

# Use `/proc/version` to identify your Linux release

Hi everyone, I'm finally back from my holidays, and simply cannot wait to share some more Unix tips with all of you!

Today I'll talk a bit more about yet another way of learning version information about your Linux OS: the `/proc/version` file. I mentioned it briefly in one of the previous posts, but would like to finish the explanations.

## What you can learn from `/proc/version`

This file will not show you the name of the actual OS release, but will instead give you specifics about the version of Linux kernel used in your distribution, and confirm the version of a GCC compiler used to build it.

If you cat the `/proc/version` file, this is what you're going to see (I'm using a RedHat 5.2 system for this):

```
rhel52# cat /proc/version
Linux version 2.6.18-92.el5 (brewbuilder@ls20-
bc2-13.build.redhat.com) (gcc version 4.1.2 20071124 (Red Hat
4.1.2-41)) #1 SMP Tue Apr 29 13:16:15 EDT 2008
```

In this output, you get to see the following information:

1. Exact version of the Linux kernel used in your OS: **Linux version 2.6.18-92.el5**
2. Name of the user who compiled your kernel, and also a host name where it happened: **brewbuilder@ls20-bc2-13.build.redhat.com**
3. Version of the GCC compiler used for building the kernel: **gcc version 4.1.2 20071124**
4. Type of the kernel – **SMP** here means Symmetric MultiProcessing kernel, the one that supports systems with multiple CPUs or multiple cpu cores

5. Date and time when the kernel was built: **Tue Apr 29 13:16:15 EDT 2008**

It's absolutely normal that the kernel is older than the overall release of yours. My example above, generated on the RedHat Enterprise Linux 5.2 system (RHEL5.2), shows the kernel birthday to be **Apr 29, 2008**. But the actual RHEL5.2 release became available to all the customers only a month later, on **May 21st 2008** (here's the original [RedHat 5.2 announcement](#)).

The reason your kernel is a bit older than the rest of the distribution is because kernel is only one part of the final product you're getting – it may take a while to compile and integrate the rest of the OS before it can be used.

## Different ways to find out Linux release information

By now, you should know quite a few ways of confirming release information about your Linux distro. Just to remind you, here they are:

- [use /etc/redhat-release or /etc/SuSE-release \(/etc/issue for Ubuntu\)](#)
- [uname helps you identify if you're using a 32bit or 64bit kernel](#)
- [lsb\\_release command shows you release information as per LSB specification](#)

This should be more than enough even for the most curious Linux users. Enjoy!

## See also:

- [Unix glossary](#)
- [Finding out your GCC compiler version](#)

---

# How To Monitor Linux Memory Usage with Watch Command

Hi all, today I'm going to teach you not one, but two really cool things in one post! First, I'll introduce you to advanced memory usage stats available on Linux systems through `/proc/meminfo` file, and then I'll explain the basics of using the [watch command](#).

## Memory usage with `/proc/meminfo`

As you know, quite a few Unix-like systems use the so-called pseudo file systems like `/proc`. It's not a real filesystem, but just a convenient representation of processes managed by your Unix OS. In Linux systems, this directory also contains quite a few files allowing you to access various information about your system. `/proc/meminfo` is one of such files, it gives you access to most of the memory usage stats.

To get a snapshot of the current state of memory usage on your Linux system, simply cat the `/proc/meminfo` file:

```
ubuntu$ cat /proc/meminfo
MemTotal:      523008 kB
MemFree:       35336 kB
Buffers:       85560 kB
Cached:        137220 kB
SwapCached:    24480 kB
Active:        327420 kB
Inactive:      91308 kB
HighTotal:     0 kB
HighFree:      0 kB
LowTotal:      523008 kB
LowFree:       35336 kB
```

```
SwapTotal:      1048568 kB
SwapFree:       998960 kB
Dirty:          504 kB
Writeback:      0 kB
Mapped:         212232 kB
Slab:           39140 kB
CommitLimit:   1310072 kB
Committed_AS:  655992 kB
PageTables:    4748 kB
VmallocTotal:  34359738367 kB
VmallocUsed:   628 kB
VmallocChunk:  34359737739 kB
```

This probably gives you more information about memory usage that you'll ever want to know, but there's quite a few really useful stats there like the MemFree or SwapFree ones, they're useful for making sure your OS environment is healthy enough in terms of having plenty of free memory for efficient operation.

## Using watch command to track progress

[watch command](#) is a really neat tool which does a simple but incredibly useful thing: it repeatedly runs a given command line and shows you the output. So, you're effectively monitoring a progress of some process by watching relevant files.

The default interval is 2, which gives enough dynamics for most of the needs.

Here's how you use this command:

```
ubuntu$ watch cat /proc/meminfo
```

So it's the same command we used in previous example, `cat /proc/meminfo`, but this time we're asking the [watch command](#) to re-run the command every 2 seconds and show us the output.

