

Tech Tips with Gnull and Voyd

Chester Gnull

Laverta Voyd

Abstract

Howdy. My husband is Chester Gnull and I'm Laverta Voyd, and I'm the lady to light a way for all you sweethearts out there who do fancy stuff with Linux. Me and my husband's gonna be bringing you tech tips just about every month now. I reckon you and yours are wondering why my husband's and me's last names don't match. Well, Chester don't like much in the way of attention, so he got the editor to change our last names so's we don't get no pesky e-mails or people messin' with us in real life.

I don't know nothing about Linux. Chester, he's the smart one, but he's not much of a talker. That's why I'm here. He don't do nothing without me, and I don't mind much cause I like talkin' and I like hosting. Chester don't understand why we gotta talk at all, but that's what the editor wants, and well, he's paying us, so we figure there ain't nothing wrong with that. So those *LJ* folks are gonna send us the tips, my Chester does the pickin' and I do the hosting. And, I say, I do love hosting, but seeing as this here's just writing stuff, we ain't gonna be serving up none of my specials like biscuits and gravy with sausage and real maple syrup, and it's all homemade but the maple syrup. But they tell me the tips are just as tasty to you Linux folk. That don't make much sense to me, but Chester says that's how it is and I believe my Chester.

Now honeys, we got some tips to start. One tip is by the editor to get things rolling. He don't get no \$100 but I figure he gets enough just being editor. So, we want you to send us some of your tips. If we put your tech tip in this here column, you get \$100. We know that ain't gonna get you no Fleetwood mobile home, and I'm talkin Park Models, not even them fancy Entertainer Models with two bathrooms. But \$100 will get you some good eats at your local Piggly Wiggly. So send them tips in, sweethearts, and we'll appreciate it real nice. You send 'em on in to techtips@linuxjournal.com and the editors will pass 'em on to Chester for ya, and we'll do the rest.

Modify initrd to Make 3ware RAID the First Serial Device

This tip makes Ubuntu see a 3ware RAID controller as the first serial device on your system in Ubuntu. -- Chester

As you can see, Chester's real wordy, huh? That's why he's wrangled me into doing this. I mean, he's my lovin' man and I know that 'cause he shows me. But it wouldn't kill ya to say three little words now and again, would it, Chester? -- Laverta

Three little words. Happy? -- Chester

You can install a RAID card in your PC and configure the BIOS to make the BIOS consider the RAID card to be the first SCSI device on your system. But, Ubuntu (and probably other distributions) do not necessarily respect your BIOS settings. For example, I have an ASUS M2N32 WS Professional motherboard, which includes a PCI-X slot for the 3ware 9550SX-4LP RAID card. I can set the BIOS to

make this card the first device. However, if I add a SATA drive, the Ubuntu initrd will see the onboard SATA as the first SCSI device on the system, in spite of the BIOS settings.

There may be a kernel boot parameter to override this behavior, but I haven't found one that works. Regardless, I like the following solution if for no other reason than it teaches one how to extract, modify an Ubuntu initrd and then reassemble it for use.

Here's why the Ubuntu initrd defies the BIOS settings. The initrd for Ubuntu runs the script shown in Listing 1.

Listing 1. The initrd scripts/local-top/udev File

```
#!/bin/sh -e
# initramfs local-top script for udev

PREREQ=""

# Output pre-requisites
prereqs()
{
    echo "$PREREQ"
}

case "$1" in
    prereqs)
        prereqs
        exit 0
    ;;
esac

# Each call to udevplug can take up to three minutes
if [ -x /sbin/usplash_write ]; then
    /sbin/usplash_write "TIMEOUT 540"
    trap "/sbin/usplash_write 'TIMEOUT 15'" 0
fi

# Load drivers for storage controllers found on the
# PCI bus; these show up the same for both IDE and
# SCSI so there's no point differentiating between
# the two. Do it in serial to try to provide some
# predictability for which wins each time.
/sbin/udevplug -s -Bpci -Iclass=0x01*

# We also need to load drivers for bridges (0x06),
# docking stations (0x0a), input devices (0x09),
# serial devices (0x0c) and "intelligent" devices
# (0x0e). This is both to support filesystems on the
# end and just in case there's a keyboard on the end
# and things go wrong.
/sbin/udevplug -Bpci -Iclass=0x0[69ace]*

# If we're booting from IDE, it might not be a PCI
# controller, but might be an old-fashioned ISA
# controller; in which case we need to load ide-generic.
/sbin/modprobe -Qb ide-generic
/sbin/udevplug -W
```

The following line, which discovers storage controllers, happens to discover the NVIDIA SATA first:

```
/sbin/udevplug -s -Bpci -Iclass=0x01*
```

You can force this script to find the 3ware controller first by adding a line that explicitly loads the 3ware module before this line. Listing 2 shows how to modify the script to do that (Listing 2 is only an excerpt from the relevant part of the script).

