

12.2.1 Network Interface Naming

Click one of the buttons to take you to that part of the video.

Network Interface Naming 0:00-0:22

The first thing you need to do when you're configuring a Linux system to connect to an Ethernet network is to install a network interface card which we just call a NIC.

Now most modern systems include an integrated Ethernet interface that's built into the motherboard itself. If this is the case you're already good to go. All you have to do is plug it into the network.

Network Interface Module Installation 0:23-1:20

If your system does not include an integrated network interface then you first of all need to power the system off. That's very important to do. And then install an Ethernet board in an available expansion slot. And then connect your Ethernet board to your network switch or hub with the appropriate drop cable. And then power the system back on.

If you installed a modern Ethernet board then your Linux operating system should detect it as it boots up and then automatically load whatever kernel modules are required.

If you purchased and installed an off brand Ethernet network interface or a very old Ethernet interface, then you may have to go to the card vendor's website and download the appropriate module and load it manually.

Most of the time that isn't necessary though. Then after the system is booted you need to be check your module configuration file and verify that the appropriate kernel module was loaded for your network interface and that an alias has been created for the new network board.

Network Interface Module Configuration 1:19-1:59

Now the file that you need to check depends upon which distribution that you're using. Some distributions will use the `/etc/modprobe.conf` file or maybe the `/etc/modules.conf` file.

Some others will use a configuration file located in the `/etc/modprobe.d` directory and that's the case in this example. Within my `/etc/modprobe.d` directory, there's a file called `50-vmnics.conf`. This configuration file specifies which kernel modules need to be loaded in order to support this network board.

Network Device Aliases Under init 1:56-3:24

The alias that gets assigned to your network interface will also depend upon your distribution. Older distributions that are based on the `init` daemon will assign `eth0` to the first Ethernet adapter in your system. It'll assign `eth1` to the second one, `eth2` to the third one, `eth3` to the fourth network adapter and so on.

Newer distributions that are based on `systemd`, instead of `init`, no longer use these aliases and believe me the transition from the old aliases to the new aliases has been difficult for those of us who grew up with Linux. We're used to the first Ethernet board in the system being `eth0`. That's the way it's always been. Well not anymore.

Now that old system worked reasonably well unless you happen to have multiple network adapters in the system, two or more. Here's what happened. Frequently, we experienced situations where the interface aliases would get swapped between network adapters.

For example, we might boot the system up one time and one particular interface in the system would be assigned an alias of `eth0` but then we would reboot the system and the next time it comes up that same network interface is assigned `eth1` instead and the other network adapter got assigned `eth0`.

As you can imagine this situation can really mess up your network services. It messes up your routing tables. It messes up your firewall rules, and it even messes up your IP addressing.

Predictable Network Interface Names 3:25-6:41

To fix this issue, the systemd daemon uses what we call predictable network interface names. One of the benefits of doing this is the fact that specific aliases are permanently assigned to specific network interface cards.

In essence, the same adapter always has the same alias assigned to it. The aliases are defined and assigned based upon several different parameters and it goes in the order shown here.

First of all, if we're dealing with an onboard network adapter one that's built into the motherboard itself, then an index number is provided by the system firmware either UEFI or the BIOS and it's used to construct the alias.

An example is shown here. First of all, en indicates that it's an Ethernet interface and then the 0 and the 1 indicates the onboard device index number, in this case it's just device number one.

In this case, the network interface integrated into the motherboard would always be assigned the alias of eno1. If the network adapter is plugged into a PCI slot instead of being integrated into the motherboard, then systemd will use the index number of the slot itself, and this is again provided by the UEFI firmware or the BIOS on the system, and it will use this information to construct the alias.

An example of that is shown here. Again, we have 'en' to indicate that this is an Ethernet interface and then 's1' indicates the hot plug slot index number. This is where the network interface card has been inserted into the motherboard.

If the first two options are not applicable, for example if we have connected a network interface into a USB connector instead of into a slot or being integrated into the motherboard, then systemd is going to try to construct an alias for the network interface using the name of the physical connector that the interface is connected to.

Here's an example here. Again, we use en to indicate that it's an Ethernet interface and then p2 identifies the bus that the interface is connected to. In this case, we're looking at USB bus 2 and the s0 indicates the slot number on the bus that the device is connected to so in this case it's connected to Slot 0.

Now there may be a situations when the first three options aren't applicable and they fail and so when this happens systemd is going to try to use the interface card's MAC address to construct an alias.

As you can see here because we're using the MAC address the alias name gets really, really long. We use en to identify that it's an Ethernet interface, and then an x to indicate we're using a MAC address, and then we put in the MAC address that's assigned by the card.

Now because this alias is a little bit funky and long and difficult to work with, this rule right here actually sometimes on some distributions is actually disabled by default. If none of these first four rules will work then systemd is going to default back to the old init naming standard and just call the interface eth0 which warms my heart and brings a tear to my eye, because that's how I learned Linux.

Non-Ethernet Device Names 6:40-7:37

In the preceding examples every single interface that we looked at started with en, because they're Ethernet interfaces. They're connected to an Ethernet switch, and they participate on an Ethernet network. However, this is not the only kind of interface you can install in the Linux system.

If you install a different type of interface in the system and connect it to a different type of network, a different prefix is used for the alias name.

For example, if we install a serial line interface in the Linux system and use it for say a WAN connection, then the name of that serial line interface is going to begin with sl instead of en to indicate that it's a serial interface not an Ethernet interface.

Likewise a lot of Linux systems will use wireless interfaces. If you've installed a wireless network interface in the system, the alias for those interfaces will begin with wl instead of en.

Summary 7:38-7:50

That's it for this lesson. In this lesson, we reviewed how Linux creates aliases for network interfaces. We first reviewed how the older init daemon created network device aliases. Then we reviewed how systemd uses predictable network interface names to create network device aliases.

Copyright © 2022 TestOut Corporation All rights reserved.