How To: List PCI Information From The Terminal (lspci)

Continuing with a theme, this article will explain how to use the terminal to view what's attached via PCI (Peripheral Component Interconnect) to your computer. PCI devices are the ones in the add-on slots in your computer. They're typically internal devices, though you can get USB powered external devices with PCI-e slots, they kind of defeat the purpose. For this article, we'll be using lscpi.

You may also be interested in reading up about lshw and lscpu. I didn't intend for this to be a 'series' but that's what it's starting to look like. That's nice, however. They're small bites that let you sample the buffet that is the Linux terminal.

This one is just a little more complicated than the last. For starters, you may not have lscpi immediately available. You may have it installed by default, but you might not. It really depends on your distro. Either way, it's in your default repositories.

If you don't have it installed, it's in the package 'pciutils'. So, use your package manager to install it. For example, with apt you'd do this:

[code]sudo apt install pciutils[/code]

Once you have it installed, you'll see the manual defines 'lspci' as:

lspci — list all PCI devices

If you don't know, PCI devices are a class of devices that are add-ons to your motherboard. They're mostly the devices that go in the slots, such as graphics cards or sound cards. They

can be used for all sorts of things, these days even being used for M.2 devices that hold SSD drives for rapid storage. They're great because they have a fast bus speed, meaning that your computer can interact with them faster because data moves faster in both directions than it does for, for example, a USB device.

Except PCI devices aren't just limited to the things that go in slots. Your motherboard probably uses that same spec to interact with other devices. For example, your Ethernet and sound card may be listed – even though they're 'on-board' and not actually add-on cards. That's your motherboard using the same sort of bus spec but not actually using a physical port. Additionally, there are different types of PCI specifications, but we don't need to get into that today.

And, well, once again the 'lspci' name tells you what it does, now that you know that's what it does. It lists PCI devices, just as the name implies. There are a couple of ways that you'd realistically want to use it.

NOTE: The results from 'lspci' are drawn from the The PCI ID Repository and may actually not be accurate. Yup. You could get inaccurate results from this command, but we throw it around daily as though it's infallible. And now you know…

Where were we? Oh, yeah… We were going to use 'lscpi' for something useful. So, let's crack open that terminal by using your keyboard to press CTRL + ALT + T and we'll first enter:

[code]lspci[/code]

That will list all your PCI devices in a quick and easy to read list. You may also want to get the verbose output and that's done like:

[code]lspci -v[/code]

The output of that command should be fairly obvious. After

all, it does what it says it does on the tin. That's also perhaps the more useful of the commands. It's definitely the one that I use most frequently.

If you take the numbers from the start of each line from the output of the above command and use the -t switch you'll actually get an understandable 'tree' output that will help you further understand what's going on inside your case without opening it up. If one PCI device has multiple entries (as many do) it'll make that easier to understand. It's simply:

[code]lspci -t[/code]

However you can easily put the two of those together and simply get a great verbose tree output with:

[code]lspci -vt[/code]

That's plenty easy to understand but some folks may find it a bit overwhelming. I don't usually need that much information, so I tend to run the command without any switches. As I bounce between devices, it's enough to just check and make sure I know what I'm working with.

NOTE: Older versions required *-vvv* for verbose and *-tree* were needed to perform those operations. The current versions simply use the *-v* and *-t* switches.

And there you have it. Yet another way to view hardware information from within the terminal. You may have noticed a trend and probably can narrow down your guess as to what the next article is gonna be about! If lscpi's parent package is not installed, it's really easy to get and you can then run the command. If you don't have access to many tools, you almost certainly have access to this one.

Like always, thanks for reading. Be sure to sign up for the newsletter. I'll only send you site-related material and won't

sell your email address to anyone. I promise, I won't ever send you any spam — just site stuff!

How To: List CPU Information From The Terminal (lscpu)

In the last article I wrote, I explained how to use 'lshw' to get information about your hardware from the terminal. This article will show you how to gather CPU information from the terminal.

This will be yet another quick article, but again this is a valuable tool to do hardware diagnostics when you don't have other tools available.

