Getting Small with Linux, Part 1

SysAdmin (/tag/sysadmin)

by Marcel Gagné on March 31, 2000

Every once in a while, you find yourself in a situation where nothing but Linux will do. Okay, so that's most days, but occasionally you wind up sitting in front of a workstation running *that other OS*. You'll want to access resources in ways where only Linux seems to provide the tools you need. The problem, of course, is that Linux isn't running on that workstation. Luckily, you've got *Linux in your pocket*, on a 3-1/2-inch floppy diskette, no less.

Sound crazy?

This week, Red Hat released version 6.2 of their excellent distribution. It comes on a CD-ROM with several hundred megabytes of information, and provides some great tools. It's a lot of Linux. So is Caldera, Slackware, and Debian. Recently, I've been spending a little time exploring an other side of Linux, asking how "little" I can have and still reap the benefits of my favorite OS. It turns out you can do quite a bit with just a little. Let me introduce you to my first candidate for tiny Linuxes, and then I'll show you just how useful these tools can be to the system administrator.

Since we are talking about Linux on a floppy, it is only fitting that I start with **LOAF**, which stands for **L**inux **O**n **A** **F**loppy. LOAF is available by surfing over to http://loaf.ecks.org/ (http://loaf.ecks.org).

When you get to the site, you'll notice there are several versions (or images) available, nine to be exact. Each image contains a specific network driver, so you'll want to get the one (or ones) that

best suit the environment in your office. I picked up **loaf3.img**, which contained the 3Com 3x59x driver. Then I popped in a blank diskette, and created my diskette using this command:

```
dd if=loaf3.img of=/dev/fd0
```

Throw a label on it, and that's all you need. To test this out, I took the diskette to the only PC in my office still running that other OS, put it in the driver and rebooted the PC.

When the login prompt comes up, just type **root** and away you go. To get LOAF to recognize your network and work on it, you need to execute the following commands:

```
ifconfig eth0 local_ip_address netmask your_network_m
route add -net network_address
echo "nameserver gateway_address" > /etc/resolv.conf
route add default gw gateway_address
```

In my case, I was using a local private Class C network at 192.168.1.0 and the client address was at 192.168.1.2. My gateway and nameserver were at 192.168.1.10 (a Linux server doing IP masquerading). Here's what it actually looks like:

```
ifconfig eth0 192.168.1.2 netmask 255.255.255.0 broad
route add -net 192.168.1.0
echo "nameserver 192.168.1.10" > /etc/resolv.conf
route add default gw 192.168.1.10
```

I typed **lynx http://www.linuxjournal.com**, and suddenly, I am surfing the Web.

LOAF includes a nice set of useful tools. You get things like **ifconfig**, **traceroute** and of course **lynx**, which lets you happily surf either the Web or your local network. Incidentally, if you just type **lynx**, you get the local LOAF documentation. (Don't worry; it's a quick read.) You also get an **ssh** (secure shell) client for secure access to your network, good old **telnet**, and **ftp.** LOAF is a great lightweight networking client. Heck, if all you wanted to do was run dumb terminal applications with an old 386, LOAF may be for you. The only catch is that LOAF doesn't know anything about hard drives. For a micro-Linux that lets you have local hard disk access, read on.

Another cool little super-tiny Linux is **tomsrtbt**, which stands for "*Tom's floppy which has a root filesystem and is also bootable.*" Hey, I couldn't make this stuff up. What Tom has achieved here is

practically a miracle of miniaturization. He uses a great little trick to stretch what he can put on a floppy by using *1.7 MB of diskette space* rather than the normal 1.4. Despite all that, what he's accomplished is still pretty amazing. You can get tomsrtbt at http://www.toms.net/rb/ (http://www.toms.net/rb).

