-то что-то выделить, надо сначала это у кого-то забрать. А у кого? У того, кому сейчас это не нужно. Но если мы можем на лету увеличить количество оперативной памяти в виртуальной машине, то уменьшить ее без перезагрузки мы не можем и тут вступает в действие механизм принудительного захвата памяти.

Специальный драйвер, отвечающий за динамическую память как бы надует в оперативке фиктивный шар (*balloon*), который будет считаться занятым и, следовательно, недоступным системе. Отсюда и пошло название этого механизма - Ballooning. А так как системе этот объем памяти недоступен, то гипервизор может отдать его другой виртуальной машине, либо использовать на собственные нужды.

При этом работа этого механизма довольно неочевидна, и система динамической памяти может заблокировать некоторый объем даже при наличии достаточно количества памяти на гипервизоре, просто потому что она не используется виртуалкой, и чтобы иметь необходимый резерв.

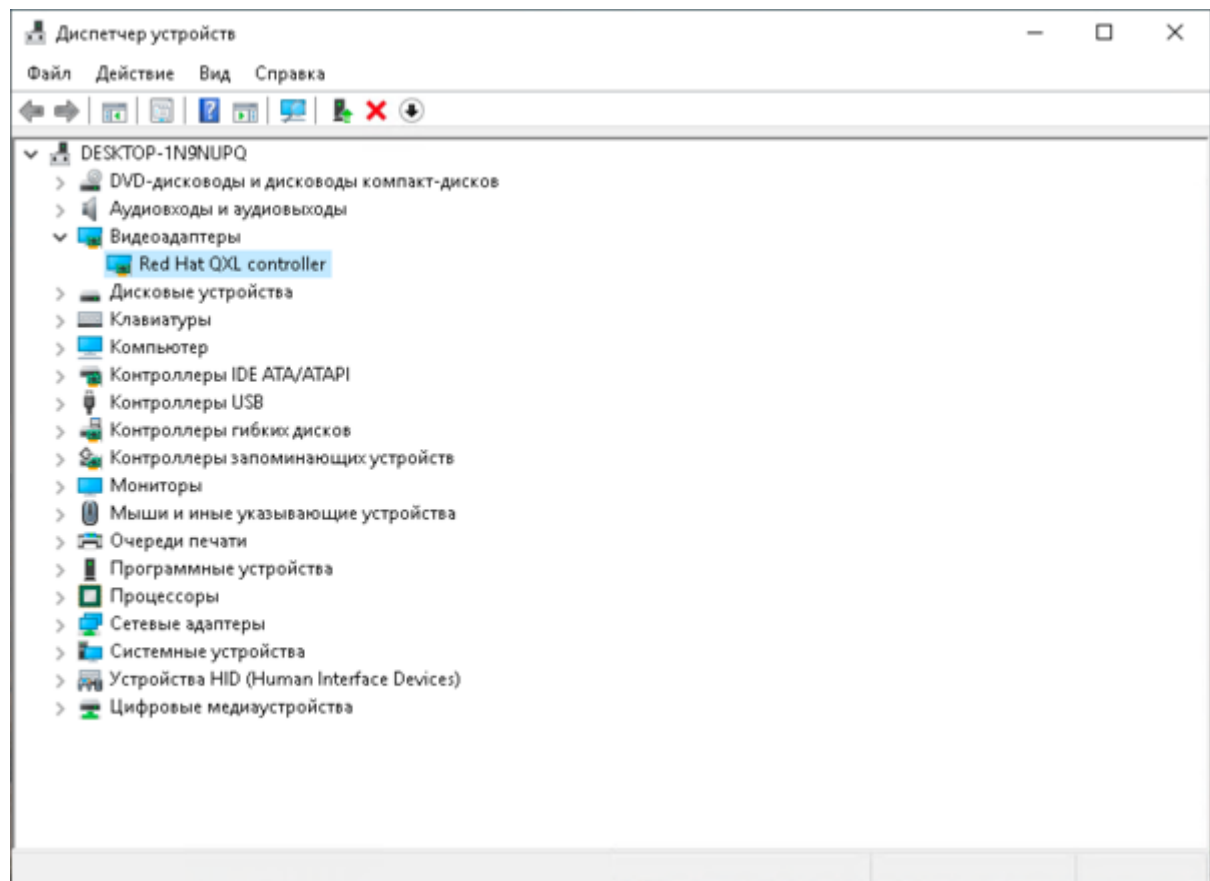
Поэтому, если у вас включена динамическая память и какое-то ее количество оказывается в состоянии Driver Locked - это нормально, однако при необходимости и при наличии на хосте свободной оперативной памяти она должна выделяться для текущей виртуальной машины.

