The vast majority of network programming in Linux is done using the socket interface. Thus, standards-compliant programs should require little massage to work properly with Linux.

Note, however, there are many enhancements and new features in the Linux networking implementation, such as new kinds of address and protocol families. For example, Linux offers the **netlink** interface, which permits opening up socket connections between kernel sub-systems and applications (or other kernel sub-systems). This has been effectively deployed to implement firewall and routing applications.

Historically, the wired Ethernet network devices have been known by a name such as **eth0**, **eth1**, etc., while wireless devices have had names like **wlan0**, **wlan1**, etc.

Basic information about active network interfaces on your system is gathered through the **ifconfig** utility:

```
$ /sbin/ifconfig
eth0      Link encap:Ethernet  HWaddr 00:22:15:2B:64:A6
          inet addr:192.168.1.100  Bcast:192.168.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:163529 errors:0 dropped:0 overruns:0 frame:0
          TX packets:112693 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:183642176 (175.1 MiB)  TX bytes:12101864 (11.5 MiB)
          Interrupt:18
eth1      Link encap:Ethernet  HWaddr 00:22:15:2B:63:BE
          inet addr:192.168.0.101  Bcast:192.168.0.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:162597 errors:0 dropped:0 overruns:0 frame:0
          TX packets:56710 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:206698846 (197.1 MiB)  TX bytes:75532637 (72.0 MiB)
          Interrupt:17

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:15115 errors:0 dropped:0 overruns:0 frame:0
          TX packets:15115 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:126793920 (120.9 MiB)  TX bytes:126793920 (120.9 MiB)
```

Information displayed includes information about the hardware MAC address, the MTU (maximum transfer unit), and the IRQ the device is tied to. Also displayed are the number of packets and bytes transmitted, received, or resulting in errors.

This machine has two network cards bound to **eth0** and **eth1**, and the loopback interface, **lo**, which handles network traffic bound to the machine. Note you can see the statistical information in abbreviated form by looking at **/proc/net/dev**, and in one quantity per line display in **/sys/class/net/eth0/statistics**:

```
$ ls -l /sys/class/net/eth0/statistics
total 0
-r--r--r-- 1 root root 4096 Mar 26 17:21 collisions
-r--r--r-- 1 root root 4096 Mar 26 17:30 multicast
-r--r--r-- 1 root root 4096 Mar 26 17:20 rx_bytes
-r--r--r-- 1 root root 4096 Mar 26 17:30 rx_compressed
-r--r--r-- 1 root root 4096 Mar 26 17:30 rx_crc_errors
-r--r--r-- 1 root root 4096 Mar 26 17:30 rx_dropped
-r--r--r-- 1 root root 4096 Mar 26 17:20 rx_errors
-r--r--r-- 1 root root 4096 Mar 26 17:30 rx_fifo_errors
-r--r--r-- 1 root root 4096 Mar 26 17:30 rx_frame_errors
-r--r--r-- 1 root root 4096 Mar 26 17:30 rx_length_errors
-r--r--r-- 1 root root 4096 Mar 26 17:30 rx_missed_errors
-r--r--r-- 1 root root 4096 Mar 26 17:30 rx_over_errors
-r--r--r-- 1 root root 4096 Mar 26 17:20 rx_packets
-r--r--r-- 1 root root 4096 Mar 26 17:30 tx_aborted_errors
-r--r--r-- 1 root root 4096 Mar 26 17:20 tx_bytes
-r--r--r-- 1 root root 4096 Mar 26 17:30 tx_carrier_errors
-r--r--r-- 1 root root 4096 Mar 26 17:30 tx_compressed
-r--r--r-- 1 root root 4096 Mar 26 17:30 tx_dropped
-r--r--r-- 1 root root 4096 Mar 26 17:21 tx_errors
-r--r--r-- 1 root root 4096 Mar 26 17:30 tx_fifo_errors
-r--r--r-- 1 root root 4096 Mar 26 17:30 tx_heartbeat_errors
-r--r--r-- 1 root root 4096 Mar 26 17:20 tx_packets
-r--r--r-- 1 root root 4096 Mar 26 17:30 tx_window_errors
```