As I write this, the latest distribution in a tarred and gzipped file is **tomsrtbt-1.7.185.tar.gz**. After downloading, you simply extract it and create the diskette from the install script. In its simplest form, the installation goes like this. Start by putting a blank diskette in your drive and follow these steps:

```
tar -xzvf tomsrtbt-1.7.185.tar.gz
cd tomsrtbt-1.7.185
./install.s
```

That is all you really need in order to create the tomsrtbt diskette. Earlier, with LOAF, we had to configure the network settings with each boot, but not so with tomsrtbt. Consequently, you will find that the default installation boots to a *pre-configured set of network addresses*, including DNS entries. To customize tomsrtbt for your own network, edit the **settings.s** file in the distribution directory. After my changes, the file looks something like this:

```
DOMAIN=yourcompany.com
IF_PORT=10baseT
DNS_1=192.168.22.10
IPADDR=192.168.22.3
NETWORK=192.168.22.0
NETMASK=255.255.255.0
PASSWD=xxxx
FD=/dev/fd0u1722
FN="b 2 60"
```

The original network numbers were in the network "192.168.1.0" and the domain was rb.com.

When I first booted, I was *delighted* to learn that the date was "Boomtime, the 14th day of Discord in the YOLD 3166". This is the **ddate**, a small program courtesy of the Church of the SubGenius, which translates Gregorian dates into those of the Discordian calendar. After a brief pause, I started exploring what tomsrtbt provides.

The sheer bulk of tools available is impressive, to say the least. tomsrtbt has **SCSI** support and **PCMCIA** support. It contains a handful of useful shell tools (in no particular order) like find, grep, vi, e2fsck, tar, cpio, and a host of other command-line tools. This

tiny Linux *does* know about hard drives, and lets you mount them. For instance, I created a directory called /mnt/ddrive and mounted my PC's D drive like this:

```
mount /dev/hd5 /mnt/ddrive
```

After doing that, I can do things such as create tar archives on the disk, which I can then ftp to another server on my network. Since there are a number of Linux file system tools available, this makes an ideal generic rescue diskette for when all else has failed. The possibilities are, as they say, virtually endless. I can also mount network drives with NFS. What LOAF gets you that tomsrtbt does not is a web browser (lynx) and a secure shell (ssh).

While this doesn't really have anything to do with what tomsrtbt does, I did like his logo and decided to include it here.

```
    .~.
    /V
   // \\
  /(   )
   ^`~'^
```

Many times, I have been working in a strange office and thought, "If I only had my Linux system here, I could do..." Or its close cousin in the land of penguin thoughts, "If only this PC were running Linux, then I could..." Well, if dragging your Linux system from one part of the building to another isn't practical, but you need the power of a Linux system, consider the power of Linux on a floppy. Not too long ago, I had an instance where a Windows machine was more or less dead. I could not boot. I could not access the network, or any device on which to back up (diskette was not an option). Important data was at stake, and the PC would not cooperate. What saved the day was a portable Linux that enabled network services, let me archive and compress the files, and then allowed me to **ftp** them to a safe place (a couple of places, actually). Once my data was safe, it was time to reinstall Windows. The lesson for the day was either "don't underestimate the power of a little Linux", or "a little Linux goes a long way". In either case, much wine was drunk that night and all was right with the world.

Whether you choose **LOAF**, **tomsrtbt** or one of the other tiny Linuxes out there will depend on precisely what you need to do. Heck, considering how little pocket space these guys take up, why not carry a few different flavors? That way, you can be ready for anything. Here's a parting thought that might encourage you to carry one of these diskettes around. If you are working on a

network where you question the security of the various machines (maybe someone is watching or listening), then you have the guarantee of an OS that is completely untainted by whatever compromise may exist on the network. After all, when your OS resides entirely in RAM, not much will be able to touch it.

If nothing else, these tiny distributions clearly illustrate the possibilities of Linux in small network appliances or embedded systems. Faithful readers, I have seen the future. Next week, we climb a step on the ladder of tiny Linux. The world is still small, but the tools are all powerful. We'll look at what system administrators could only have *dreamed* of a few years back. Until next week, remember, even the littlest Linux can measure up.
email: ljeditors@ssc.com (mailto:ljeditors@ssc.com)