Listing 2. Add the line to discover the 3ware card first.

```
/sbin/modprobe 3w-9xxx

# Load drivers for storage controllers found on the
# PCI bus; these show up the same for both IDE and
# SCSI, so there's no point differentiating between
# the two. Do it in serial to try to provide some
# predictability for which wins each time.
/sbin/udevplug -s -Bpci -Iclass=0x01*
```

This forces the script to discover the 3ware RAID card first and assign it as `/dev/sda` before `udevplug` discovers the rest of the PCI storage controllers.

The trick here is that you need to unpack the default `initrd` file that comes with Ubuntu, modify this script, and then repack it and use it instead of the default `initrd`.

Here's one way to do that. These instructions assume you are using Ubuntu Dapper AMD64 with the kernel `2.6.15-27-amd64-generic`. If you're using some other kernel, you must change the command accordingly. You can be more careful than I have been with these instructions and use `sudo` for all the appropriate commands. However, I jumped into a root shell with the `sudo -s -H` command to make this easier to read:

```
$ sudo -s -H
(enter password)
# cd /root
# mkdir initrd-tmp
# cd initrd-tmp
# gzip -dc /boot/2.6.15-27-amd64-generic | cpio -id
```

This unpacks your `initrd` so that you can manipulate its contents. Now, edit this file. (Use whichever editor suits you. I am using `vi` as an example.)

```
# vi scripts/local-top/udev
```

This is the file that contains the code in Listing 1. Add the `modprobe` command as shown in Listing 2. Save the file.

All this assumes that the module `3w-9xxx` exists in your `initrd`. If it doesn't, or you need some other module in your `initrd`, you'll have to copy it to the following location (once again, this assumes you are using the `2.6.15-27-amd64-generic` kernel -- modify as necessary for your setup):

```
# cp <module> /root/initrd-tmp/lib/modules/
↳2.6.15-27-amd64-generic/kernel/drivers/scsi
```

Now you need to repack the `initrd` file. I suggest that you name this `initrd` something other than the original, so that if you have done something wrong, you can revert to the original easily.

Here is how to repack the file to a new `initrd`. This assumes your current working directory is still `/root/initrd-tmp`:

```
# find . | cpio --quiet --dereference -o -H newc | gzip
```

```
↵-9 > /boot/2.6.15-27-amd64-generic-3w
```

Now change your bootloader to add another boot option to use the new initrd file. You can replace the existing boot entry, but that's asking for trouble (although GRUB, for example, lets you edit a boot entry at boot time, so there's always hope if you use GRUB). If you use GRUB, specify the modified initrd as the initrd image, like this:

```
initrd    /boot/initrd.img-2.6.15-27-amd64-generic-3w
```

Reboot, and try it out.

This should work for cards other than the 3ware if you are having the same problem with another RAID card (or even some other storage card). All you have to do is change `/sbin/modprobe` to load the appropriate module for your card. Don't forget to check to see whether the driver module exists in the unpacked initrd before you pack it again.

-- Nicholas Petreley

Knoppix Does More Than Show Off Linux to Windows Users

Your computer won't boot because you been using one of them unofficial kernels, I bet. That'll get you in a heap of trouble. It's yer own fault. Boot a Linux live CD to fix the damage you did. -- Chester

It happens to the best of us, you sit at your computer in the morning, turn it on, and find that it won't boot properly. After an hour of troubleshooting, diagnostics and grumbling, you come to the conclusion that something about your hard drive is toast. You think of all the files you may have just lost in the process and curse the fact that you didn't back up diligently enough.

Most of the time when your OS is dead, your files are still intact on the drive; you just have to find a way to get to them. In some cases, your problem may be that the root partition is too corrupted to mount it, but not so corrupt that you can't restore it. For example, your root partition may be formatted as XFS, and all you need to do is run a utility like `xfs_repair` on the partition to get things back in order.

