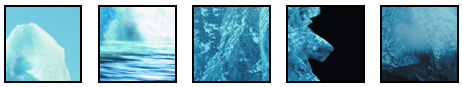

Informative article November, 2003.

Adding hard drives to a Linux host.

stijn.huyghe@thti.telindus.be

Introduction.

It is a simple fact, applications and operating systems demand more and more performance and hard disk space. Sooner or later, everyone is confronted with the fact that your workstation starts to lack available disk space.

In this article, we are going to show you step-by-step how you can add additional hard drives to a Linux system with a minimal amount of frustration.

Note: before start, make sure that you have a working backup.

Situation of the host before the operation.

Welcome to “TheHive”.
“TheHive” is a Fedora Linux host (http://fedora.redhat.com) that needs some additional storage.

Let’s take a look at the current situation by using the “df” command. “df” will report file system disk space usage.

```
[root@TheHive root]# df -h
File system           Size  Used Avail Use% Mounted on
/dev/hdb5             6.6G  1.8G  4.5G  29% /
/dev/hdb1              99M  8.3M   86M   9% /boot
/dev/hde1             113G   80G   28G  74% /data1
/dev/hdf1             113G   74G   34G  70% /data2
/dev/hdb2             4.9G   38M  4.6G   1% /home
none                  62M     0   62M   0% /dev/shm
```

We learn from this output that:
- We are dealing with IDE disks.
IDE disks are identified by “/dev/hd<drive number><partition number>” by the Linux kernel.
- On the first IDE controller, we have one IDE disk (“/dev/hdb”) that holds multiple partitions: “hdb5”, “hdb1”, “hdb2”. These partitions are mounted on “/”, “/boot”, “/home”. This drive is the system disk of “TheHive” that contains the Linux operating system etc.
- The second IDE controller doesn’t carry any disk (there is no “/dev/hdc” or “/dev/hdd”.
- The third IDE controller carries two IDE disks: “/dev/hde” and “/dev/hdf” that contain both one partition (“hde1” and “hdf1”). These are mounted on “/data1” and “/data2”.

A quick view in “/etc/fstab” returns the current fstab configuration:

```
[root@TheHive root]# cat < /etc/fstab
LABEL=/                 /                       ext3    defaults        1 1
LABEL=/boot             /boot                   ext3    defaults        1 2
LABEL=/data1            /data1                  ext3    defaults        1 2
LABEL=/data2            /data2                  ext3    defaults        1 2
none                    /dev/pts                devpts  gid=5,mode=620  0 0
```

```
LABEL=/home          /home              ext3    defaults        1 2
none                 /proc              proc    defaults        0 0
none                 /dev/shm           tmpfs   defaults        0 0
/dev/hdb3            swap               swap    defaults        0 0
/dev/fd0             /mnt/floppy        auto    noauto,owner,kudzu 0 0
```

Note: “fstab” contains descriptive information about the various file system present on the system. Each file system is described on a separate line. These file systems will be mounted on boot time using the parameters set in this file (“man fstab”).

Add the hard drives to the system.

Add the hard drives to the system. We added two IDE drives of 80G to the system.

Make sure that when you add drives, they are recognized by the BIOS (make sure that your master and slave settings are set ok).
If your BIOS detects the drives on the IDE cable, but their size is wrongly detected, don’t worry. Check your BIOS settings and/or try to upgrade your BIOS of your motherboard (consult the website of your vendor).

If an upgrade of the BIOS doesn’t work, don’t worry. Although your BIOS does not detect the complete capacity of the disk, Linux will be able to use the full capacity of the drive in most cases.

In our case, “TheHive” is running on the famous ABIT BP-6 motherboard with the latest firmware that does not support drives of 120G. But if you take a look in the data on top, you can note that Linux is able to use the full capacity of these drives - although the BIOS does not support them.

Partition layout of “TheHive”.

After you added the hard drives, let’s take a look at the current partition layout with “fdisk”. ”Fdisk” is a partition table manipulator for Linux and used with the option “-l” it will list the partition tables for the specified devices and exits (see “man fdisk”).

```
[root@TheHive root]# fdisk -l

Disk /dev/hdg: 81.9 GB, 81964302336 bytes
255 heads, 63 sectors/track, 9964 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot    Start       End    Blocks   Id  System
/dev/hdg1             1      9964  80035798+   7  HPFS/NTFS

Disk /dev/hdh: 81.9 GB, 81964302336 bytes
255 heads, 63 sectors/track, 9964 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot    Start       End    Blocks   Id  System
/dev/hdh1             1      9964  80035798+   7  HPFS/NTFS

Disk /dev/hde: 122.9 GB, 122942324736 bytes
255 heads, 63 sectors/track, 14946 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot    Start       End    Blocks   Id  System
```

```
/dev/hde1   *         1     14946 120053713+  83  Linux

Disk /dev/hdf: 122.9 GB, 122942324736 bytes
255 heads, 63 sectors/track, 14946 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot    Start       End    Blocks   Id  System
/dev/hdf1   *         1     14946 120053713+  83  Linux

Disk /dev/hdb: 13.0 GB, 13020069888 bytes
255 heads, 63 sectors/track, 1582 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot    Start       End    Blocks   Id  System
/dev/hdb1   *         1        13    104391   83  Linux
/dev/hdb2            14       650   5116702+  83  Linux
/dev/hdb3           651       714    514080   82  Linux swap
/dev/hdb4           715      1582   6972210    f  Win95 Ext'd (LBA)
/dev/hdb5           715      1582   6972178+  83  Linux
```