The result of running a watch command is going to be a constantly refreshed console showing something like this:

Every 2.0s: cat /proc/meminfo

Fri Feb 13 03:51:01 2009

```
MemTotal:      523008 kB
MemFree:       46396 kB
Buffers:       82636 kB
Cached:        131044 kB
SwapCached:    24480 kB
Active:        308512 kB
Inactive:      99372 kB
HighTotal:     0 kB
HighFree:      0 kB
LowTotal:      523008 kB
LowFree:       46396 kB
SwapTotal:     1048568 kB
SwapFree:      998960 kB
Dirty:         832 kB
Writeback:     0 kB
Mapped:        211076 kB
Slab:          39132 kB
CommitLimit:   1310072 kB
Committed_AS: 654860 kB
PageTables:    4856 kB
VmallocTotal: 34359738367 kB
VmallocUsed:   628 kB
VmallocChunk: 34359737739 kB
```

This output gets refreshed every 2 seconds, so the numbers shown are constantly updated.

That's it for today! There are limitless possibilities for monitoring various processes using **watch command** and I'll be sure to cover them in the future, but for now – have a great weekend and hope Friday 13th turns out great!

## See also:

- [How to determine physical memory size in Linux](#)
- [Unix file types](#)
- [Unix glossary](#)

---

# Unix Tutorial Digest: Interesting Links #1

Every week there's a few announcements or articles which I find particularly interesting, and so I've decided to share them with you. I'm not a Unix guru (yet), but if any of the listed materials require further explanation – do feel free to ask and I'll be glad to help.

## Ubuntu 8.04.1 release

About a week ago, the first update to Ubuntu 8.04 was announced – [Ubuntu 8.04.1 TLS](#). I have completed my experiment of using Ubuntu Hardy as my desktop OS a few weeks ago, and so haven't upgraded yet – but I think this release is not so useful for anyone who's been automatically updating their system – it's just another milestone and a way to download a complete Ubuntu 8.04.1 as one image.

The highlights for me would be Firefox upgraded to the final 3.0 release and Gnome upgrade (it's 2.22.2 in this release).

## Gentoo Linux 2008.0 release

For some of you, it's probably been a long-awaited release. Move to 2.6.24 kernel provided support for much more hardware, and this is bound to look good with the updated and much improved Gentoo installer.

Read more in the official [Gentoo Linux 2008.0 announcement](#).

## Cache poisoning vulnerability in DNS

Dan Kaminsky has found quite a [nasty weakness in DNS](#)

implementations: deficiencies in the DNS protocol and common DNS implementations facilitate DNS cache poisoning attacks.

Thanks to the seriousness of the problem and a great coordination, most of the vendors were given the time to publish a fix, so the [Vulnerability Note VU#800113](#) contains a comprehensive list of vulnerable implementations of DNS (both server and client sides are affected, by the way!) and links to fixes provided by various vendors.

Whether you're managing a server farm or just a Linux desktop – be sure to update!

## **Wine 1.1.1 release**

Things are going much faster with [Wine](#) development after the 1.0 release – it didn't take long for the 1.1 to appear, and now almost every other week brings another great update with tons of bugs fixed.

[Wine 1.1.1 release](#) includes more than 50 bugfixes and hundreds of changes since Wine 1.1.0, notably the fixes for Adobe Photoshop CS3 and Microsoft Office 2007 installers, as well as improved video playback and many other improvements.

---

# **How To Determine Physical Memory Size in Linux**

If you're logged in at some remote Linux system and need to quickly confirm the amount of available memory, there's a few commands you will find quite useful.

## free – free and used memory stats

free command is the most obvious choice for a first command when it comes to your RAM.

Simply run it without any parameters, and it will show you something like this:

```
ubuntu# free
              total            used            free           shared
buffers      cached
Mem:         4051792          4024960          26832             0
63768       3131532
-/+ buffers/cache:      829660      3222132
Swap:        4096492           43212          4053280
```

For this exercise, you're only interested in the "total" column of the first line. 4051792 confirms that my home PC seems to have around 4Gb of memory available for Ubuntu to use.

## Using dmesg to check memory size as recognized by Linux kernel

**dmesg** command shows you the last status messages reported by your OS kernel, and since every boot procedure includes scanning the hardware and confirming the devices and resources recognized by the kernel, you can see some basic information by using **dmesg**.

For our purpose, we need to filter out the memory stats:

```
ubuntu# dmesg | grep Memory
[  18.617904] Memory: 4043492k/5242880k available (2489k
kernel code, 150360k reserved, 1318k data, 320k init)
```

Once again, the overall amount of memory confirms that 4Gb of RAM were still found during the last time my PC booted up.

# Using `/proc/meminfo` to confirm the RAM size

`/proc/meminfo` is one of the special files managed by Linux kernel. It's a clear text presentation of the most vital memory stats of your system (this means you can do something like `cat /proc/meminfo` to see all the parameters)

This is what you need to do to get the total size of your physical memory:

```
ubuntu# grep MemTotal /proc/meminfo  
MemTotal:      4051792 kB
```

That's it for today, enjoy!