# [进入正题](#_sudo_dd_if=/dev/zero)



linux创建交换分区设置

[zentilia/Shutterstock.com](https://www.shutterstock.com/image-photo/closeup-inside-harddrives-134428790) [zentilia / Shutterstock.com](https://www.shutterstock.com/image-photo/closeup-inside-harddrives-134428790)

Add swap space to a Linux computer, or increase the swap space that’s already present, without messing about with partitions. We show you the easy way to tailor your swap space.

将交换空间添加到Linux计算机，或增加已经存在的交换空间，而无需弄乱分区。 我们向您展示了定制交换空间的简便方法。

**交换文件与交换分区 (Swap Files vs. Swap Partitions)**

There are several scenarios where you might want to increase existing or add new swap space to your Linux computer.

在几种情况下，您可能希望增加现有空间或向Linux计算机添加新的交换空间。

* Perhaps your swap space is frequently running at maximum or close to maximum.

也许交换空间经常以最大或接近最大的速度运行。

* It’s easy to click the wrong option during the installation process and to decline adding swap to your system inadvertently.

很容易在安装过程中单击错误的选项，并拒绝在不经意间将交换添加到系统中。

* Maybe you previously decided that you had so much [random access memory](https://en.wikipedia.org/wiki/Random-access_memory) (RAM) you didn’t need any swap, and you’ve changed your mind.

也许您先前认为您有这么多的[随机存取存储器](https://en.wikipedia.org/wiki/Random-access_memory) (RAM)，不需要进行任何交换，但是您改变了主意。

* Sometimes you inherit the administration of a system that has no swap, for reasons you’ll never be able to discover.

有时，由于永远无法发现的原因，您继承了没有交换的系统的管理。

The simple solution to all of these is to add a [swap file to your computer](https://en.wikipedia.org/wiki/Paging#Swap_files_and_partitions). This is a special file, pre-allocated and reserved for use as swap space. A swap file will work in tandem with any existing swap you may have, whether that is a swap file or a swap partition.

所有这些的简单解决方案是将[交换文件](https://en.wikipedia.org/wiki/Paging" \l "Swap_files_and_partitions)添加[到您的计算机](https://en.wikipedia.org/wiki/Paging#Swap_files_and_partitions) 。 这是一个特殊文件，已预先分配并保留用作交换空间。 交换文件将与您可能拥有的任何现有交换(包括交换文件或交换分区)协同工作。

At one time, there was a performance hit for using a swap file compared to a swap partition. That’s no longer the case with improvements in the performance of mechanical (spinning) hard drives and more efficiency in the swap functions within the Linux operating system. In fact, some Linux distributions now default to creating swap files rather than swap partitions.

一次，与交换分区相比，使用交换文件的性能受到了影响。 机械(旋转)硬盘性能的提高和Linux操作系统中交换功能的更高效率不再是这种情况。 实际上，某些Linux发行版现在默认创建交换文件，而不是交换分区。

Swap isn’t just used as a means to free up RAM when you’re running low on memory. It’s an important part of a well functioning system. Without any swap, sane memory management becomes very difficult for the kernel to achieve. Let’s look at the easy way to add some swap space.

交换不仅仅用于在内存不足时释放RAM。 这是运作良好的系统的重要组成部分。 如果不进行任何交换，则对于内核而言，健全的内存管理将变得非常困难。 让我们看一下添加一些交换空间的简单方法。

**在我们深入之前：Btrfs和SSD (Before We Dive In: Btrfs and SSDs)**

There are two points we’d like to discuss quickly.

我们想快速讨论两点。

The [Btrfs file system](https://btrfs.wiki.kernel.org/index.php/Main_Page) has certain caveats about swap files. At one time, there was a conflict between the [copy-on-write](https://en.wikipedia.org/wiki/Copy-on-write) nature of Btrfs, which wanted to operate in one way and swap files that needed to operate in another. Some functions that swap files depend on were not implemented, and some assumptions that were made about block numbering within the swap files did not hold true with Btrfs. So swap files were not supported.