Just like last time, the command is self-evident once you really look at it. The command is 'lscpu' and it does what you'd expect — 'list CPU information'. It is defined as thus in the manual:

lscpu — display information about the CPU architecture

So, let's give it a shot. Crack open your default terminal emulator by pressing CTRL + ALT + T. And, let's just start by entering:

[code]lscpu[/code]

And, there you have some fairly easy to understand. It's often used as a quick way to tell if your CPU supports 64 bit instructions. You'll see something like this if it does:

[code]Architecture: x86_64[/code]

NOTE: Pretty much all modern computer hardware supports the 64 bit instruction set, but there are still some 32 bit machines out there and being used.

Unlike the 'lshw' command, this one doesn't need to be run as an administrator. It runs as a regular user just fine and running it as an admin doesn't get you any more information.

Also unlike the 'lshw' command, there isn't any other way to run it that's all that interesting. You can read the man page, but you'll seldom need to use this in any other way other than the way described here. Just type the command, get the information, and move on with whatever it was you were doing!

See? I told you this one would be another quick and easy article. Thanks for reading and don't be scared of signing up to the newsletter. It's not like you'll be inundated with piles of unwanted email! You'll just get notified when there's a new article and I promise to not sell your email address to anyone.

How To: List Hardware From The Terminal (lshw)

You may not always have inxi available. The person trying to help you may want more specific info. You may need more information about your hardware. Who knows? There's all sorts of reasons to use 'lshw' in your day-to-day computing.

Today is another short and easy article (you're welcome). It's just a very brief command, one of several, that we're going to

learn how to use. The command in question is 'lshw'. The manual helpfully defines it as:

lshw — list hardware

Which, now that you look at it, makes perfectly good sense. This is one of those Linux commands that's not at all cryptic when you think about it. It looks like what it does, it does what you'd intuitively think it does. (It's actually a subset of the info you get with 'hwinfo', but that's not important right now!)

What is important is that there are two realistic ways to use it — and both of them should be run as an administrator. The first is just as you'd expect:

[code]sudo lshw[/code]

That will output a ton of information about the hardware you have in (or connected to) your computer. It's rather overwhelming and it's not often that you'll be asked to post all of it. You may be asked to (or want to) run it with the -C option, such as:

[code]sudo lshw -C cpu[/code]

You may even be asked to use grep with it, such as:

[code]sudo lshw | grep wireless[/code]

Then, there's one more way to run the program (under normal circumstances) and that's to run the whole thing and to get the whole output in a shorter format. The command is probably just like you'd expect.

[code]sudo lshw -short[/code]

Lo and behold! Would you look at that! Ha! Isn't that fantastic? You actually get readable output that's suitable for your normal user who just wants to know what sorta

hardware they're working with while getting a few extra bits of info! It even includes juicy nuggets like:

[code]/1 wlxe4beed0e5f5c network Wireless interface[/code]

Now, if you want to refine it even further, why not try this:

[code]sudo lshw -short | grep network[/code]

And, there you have it… You have another way to see your hardware in the terminal and it should be consistent across any major distro you touch. You shouldn't have any trouble using the command and it's easy to remember when you want to 'list the hardware'. There are other uses, but those two are the most common. For more information about the lshw command try:

[code]man lshw[/code]

or

[code]info lshw[/code]

Like always, thanks for reading. Feel free to subscribe to the newsletter. I promise to not give away your private information — and you never know what sorta incentives I may stick in there. See? I told you this one would be short and easy!

How To: Time a Command

Have you ever wanted to know how long it takes to complete a command that you entered in the Linux terminal? Well, wonder no more!

This is going to just be a pretty quick article and easy to

follow. There's not a whole lot to explain and it's pretty straightforward. Like often, let's crack open your default terminal by pressing CTRL + ALT + T on your keyboard.

Now, let's take the command:

[code]ls -la[/code]

Unless you have a lot of files, that completes pretty quickly. But, how fast does it really take? Well, simply add the 'time' command before it. Time is simply described in the man page as:

time – run programs and summarize system resource usage

And, for today, it's going to be pretty easy to use that command. To find out how long it took to list all the files and folders in a directory, you could use:

[code]time ls -la[/code]

Note how it tells you the time beneath the results and, if you want to try something bigger, you can take a look at this command:

[code]for i in {0..99999}; do echo "I love LinuxTips!"; done[/code]

That should take a just a little more time, but you can actually see how long it really took by adding 'time' in front of it. So:

[code]time for i in {0..99999}; do echo "I love LinuxTips!"; done[/code]

The output at the end is something like this:

real 0m0.566s
user 0m0.423s
sys 0m0.143s

The 'real' is how much time it really took. The 'user' is how much time it took for the user. The 'sys' is how much time it took for the system — the amount of time that the kernel actually devoted to it.