Some distributions come with a repair disk, and some installation disks have a repair option. But, you might find it more useful to boot to a live CD to make repairs, because a live CD may put more utilities at your disposal than a repair disk. Knoppix is one of many live CD versions of Linux that runs straight from the CD and allows you access to the hard drives.

Even if you are in a worst-case scenario and have to recover individual files, all you need to recover files, or possibly the entire contents of the hard drive, is a copy of Knoppix (or your favorite live CD distro) and a portable hard drive, jump drive or some other kind of USB portable storage device. Or, if you have an unused SATA or IDE spot in your system, you always can open up the computer and plug in the extra drive (properly configured, of course). If you go portable, then how big the portable storage device is depends on how much you want to save.

Double-check the BIOS on your target computer to make sure it is set to boot from a CD. If your BIOS allows you to interrupt the boot sequence with the Esc key, F8, or some other key in order to choose which drive to boot, you may not even have to reconfigure your BIOS. Regardless, boot from CD, and Knoppix should boot up automatically into the desktop.

Once in the desktop, all that's left to do is search the computer's hard drive and find the files to salvage and

transfer to your portable media device or additional internal device. Finding the files will require that you know where the file is on the hard drive, and this will be more or less difficult depending on the filesystem on the drive that was corrupted.

-- Brad Hall

Finding Disk Space and inode Hogs

I knowed somebody was gonna get to this problem sooner or later. You get too many inodes on your system, and you're asking for another heap of trouble. This tells you how to find out and fix it. -- Chester

One of the most common tasks of a system administrator is storage management. When you're faced with a full or almost full filesystem, it's good to have a few tools at your disposal to help figure out "where" the hog is.

Searching for space hogs is very easy. With one simple command you can total up the contents of every directory in a tree (/usr in this case) and see which are the largest:

```
du -k /usr/ |sort -n |tail -30
167156 /usr/include
168960 /usr/share/icons
173972 /usr/share/texmf
244332 /usr/bin
263144 /usr/lib/openoffice.org2.0
265492 /usr/share/doc
344536 /usr/share/locale
1223992 /usr/lib
1959412 /usr/share
4159996 /usr/
```

As you can see, /usr/share/ and /usr/lib/ are pretty big, and you can drill down further by going up the list.

A somewhat rarer situation is running out of inodes in a filesystem. In this case, you will see available space, but the system will be unable to write new files because it has run out of inodes. To find inode hogs, use this quick Perl script named inodu:

```
#!/usr/bin/perl -w

my $start=$ARGV[0];

foreach $object (`find '$start'`){
    my @parts=split(/\//,$object);
    while(pop(@parts) ){
        my $object = join('/',@parts);
        $object =~ s/\/+\/\//g;
        $object2qty{$object}++;
    }
}

foreach $object (sort { $object2qty{$a} <=> $object2qty{$b} }
    keys %object2qty){
    print $object2qty{$object} . "\t${object}\n";
}
}
```

This will total up the number of filesystem objects in each directory and supply an output much like the

previous example. Use it like this:

```
cd /usr
./inodu .
10420 ./include
10973 ./share/texmf
12012 ./share/man
13207 ./share/doc
14953 ./share/icons
16481 ./src/kernels
17201 ./src
22982 ./lib
105527 ./share
174270 .
174271
```

As you can see, again share and lib are the inode hogs using more than 100,000 inodes!

If you find yourself in any of these situations, there are a number of ways to create more free space or inodes. First, look for log files that can be purged, moved or compressed. Ask users to clean up their home directories. Remove any unnecessary software. If you are using Linux LVM and ext3fs, you can expand the filesystem using `lvresize` and `resize2fs` to grow a filesystem. This creates more free space and inodes, but only if you have free space in your volume group. If you have free disk space, you can create a new partition (for, say, your `/var` tree), move the files to that partition and mount it as `/var`. As a last resort, you can move files and directories and use symlinks so the old path still works. I say “last resort” because this method can get out of hand very quickly and can make things very confusing.

-- Matthew Hoskins

Credits

- Nicholas Petreley is Editor in Chief of *Linux Journal*.
- Brad Hall lives in Jacksonville, Florida, with his pet chickens and life-size cardboard cutout of *Star Trek: DS9's* Dr. Bashir.
- Matthew Hoskins is Senior Information Systems Analyst at the New Jersey Institute of Technology.

Linux Journal pays \$100 US for any tech tips we publish. Send your tips with your contact information to [<techtips@linuxjournal.com>](mailto:techtips@linuxjournal.com).