[Btrfs文件系统](https://btrfs.wiki.kernel.org/index.php/Main_Page)对交换文件有一些警告。 一次，Btrfs [的写时复制](https://en.wikipedia.org/wiki/Copy-on-write)特性(想要以一种方式进行操作)与交换需要以另一种方式进行操作的文件之间存在冲突。 交换文件所依赖的某些功能尚未实现，并且有关交换文件中块编号的某些假设在Btrfs中不成立。 因此不支持交换文件。

Since kernel 5.0, you [can have swap files in Btrfs file systems](https://btrfs.wiki.kernel.org/index.php/FAQ#Does_btrfs_support_swap_files.3F) if they are set up with the following requirements:

从内核5.0开始，如果按以下要求设置了[交换文件](https://btrfs.wiki.kernel.org/index.php/FAQ" \l "Does_btrfs_support_swap_files.3F) ，则[可以在Btrfs文件系统中拥有](https://btrfs.wiki.kernel.org/index.php/FAQ" \l "Does_btrfs_support_swap_files.3F)它们：

* No copy-on-write (NOCOW) files.

没有写时复制(NOCOW)文件。

* They’re not compressed.

他们没有被压缩。

* They don’t straddle different hard drives.

它们不会跨在不同的硬盘上。

Most readers will be using the [default ext4 file system](https://www.howtogeek.com/howto/33552/htg-explains-which-linux-file-system-should-you-choose/), so this won’t be a concern to them.

大多数读者将使用[默认的ext4文件系统](https://www.howtogeek.com/howto/33552/htg-explains-which-linux-file-system-should-you-choose/) ，因此他们不必担心。

When [Solid-State Drives (SSDs)](https://www.howtogeek.com/howto/45359/htg-explains-whats-a-solid-state-drive-and-what-do-i-need-to-know/) were first available, there was a concern about using them in situations that had frequent file system writes. People were warned off from putting swap space on SSDs, and even to avoid system logging to SSDs.

当[固态驱动器(SSD)](https://www.howtogeek.com/howto/45359/htg-explains-whats-a-solid-state-drive-and-what-do-i-need-to-know/)首次可用时，存在着在频繁写入文件系统的情况下使用它们的担忧。 人们被警告不要在SSD上放置交换空间，甚至避免将系统记录到SSD。

This is much less of a concern nowadays, and [many SSDs that are on sale have life expectancies that will outlast most PCs](https://www.howtogeek.com/322856/how-long-do-solid-state-drives-really-last/). A swap file on an SSD will have a far better performance than a swap partition on a mechanical hard drive.

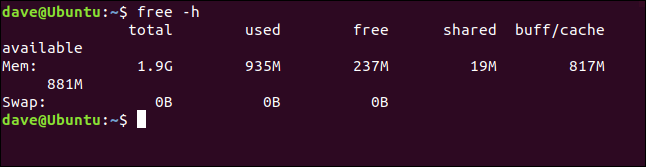
如今，这已不再是一个令人担忧的问题，而且[许多正在销售的SSD的预期寿命将比大多数PC都要长](https://www.howtogeek.com/322856/how-long-do-solid-state-drives-really-last/) 。 SSD上的交换文件比机械硬盘上的交换分区具有更好的性能。

**检查现有交换空间 (Checking Existing Swap Space)**

Look before you leap. Let’s check what swap space is available on your computer. You can do this two ways, and we’ll use both. [The free command will display the used and free memory](https://www.howtogeek.com/456943/how-to-use-the-free-command-on-linux/). The -h (human readable) option will cause free to use sensible units when it displays the memory values.

三思而后行。 让我们检查一下计算机上可用的交换空间。 您可以通过两种方式执行此操作，我们将同时使用这两种方式。 [free命令将显示已用和可用的内存](https://www.howtogeek.com/456943/how-to-use-the-free-command-on-linux/) 。 -h (人类可读)选项在显示内存值时将导致free使用明智的单位。

free -h



The output from free shows that there is no swap space configured on this machine.

free的输出表明该机器上没有配置交换空间。

Swap is never discussed without RAM and free RAM cropping up. So it’s worth noting that the free RAM is given as 237 MB. Don’t mistake that for the total of RAM available for use. That value is provided by the “available” figure, which is 881 MB.