We added two hard disks from an old host running Microsoft Windows to the system. The “fdisk –l” command shows us very clearly that there are two new devices: “/dev/hdg” and “/dev/hdh” both containing a NTFS partition.
The new drives (“/dev/hdg” and “/dev/hdh”) were installed on the 4th IDE controller.

Removing and adding partitions.

Let’s remove the old partitions on “/dev/hdg” and “/dev/hdh” and add some new partitions using fdisk (“man fdisk”). Commands are marked in bold. Comment is added using “**Comment:**”

[root@TheHive root]# fdisk /dev/hdg

```
The number of cylinders for this disk is set to 9964.
There is nothing wrong with that, but this is larger than 1024,
and could in certain setups cause problems with:
1) software that runs at boot time (e.g., old versions of LILO)
2) booting and partitioning software from other OSs
   (e.g., DOS FDISK, OS/2 FDISK)

Command (m for help): help
h: unknown command
Command action
   a   toggle a bootable flag
   b   edit bsd disklabel
   c   toggle the dos compatibility flag
   d   delete a partition
   l   list known partition types
   m   print this menu
   n   add a new partition
   o   create a new empty DOS partition table
   p   print the partition table
   q   quit without saving changes
   s   create a new empty Sun disklabel
   t   change a partition's system id
```

u change display/entry units
v verify the partition table
w write table to disk and exit
x extra functionality (experts only)

Print partition info:
Command (m for help): **p**
Comment: print the partition table on "/dev/hdg".

Disk /dev/hdg: 81.9 GB, 81964302336 bytes
255 heads, 63 sectors/track, 9964 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

Device	Boot	Start	End	Blocks	Id	System
/dev/hdg1		1	9964	80035798+	7	HPFS/NTFS

Comment: delete the partition on "/dev/hdg".
Command (m for help): **d**
Selected partition 1

Comment: verify if the partition has been deleted.
Command (m for help): **p**

Disk /dev/hdg: 81.9 GB, 81964302336 bytes
255 heads, 63 sectors/track, 9964 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

Device	Boot	Start	End	Blocks	Id	System

Comment: create a new partition on "/dev/hdg", create a primary partition.
Command (m for help): **n**
Command action
e extended
p primary partition (1-4)
p
Partition number (1-4): **1**
First cylinder (1-9964, default 1): **1**
Using default value 1
Last cylinder or +size or +sizeM or +sizeK (1-9964, default 9964):
Using default value **9964**

Comment: write the changes to disk.
Command (m for help): **w**
The partition table has been altered!

Calling ioctl() to re-read partition table.
Syncing disks.

Checking…

Comment: verify the modifications.
[root@TheHive root]# fdisk –l

Disk /dev/hdg: 81.9 GB, 81964302336 bytes
255 heads, 63 sectors/track, 9964 cylinders

Units = cylinders of 16065 * 512 = 8225280 bytes

Device	Boot	Start	End	Blocks	Id	System
/dev/hdg1		**1**	**9964**	**80035798+**	**83**	**Linux**

Comment: the partition was created!

Comment: remove the old NTFS partition on "/dev/hdh" and add a new partition.

[root@TheHive root]# fdisk /dev/hdh

The number of cylinders for this disk is set to 9964.
There is nothing wrong with that, but this is larger than 1024,
and could in certain setups cause problems with:
1) software that runs at boot time (e.g., old versions of LILO)
2) booting and partitioning software from other OSs
(e.g., DOS FDISK, OS/2 FDISK)

Comment: print current partition information of "/dev/hdh".
Command (m for help): **p**

Disk /dev/hdh: 81.9 GB, 81964302336 bytes
255 heads, 63 sectors/track, 9964 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

Device	Boot	Start	End	Blocks	Id	System
/dev/hdh1		1	9964	80035798+	7	HPFS/NTFS

Comment: delete partition 1 (NTFS partition in this case).

Command (m for help): **d 1**
Selected partition 1

Comment: check the results.
Command (m for help): **p**

Disk /dev/hdh: 81.9 GB, 81964302336 bytes
255 heads, 63 sectors/track, 9964 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

Comment: NTFS partition has been removed, let's add a new primary partition.

