Technical Report
Algologic Research & Solutions
Version ID: 20090815a

Two is better than one !

S. Parthasarathy
drpartha@gmail.com

Algologic Research and Solutions
78 Sancharpuri Colony
Bowenpally P.O.
Secunderabad 500 011
INDIA

This article introduces the concept of twin-boot (as opposed to dual boot). Twin-boot involves two (or more) Linux distros on the same machine, and two hard disk drives (optional but highly recommended). The distros are cleverly installed, so as to make the best use of the available hardware redundancy. You can do more, and do it faster, with this kind of an arrangement.

The twin-boot experiment

It all started when I was decommissioning my old, self-assembled PC. I was building another Linux box with bigger/better hardware. I got a 80 GB disk, to replace my old 40 GB disk, and a brand new mother board with built in sound interface, and a bigger RAM, and a brand new flat panel LCD monitor. My GOD, I was rich ! Those were good old days when a little was enough to make one feel at the top of the world. Instead of trading in the old (smaller) drive for a new drive, I decided to keep it with me. Moreover, Linux distros were getting easier and more accessible, courtesy, magazines like Linux for You. So, instead of installing just one distro, I chose to install two distros and learn how to handle both. These choices led me to the big question – what would be the best way to get the most out of the hardware and the software I now had with me ? The rest of this article will explain what I did. It may give you some ideas and make you extract more out of your Linux boxes. The “twin boot” experiment itself is fun to go through, and will help you experience several finer points of how Linux works.

The term dual boot has traditionally been used to denote a system where we have Linux and the other black-box OS together. The term “twin boot” is introduced in this article, to denote an “unpolluted Linux box (ULB)”, which has two Linux distributions (and no black-box OS). The twin boot concept helps us to effectively get rid of the cancer which has infected all of us. Additionally, twin boot sytems have two separate hard disks, although it is possible to build twin boot systems on a single hard disk. Having two disks gives us some advantages, as we will see later. There could be several ways to install two Linux distros on a system with two drives. This article shows one way in which this can be done, and discusses the various isues involved. Of course, you can extend the idea to more than two distros, and maybe more than two disks, on the same machine.

A word of caution, before you start making your twin-boot system. Do NOT attempt this if you are a novice and not fully aware of how Linux works. Setting up partitions, setting up the boot loader, and fine tuning the installation etc. , require some prior knowledge and experience in Linux troublshooting and administration. All this is not rocket science, but you can still make serious blunders. So, be careful. Keep a Linux guru close by, as your helpline, if you are still learning the ropes. Since you will be working as root, most of the time, you have to be extra careful.

How to do this

The principle involved in this approach is to install the two distros on the same disk (in two different partitions of course). We set up swap spaces for the two distros, on the other disk. We also make a partition whose mountpoint will be /home. This single /home will be shared by both the distros. For convenience, let us call the two distros as distro A and distro B. And call the two drives as drive P and drive Q. The architecture we choose would look like in Figure 1. The dotted arrow-lines show that the same /home is shared by both distros. There are two swaps, one of each distro (there is a reason for this). Both distros sit on the drive Q (the bigger one), whereas the /home and the swaps are on drive P (the smaller one). In fact, this arrangement is a modified form of the arrangement explained in an article which appeared in Linux for You[1] (February 2009). A similar effort is also explained in the article located at : `http://www.go2linux.org/dual-boot-two-linux-distros-debian-and-mandriva` . This article has a large number of screenshots, to make things easier for you.

In my case, distro A was Open Suse 11, and distro B was Ubuntu 9.04. The drive P was a 40 GB IDE drive and drive Q was a 80 GB SATA drive.