没有RAM和可用RAM的出现，绝不会讨论交换。 因此，值得注意的是，可用RAM为237 MB。 不要误以为可用的RAM总量。 该值由“可用”数字提供，为881 MB。

Linux uses free RAM for its own purposes, such as file caching and kernel buffers. The amount of RAM dedicated to that is the “buff/cache” figure, which is 871 MB. But that memory is still regarded as—and counted as—“available.” The contents of the “buf/cache” RAM can be discarded immediately and used by any application that needs some memory.

Linux将空闲的RAM用于自己的目的，例如文件缓存和内核缓冲区。 专用于此的RAM数量是“ buff / cache”数字，为871 MB。 但是，该内存仍被视为(并算作)“可用”。 “ buf / cache” RAM的内容可以立即丢弃，并可由需要内存的任何应用程序使用。

Another way to check if swap space is available is to use the swapon command. The --show option doesn’t make any [changes to the swap on your computer](http://man7.org/linux/man-pages/man8/swapon.8.html). It only provides statistics.

检查交换空间是否可用的另一种方法是使用swapon命令。 --show选项不会[对计算机上的交换](http://man7.org/linux/man-pages/man8/swapon.8.html)进行任何[更改](http://man7.org/linux/man-pages/man8/swapon.8.html) 。 它仅提供统计信息。

swapon *--show*

swapon --show in a terminal window

If there is no output from this command, there’s no swap configured.

如果此命令没有输出，则没有配置交换。

If these commands had revealed some swap space is already configured, the size of the existing swap space should be factored into decisions regarding the size of the swap file you’re going to create.

如果这些命令表明已经配置了一些交换空间，则应在决定要创建的交换文件大小时考虑现有交换空间的大小。

**我需要多少交换空间？ (How Much Swap Space Do I Need?)**

The traditional response was “twice the amount of RAM you have.” But this was coined when computers used to have very limited RAM. As RAM has become cheaper, and programs and games more demanding of memory, PC specifications have adjusted accordingly. Home PCs with 32 GB of RAM are not uncommon nowadays. And you’re not going to allocate 64 GB of hard drive space to swap space if you’ve got 32 GB of RAM. That’s plainly excessive.

传统的响应是“将您的RAM数量增加两倍”。 但这是在计算机过去具有非常有限的RAM时创造出来的。 随着RAM变得越来越便宜，程序和游戏对内存的要求越来越高，PC规格也相应地进行了调整。 如今，具有32 GB RAM的家用PC并不少见。 如果您有32 GB的RAM，则不会分配64 GB的硬盘驱动器空间来交换空间。 显然太过分了。

The amount of swap you need is as an incendiary subject, comparable to “which is the best editor.” One of the most sensible discussions we’ve seen on this topic is in the [Ubuntu swap FAQ](https://help.ubuntu.com/community/SwapFaq#How_much_swap_do_I_need.3F). It’s a short and commonsense approach (although, like many people, they misunderstand [how swappiness works on Linux](https://www.howtogeek.com/449691/what-is-swapiness-on-linux-and-how-to-change-it/)). There’s a handy table that shows a recommended amount of swap space for the amount of RAM your system has, and whether you hibernate your computer or not.

您需要的交换数量是一个燃烧的主题，相当于“这是最好的编辑器”。 关于此主题的最明智的讨论之一是[Ubuntu交换常见问题解答](https://help.ubuntu.com/community/SwapFaq#How_much_swap_do_I_need.3F) 。 这是一种简短且常识的方法(尽管像许多人一样，他们误解[了swappiness在Linux上的工作原理](https://www.howtogeek.com/449691/what-is-swapiness-on-linux-and-how-to-change-it/) )。 有一个方便的表格，其中显示了建议的交换空间，以表示系统具有的RAM量，以及是否使计算机Hibernate。

And the good news is, it doesn’t really matter what value you pick. We can always remove a swap file and replace it with a bigger one or, for that matter, a smaller one. Or you could just add another swap file.