Command (m for help): **n**
Command action
e extended
p primary partition (1-4)
p
Partition number (1-4): **1**
First cylinder (1-9964, default 1): **1**
Last cylinder or +size or +sizeM or +sizeK (1-9964, default 9964):
Using default value 9964

Comment: verify if the partition has been created.
Command (m for help): **p**

Disk /dev/hdh: 81.9 GB, 81964302336 bytes

255 heads, 63 sectors/track, 9964 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

Device	Boot	Start	End	Blocks	Id	System
/dev/hdh1		1	9964	80035798+	83	Linux

Comment: write modifications to disk.
Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.
Syncing disks.

Comment: verify the current partitioning scheme using fdisk. New partitions are marked in bold.

[root@TheHive root]# fdisk -l

Disk /dev/hdg: 81.9 GB, 81964302336 bytes
255 heads, 63 sectors/track, 9964 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

Device	Boot	Start	End	Blocks	Id	System
/dev/hdg1		**1**	**9964**	**80035798+**	**83**	**Linux**

Disk /dev/hdh: 81.9 GB, 81964302336 bytes
255 heads, 63 sectors/track, 9964 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

Device	Boot	Start	End	Blocks	Id	System
/dev/hdh1		**1**	**9964**	**80035798+**	**83**	**Linux**

…

Create file systems on the partitions.

To create a file system on the new partitions, you need to use the “mkfs” command (“man mkfs”). By default, it will create an ext2 file system. You can use other commands or options to this command to create other file system types like “mkfs.ext3” to create an ext3 file system or “mkfs.reiserfs” for the Reiser file system.

We are going to create two ext3 file systems on “/dev/hdh1” and “/dev/hdg1” using the “mkfs.ext3” command.

Comment: create ext3 file system on “/dev/hdh1”.

[root@TheHive root]# mkfs.ext3 /dev/hdh1
mke2fs 1.34 (25-Jul-2003)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
10010624 inodes, 20008949 blocks
1000447 blocks (5.00%) reserved for the super user

First data block=0
611 block groups
32768 blocks per group, 32768 fragments per group
16384 inodes per group
Superblock backups stored on blocks:
32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
4096000, 7962624, 11239424

Writing inode tables: done
Creating journal (8192 blocks): done
Writing superblocks and filesystem accounting information: done

This filesystem will be automatically checked every 32 mounts or
180 days, whichever comes first. Use tune2fs -c or -i to override.

Comment: create an ext3 file system on "/dev/hdg1".

[root@TheHive root]# mkfs.ext3 /dev/hdg1
mke2fs 1.34 (25-Jul-2003)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
10010624 inodes, 20008949 blocks
1000447 blocks (5.00%) reserved for the super user
First data block=0
611 block groups
32768 blocks per group, 32768 fragments per group
16384 inodes per group
Superblock backups stored on blocks:
32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
4096000, 7962624, 11239424

Writing inode tables: done
Creating journal (8192 blocks): done
Writing superblocks and filesystem accounting information: done

This filesystem will be automatically checked every 24 mounts or
180 days, whichever comes first. Use tune2fs -c or -i to override.

Modifying "/etc/fstab".

If you want that the file systems are mounted automatically on boot time, make sure that you modify "/etc/fstab" and that you create the corresponding directories to mount them on (here: "/data3" and "/data4")

```
LABEL=/             /                ext3    defaults        1 1
LABEL=/boot         /boot            ext3    defaults        1 2
LABEL=/data1        /data1           ext3    defaults        1 2
LABEL=/data2        /data2           ext3    defaults        1 2
none                /dev/pts         devpts  gid=5,mode=620  0 0
LABEL=/home         /home            ext3    defaults        1 2
none                /proc            proc    defaults        0 0
none                /dev/shm         tmpfs   defaults        0 0
```

```
/dev/hdb3               swap                 swap    defaults        0 0
/dev/fd0                /mnt/floppy          auto    noauto,owner,kudzu 0 0
/dev/hdg1               /data3               ext3    defaults        1 2
/dev/hdh1               /data4               ext3    defaults 1 2
```

Verify the results:

Let's verify the results on "TheHive" using "df –h".
Result: the new disks are ready to be used!

```
Last login: Mon Nov 24 22:57:35 2003
[root@TheHive root]# df -h
Filesystem            Size  Used Avail Use% Mounted on
/dev/hdb5             6.6G  1.8G  4.5G  29% /
/dev/hdb1              99M  8.3M   86M   9% /boot
/dev/hde1             113G   80G   28G  74% /data1
/dev/hdf1             113G   74G   34G  70% /data2
/dev/hdg1              76G   33M   72G   1% /data3
/dev/hdh1              76G   33M   72G   1% /data4
/dev/hdb2             4.9G   38M  4.6G   1% /home
none                  62M     0   62M   0% /dev/shm
[root@TheHive root]#
```