So, there you have it! You can use the time command to find out how long it takes to run stuff in your terminal. If you're playing with scripting and you're looking to optimize it, this is a valuable tool. If you're just a bit curious, then you now have a new tool.

I told you that it'd be quick and easy! Like always, thanks for reading. Feel free to sign up for the newsletter. I promise to not spam you or sell your email address.

How To: Make Ubuntu Show Asterisks When Typing Password

By default, Ubuntu doesn't show anything when you enter your password in the terminal. This is for security reasons. Someone shoulder-surfing won't be able to see the number of characters in your password. This is how to get some feedback when you enter your password in the terminal.

This one is pretty easy and shouldn't take very long. First, let's open your terminal. Press CTRL + ALT + T and your default terminal should open. Yay!

Now, we need to edit the sudoers file. It's done like this:

[code]sudo nano /etc/sudoers[/code]

Enter your password and hit enter, of course. (This will be the last time you enter your password in the terminal without some sort of visual feedback!)

Now it gets a little tricky.

Use the down arrow until your at the start of the line that says:

[code]Defaults   mail_badpass[/code]

Press the ENTER button. This should move that line down and leave a blank line above it. Use the arrow button to move up to that blank line and enter:

[code]Defaults[/code]

Then press the TAB button on your keyboard. This will move the cursor to the right location. Add this text:

[code]pwfeedback[/code]

The entire line should look something like:

[code]Defaults   pwfeedback[/code]

Note: This spacing isn't really required so much as it is done for convention and to aid in ease of reading/processing information-dense text more accurately and swiftly. If you want to be diligent, you can even leave a comment, prefaced with a **#**, remarking that you made a change and why you made a change. Comments should be on their own lines.
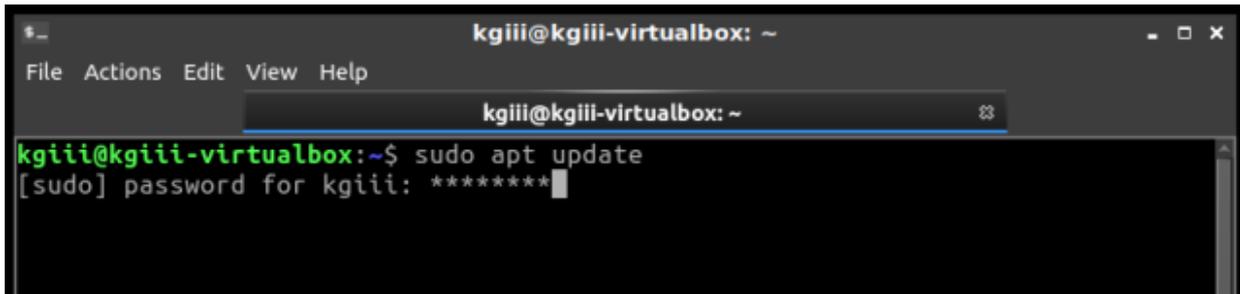
Anyhow…

Now, you simply need to save the file. If you've been following along, you'll already know how to do that. If not, here it is again:

Press CTRL + X, then Y, and then ENTER.

Congrats, you're done! You may need to close and reopen your terminal to notice the difference. Test it by opening a new terminal window and tying in:

[code]sudo apt update[/code]

Type your password when prompted and you'll hopefully see some asterisks as feedback. It should look a little like this:

See? Asterisks for feedback in the terminal.

As always, thanks for reading. Feel free to sign up for the newsletter. The only emails you'll get are notifications when there are new articles. I won't even send you any spam, I pinky swear! Also, I think I'm going to settle on the Helvetica font. It's pretty clear, easy to read, and easy to distinguish numerals from alphabetical characters. I should probably go back through my old articles and make this consistent, but it's too much fun writing new articles!