好消息是，选择什么值并不重要。 我们总是可以删除一个交换文件，并用一个较大的文件或一个较小的文件替换它。 或者，您可以仅添加另一个交换文件。

Pick [a swap file size from the table](https://help.ubuntu.com/community/SwapFaq#How_much_swap_do_I_need.3F), and run it for a while. Monitor your system’s use of the swap space. If fine-tuning is required, changes are easily made. With swap files, It’s a two-minute job. Compare that to adjusting partitions on a live Linux computer.

[从表中](https://help.ubuntu.com/community/SwapFaq#How_much_swap_do_I_need.3F)选择[一个交换文件大小](https://help.ubuntu.com/community/SwapFaq#How_much_swap_do_I_need.3F) ，然后运行一段时间。 监视系统对交换空间的使用。 如果需要微调，则可以轻松进行更改。 使用交换文件，这是两分钟的工作。 将其与调整实时Linux计算机上的分区进行比较。

# 创建交换文件 (Creating the Swap File)

You shouldn’t use the fallocate command [to create your swapfile](http://man7.org/linux/man-pages/man1/fallocate.1.html). This is from the man page for swapon:

您不应该使用fallocate命令[来创建交换文件](http://man7.org/linux/man-pages/man1/fallocate.1.html) 。 这是来自swapon的手册页：

1. The swap file implementation in the kernel expects to be able to write to
2. the file directly, without the assistance of the file system.
4. This is a problem on files with holes or on copy-on-write files on file
5. systems like Btrfs. Commands like cp(1) or truncate(1) create files with
6. holes. These files will be rejected by swapon.
8. Preallocated files created by fallocate(1) may be interpreted as files
9. with holes too depending of the filesystem. Preallocated swap files are
10. supported on XFS since Linux 4.18.
12. The most portable solution to create a swap file is to use dd(1) and
13. /dev/zero.

So, although fallocate is faster, we’ll use dd to [create the swap file](http://man7.org/linux/man-pages/man1/dd.1.html). The machine used to research this article has two GB of RAM. We’re going to create a one GB swap file.

因此，尽管fallocate速度更快，但我们将使用dd [创建交换文件](http://man7.org/linux/man-pages/man1/dd.1.html) 。 用于研究本文的计算机具有两个GB的RAM。 我们将创建一个1 GB的交换文件。

The options are:

选项包括：

* **if**: The input file. In this example, we’re using /dev/zero which will provide a stream of zero bytes.

**if** ：输入文件。 在此示例中，我们使用/dev/zero ，它将提供零字节流。

* **of**: The output file. We’re going to create a file in the root directory, called swapfile.

**of** ：输出文件。 我们将在根目录中创建一个名为swapfile 。

* **bs**: The block size in bytes. This specifies how many bytes to read from the input file and to write to the output file, at a time.

**bs** ：块大小(以字节为单位)。 这指定一次要从输入文件读取并写入输出文件的字节数。

* **count**: How many blocks to read and write. Multiply this number by the bs value to get the file size.

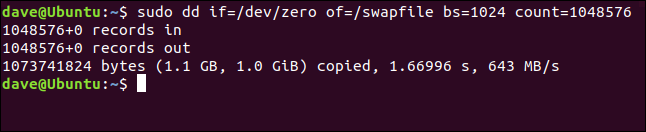
**count** ：要读写的块**数** 。 将此数字乘以bs值即可得到文件大小。

sudo dd if=/dev/zero of=/swapfile bs=1024 count=1048576

sudo dd if=/dev/zero of=/swapfile bs=1024 count=1048576 in a terminal window

Some statistics are provided when the file is created.