## Гостевые устройства

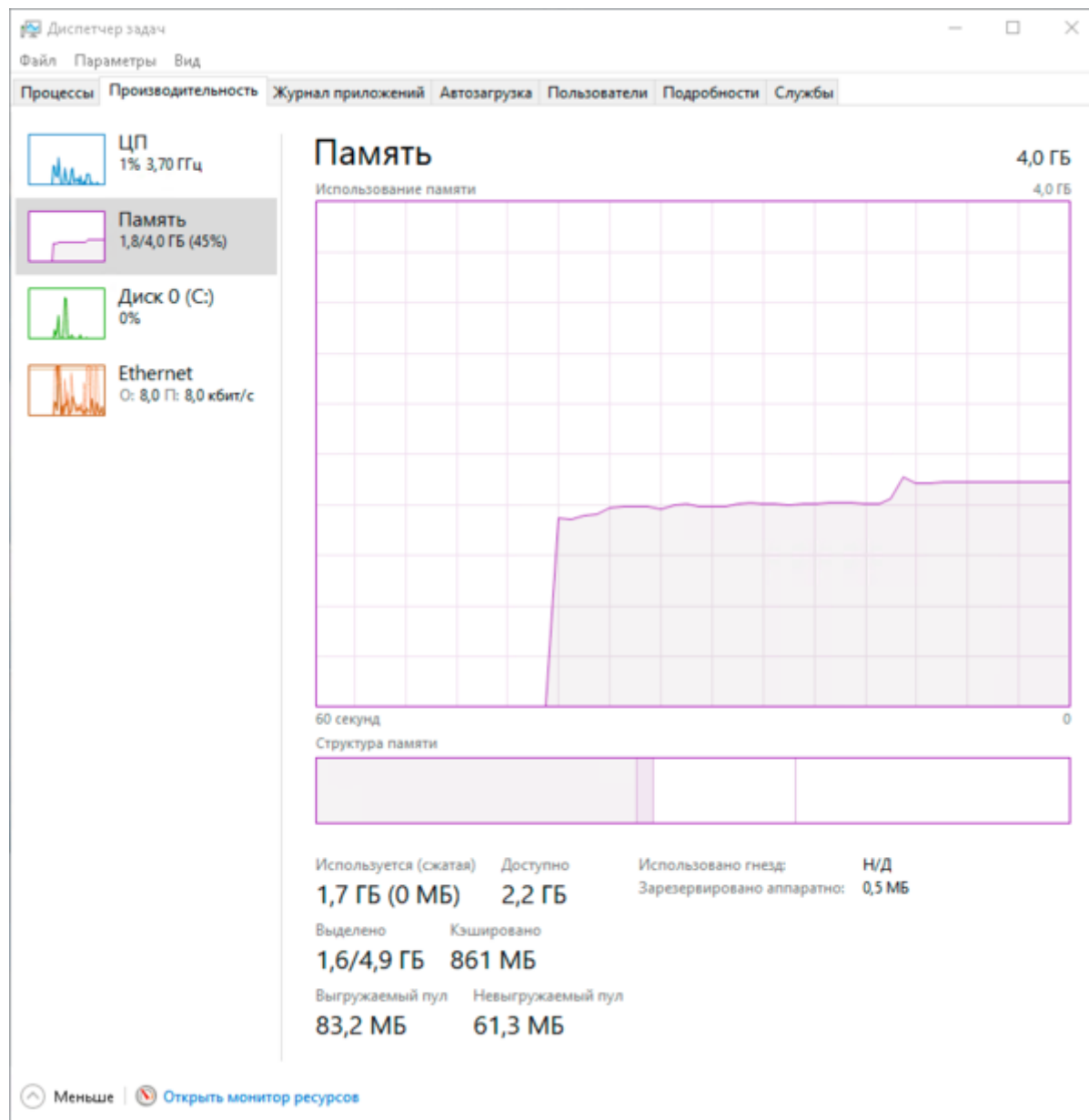
---

Однако в нашем случае память не выделялась и вообще Ballooning был выключен. Поэтому будем искать дальше. Следующие на очереди - гостевые устройства, которые эмулируются гипервизором и используются виртуальной машиной вместо реального оборудования. Обычно для их работы следует установить специальные гостевые драйвера, но некоторые виртуальные устройства могут работать и со стандартными драйверами. Мы сначала хотели написать - нормально работать, но отказались от такой формулировки, потому что данная проблема лежала именно в таком "нормально" работающем устройстве со стандартным драйвером.

Виновником всего происходящего оказался **Базовый видеоадаптер Microsoft**, выяснено это было эмпирическим путем, так как на все остальное, кроме него, гостевые драйвера стояли. Поэтому мы вручную установили для видеоадаптера драйвер из состава VirtIO (гипервизор Proxmox).



Перезагружаемся и видим вполне реальную картину потребления памяти:



И никакой мистики, тайн, интриг и срывов покровов - просто один неправильно работающий драйвер.

## Выводы

Работая с виртуальными машинами, следует понимать, что многие подходы, привычные по обычному железу, здесь не работают, хотя и на обычном железе стандартный драйвер способен принести ряд проблем. В любом случае обязательно устанавливайте и обновляйте гостевые драйвера, если даже особой необходимости в этом не видно. Как показывает практика - гостевые драйвера не просто так придуманы.

Второй момент - это сугубо виртуальные технологии, такие как динамическая память, применяя их вы должны понимать для чего вы это делаете и иметь представление, хотя бы в общих чертах, о механизмах их работы. В подавляющем большинстве случаев, при использовании данных механизмов, инструменты ОС будут показывать "неверное" количество свободной памяти.

Также следует понимать применимость данных технологий, динамическая память хорошо помогает в кратковременных ситуациях, требующих повышенного расхода оперативной памяти, если же памяти для всех виртуальных машин действительно недостаточно, то никакая динамическая память вам не поможет, сколько не говори халва - слаще во рту не станет.

Поэтому применяйте технологии виртуализации благоразумно и постарайтесь хотя бы в общих чертах понимать механизмы их действия, чтобы правильно интерпретировать поведение системы и отличать нормальные ситуации, от действительных проблем.

- **Категории:**

- [Windows 10/11](#),
- [Windows Server](#),
- [Виртуализация](#),
- [Системному администратору](#).

- **Теги:**

- [Hyper-V](#),
  - [Proxmox](#),
  - [Windows](#),
  - [Windows Server](#),
  - [Виртуализация](#),
  - [Производительность](#)
-