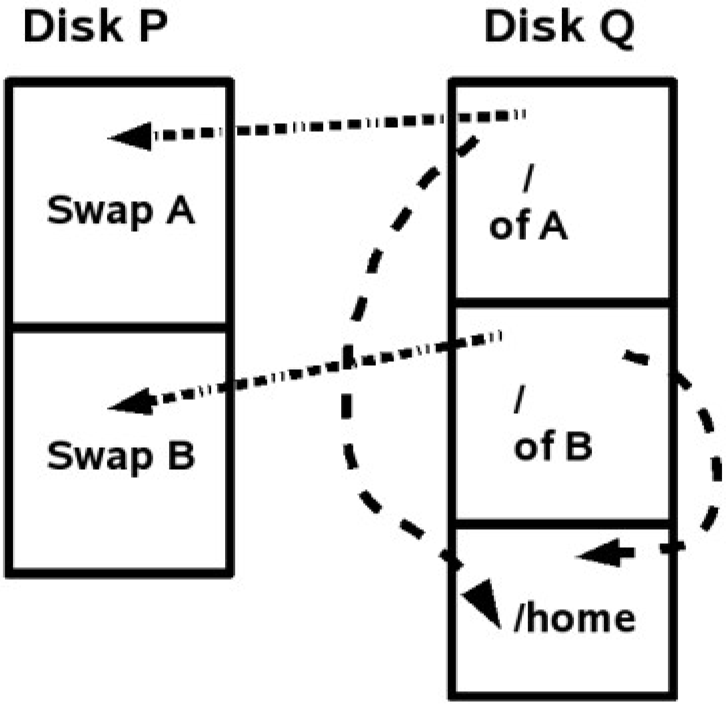


Figure 1 : A twin-boot system

The major steps involved in creating a twin-boot systems are :

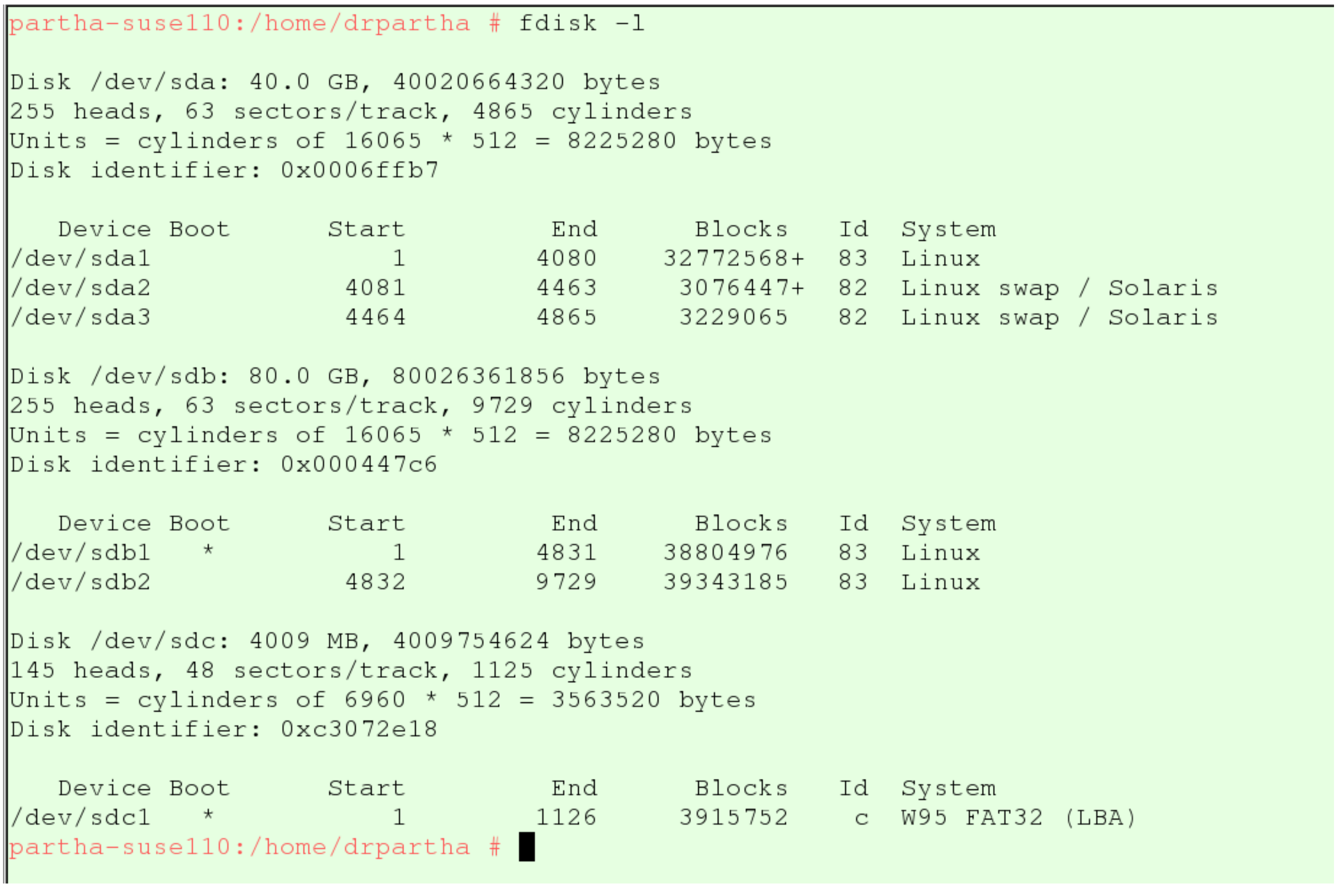
1. Create partitions on Disk P and Disk Q, as above.
2. Install the two OS on the disk Q. Configuer each OS to use the appropriate