创建文件时会提供一些统计信息。



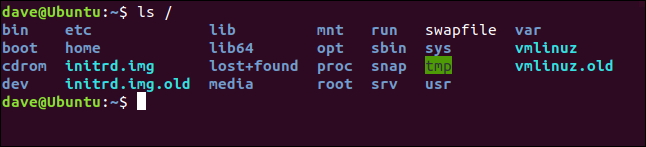
We can see the number of blocks (records) that were written to the file, the size of the file, the time taken to create the file, and the effective data transfer rate.

我们可以看到写入文件的块(记录)数，文件大小，创建文件所花费的时间以及有效数据传输率。

Use the ls command to see the file in the root directory:

使用ls命令查看根目录中的文件：

ls /

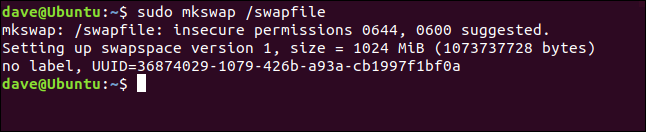


**准备交换文件 (Preparing the Swap File)**

We need to [prepare the swap file](http://man7.org/linux/man-pages/man8/mkswap.8.html) with the mkswap command before it can be used. We don’t need to provide any parameters to mkswap other than the path and name of the file:

我们需要先使用mkswap命令[准备交换文件](http://man7.org/linux/man-pages/man8/mkswap.8.html) ，然后才能使用它。 除了文件的路径和名称之外，我们不需要为mkswap提供任何参数：

sudo mkswap /swapfile



The file is prepared for use as a swap file. Note the warning about file permissions. We’ll need to change those so that the root user is the only one who can read and write to the swap file.

该文件已准备好用作交换文件。 请注意有关文件权限的警告。 我们需要更改这些设置，以便root用户是唯一可以读写交换文件的用户。

**使用交换文件 (Using the Swap File)**

The default permissions are too liberal, we need to restrict them so that only root can use the swapfile. [Use chmod to change the file permissions](https://www.howtogeek.com/437958/how-to-use-the-chmod-command-on-linux/):

默认权限太宽松了，我们需要限制它们，以便只有root才能使用swapfile。 [使用chmod更改文件权限](https://www.howtogeek.com/437958/how-to-use-the-chmod-command-on-linux/) ：

sudo chmod 600 /swapfile

sudo chmod 600 /swapfile in a terminal window

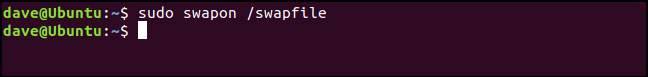
This removes all permissions from the file group members and others, but allows the file owner, root, to read and write to the file.

这将删除文件组成员和其他成员的所有权限，但允许文件所有者(根)读取和写入文件。

We need to use the swapon command to let Linux know there is a new swap file available to use. We only need to provide the path and the filename:

我们需要使用swapon命令让Linux知道有一个新的交换文件可供使用。 我们只需要提供路径和文件名：

sudo swapon /swapfile



The swap file is now active.

交换文件现在处于活动状态。

**将交换文件添加到fstab (Adding the Swap File to fstab)**

To make sure your swap file is available after a reboot, [add it to the /etc/fstab file](https://www.howtogeek.com/444814/how-to-write-an-fstab-file-on-linux/). You can use any text editor you prefer, but we’ll show the process [using the graphical Gedit text editor](https://www.howtogeek.com/413514/how-to-edit-text-files-graphically-on-linux-with-gedit/).

为确保交换文件在重启后可用， [请将其添加到/etc/fstab文件中](https://www.howtogeek.com/444814/how-to-write-an-fstab-file-on-linux/) 。 您可以使用自己喜欢的任何文本编辑器，但是我们将[使用图形化的Gedit文本编辑器](https://www.howtogeek.com/413514/how-to-edit-text-files-graphically-on-linux-with-gedit/)显示该过程。

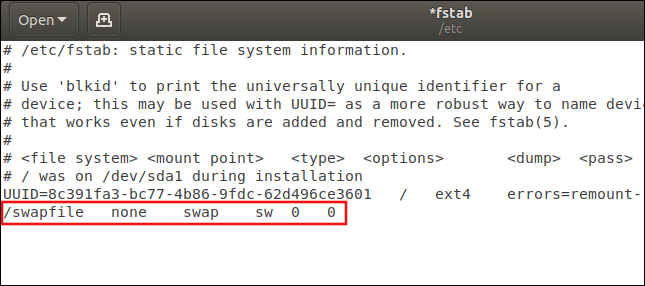
sudo gedit /etc/fstab

sudo gedt /etc/fstab in a terminal window

The line we need to add to the bottom of the file is:

我们需要添加到文件底部的行是：

/swapfile    none    swap    sw    0    0



The fields are:

字段是：

* **File system**: The path and name of the swap file.

**文件系统** ：交换文件的路径和名称。

* **Mount point**: The file isn’t mounted like a file system, so the entry is “none.”

**挂载点** ：文件的挂载方式不像文件系统那样，因此条目为“ none”。

* **Type**: This is “swap.”

**类型** ：这是“交换”。

* **Options**: At boot time swapon -a (start all devices marked as swap) will be called from one of the boot scripts. This option tells Linux to treat this entry as a swap resource that should come under the control of that swapon -a command. It is common to see “defaults” used here because there is a mistaken belief amongst some Linux users that this field is ignored. As we shall see, that is not the case. So it makes sense to use the correct option.

**选项** ：在引导时， swapon -a (启动所有标记为swap的设备)将从一个引导脚本中调用。 此选项告诉Linux将此项视为交换资源，应在该swapon -a命令的控制之下。 经常在这里看到“默认值”是因为一些Linux用户错误地认为此字段被忽略了。 我们将看到，事实并非如此。 因此，使用正确的选项很有意义。

* **Dump**: This can be set to zero. It is irrelevant in this case.

**转储** ：可以设置为零。 在这种情况下无关紧要。

* **Pass**: This can be set to zero. It is irrelevant in this case.

**通过** ：可以设置为零。 在这种情况下无关紧要。

Save the changes and close the editor.

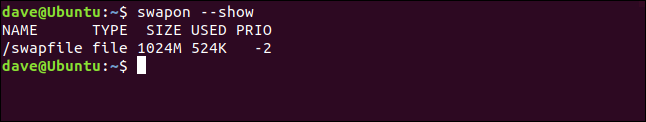
保存更改并关闭编辑器。

**检查交换使用情况 (Checking Swap Usage)**

To see if your swap space is being used, use the swapon command with the --show option:

要查看是否正在使用交换空间，请使用带有--show选项的swapon命令：

swapon *--show*



The columns are:

这些列是：

* **Name**: The name of the swap partition or swap file.

**名称** ：交换分区或交换文件的名称。

* **Type**: The type of swap device.

**类型** ：交换设备的类型。

* **Size**: The size of the swap resource.

**大小** ：交换资源的大小。

* **Used**: The amount of used swap space.

**已用** ：已使用的交换空间量。

* **Prio**: The priority of this swap space.

**Prio** ：此交换空间的优先级。

**交换空间优先级 (The Swap Space Priority)**

Each swap space is allocated a priority. If you don’t provide one, one is automatically allocated. Automatically allocated priorities are always negative.  The range of priorities that can be manually allocated is 0 to 32767. Swap resources with higher priorities are used first.

每个交换空间都分配了一个优先级。 如果您不提供，则会自动分配一个。 自动分配的优先级始终为负。 可以手动分配的优先级范围是0到32767。优先级高的交换资源将首先使用。

If more than one swap space has the same priority they are used alternately until they are both full, then the system looks for the swap space with the next lowest priority. If you only have a single swap space then the priority is irrelevant of course. But we’ll change the priority of the swapfile we’ve created to demonstrate how to do it.

如果多个交换空间具有相同的优先级，它们会交替使用，直到它们都已满，然后系统会寻找具有次低优先级的交换空间。 如果只有一个交换空间，则优先级当然是无关紧要的。 但是，我们将更改为演示如何创建交换文件而设置的优先级。

To set a priority, add the  pri=  (priority) option to the /etc/fstab entry. Edit the line you added to /etc/fstab to look like this:

要设置优先级，请将pri= (优先级)选项添加到/etc/fstab条目中。 编辑添加到/etc/fstab ，如下所示：

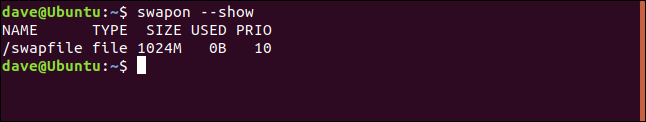
/swapfile    none    swap    sw,pri=10    0    0

That is, add pri=10 to the options field, separated from the “sw” with a comma. Do not leave any spaces between the “sw”, the comma, and the “pri=10.” Reboot your computer and use the swapon --show command:

也就是说，将pri=10添加到选项字段，并用逗号将其与“ sw”分开。 在“ sw”，逗号和“ pri = 10”之间不要留任何空格。 重新启动计算机，然后使用

swapon --show命令：

swapon *-- show*



The priority of this swap space has been elevated to 10. Which is proof positive that the options field in the /etc/fstab entry is not ignored.

该交换空间的优先级已提高到10。这证明了/etc/fstab条目中的options字段未被忽略。

# 交换空间变得简单 (Swap Space Made Easy)

Cutting through the exposition and explanation, we can create a new swap file as easily and quickly as this:

通过阐述和解释，我们可以像这样轻松而快速地创建一个新的交换文件：

# sudo dd if=/dev/zero /of=/swapfile2 bs=1024 count=104857

sudo mkswap /swapfile2

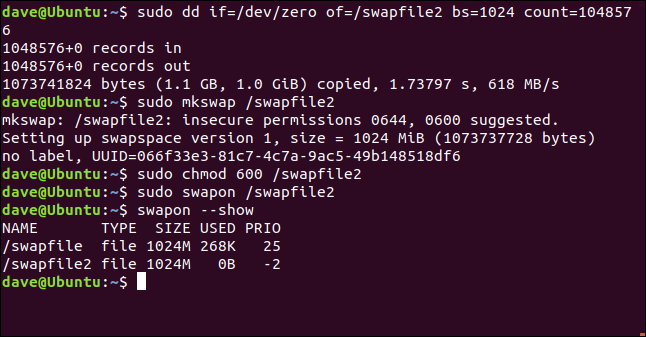
sudo chmod 600 /swapfile2

sudo swapon /swapfile2

And let’s check that it worked:

让我们检查一下它是否有效：

swapon *--show*



If you want to make that permanent drop, it into your /etc/fstab file.

如果要永久删除，请将其放入/etc/fstab文件中。

Boom. Job done.

繁荣。 任务完成。

翻译自: <https://www.howtogeek.com/455981/how-to-create-a-swap-file-on-linux/>

linux创建交换分区设置

linux 创建swap交换分区

swap 作为 linux 系统虚拟内存一样的存在，当系统的物理内存不够用的时候，就需要将物理内存中的一部分空间释放出来，以供当前运行的程序使用。那些被释放的空间可能来自一些很长时间没有什么操作的程序，这些被释放的资源被临时保存到Swap空间中，等到那些程序要运行时，再从Swap中恢复保存的数据到内存中 ，具体物理内存与虚拟内存占比 本人服务器 8G物理内存 swap 同样为8G， 以下为创建命令(服务器事先并没有swap分区)

创建

dd if=/dev/zero of=/swapfile1 bs=1024 count=8388608

1

2

//创建swap文件系统

mkswap /swapfile1

1

2

//启用交换分区文件

swapon /swapfile1

1

2

//系统启动启用swap 最后一行添加 /swapfile1 swap swap defaults 0 0

vi /etc/fstab

1

2

卸载删除

// 卸载swap文件

swapoff /swapfile1

1

2

3

//修改/etc/fstab文件 去掉最后一行 /swapfile1 swap swap defaults 0 0

vi /etc/fstab

1

2

//删除文件

rm -rf /swapfile1