[1] Improve multi-OS performance through cross swapping, Saurav Sengupta, Linux For You, February 2009.

partitions. Create user accounts.

3. Tweak the menu.lst file of GRUB on the first OS, so that it can launch any of the two OS interactively at boot time.
4. Boot OS A and verify that the intended partitions have been mounted as planned. Use "*mount"* command, or *"cat /etc/fstab"* command. Reboot, and repeat with distro B.

Create partitions on P and Q : Use a live CD (e.g. Knoppix), since you do not have any OS installed on the system just now. The command "fdisk -l" will give a clear picture of all the disks and their available partitions. Here is what we would see, if we did a fdisk -l on our system:



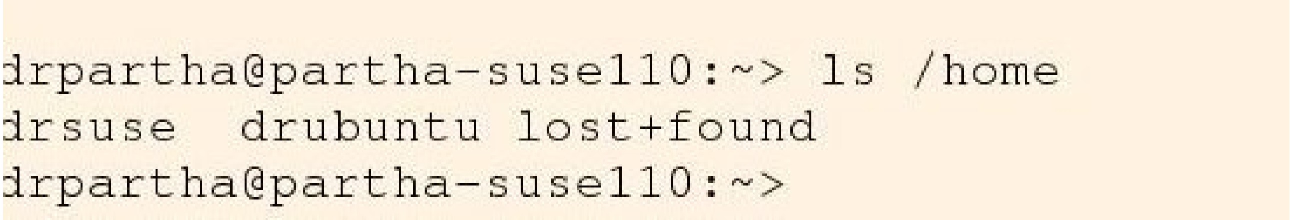
```
partha-suse110:/home/drpartha # fdisk -l

Disk /dev/sda: 40.0 GB, 40020664320 bytes
255 heads, 63 sectors/track, 4865 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Disk identifier: 0x0006ffb7

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1               1        4080    32772568+  83  Linux
/dev/sda2            4081        4463     3076447+  82  Linux swap / Solaris
/dev/sda3            4464        4865     3229065   82  Linux swap / Solaris

Disk /dev/sdb: 80.0 GB, 80026361856 bytes
255 heads, 63 sectors/track, 9729 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Disk identifier: 0x000447c6

   Device Boot      Start         End      Blocks   Id  System
/dev/sdb1   *           1        4831    38804976   83  Linux
/dev/sdb2            4832        9729    39343185   83  Linux

Disk /dev/sdc: 4009 MB, 4009754624 bytes
145 heads, 48 sectors/track, 1125 cylinders
Units = cylinders of 6960 * 512 = 3563520 bytes
Disk identifier: 0xc3072e18

   Device Boot      Start         End      Blocks   Id  System
/dev/sdc1   *           1        1126     3915752    c  W95 FAT32 (LBA)
partha-suse110:/home/drpartha # █
```

Notice that we made just one partition for /home but two partitions for swap. The recommended size of the swap is still a matter of debate and confusion. I chose 2 GB for each swap (on /dev/sda), just to be safe. Each swap partition is reserved for a specific

OS. It is not a good idea to make two OSs write on the same swap partition, because that can cause serious inconsistencies and problems. For instance, every time you shut down Linux properly and subsequently restart it, you get a clean swap partition. If a distro crashes or hangs or is abruptly shutdown, the swap space is not deleted and may lead to inconsistencies in the next reboot. A good article about swap can be found at : `http://www.xenotime.net/linux/doc/swap-mini-howto.txt` .

Install the two OSs : While installing your OS make sure you select the right device/partition and mountpoint for each distro. Do not select “format partition” for the /home partition.

Create user accounts on each OS : Each OS must have its own, distinct, user accounts. That is, you cannot have the same user name on both OS. This is because, when you create a user, the OS adds many hidden files (dot files), to store various configuration data. Such files may get messed up if used with another OS for which they were not made. The /home on my machine looks like this :



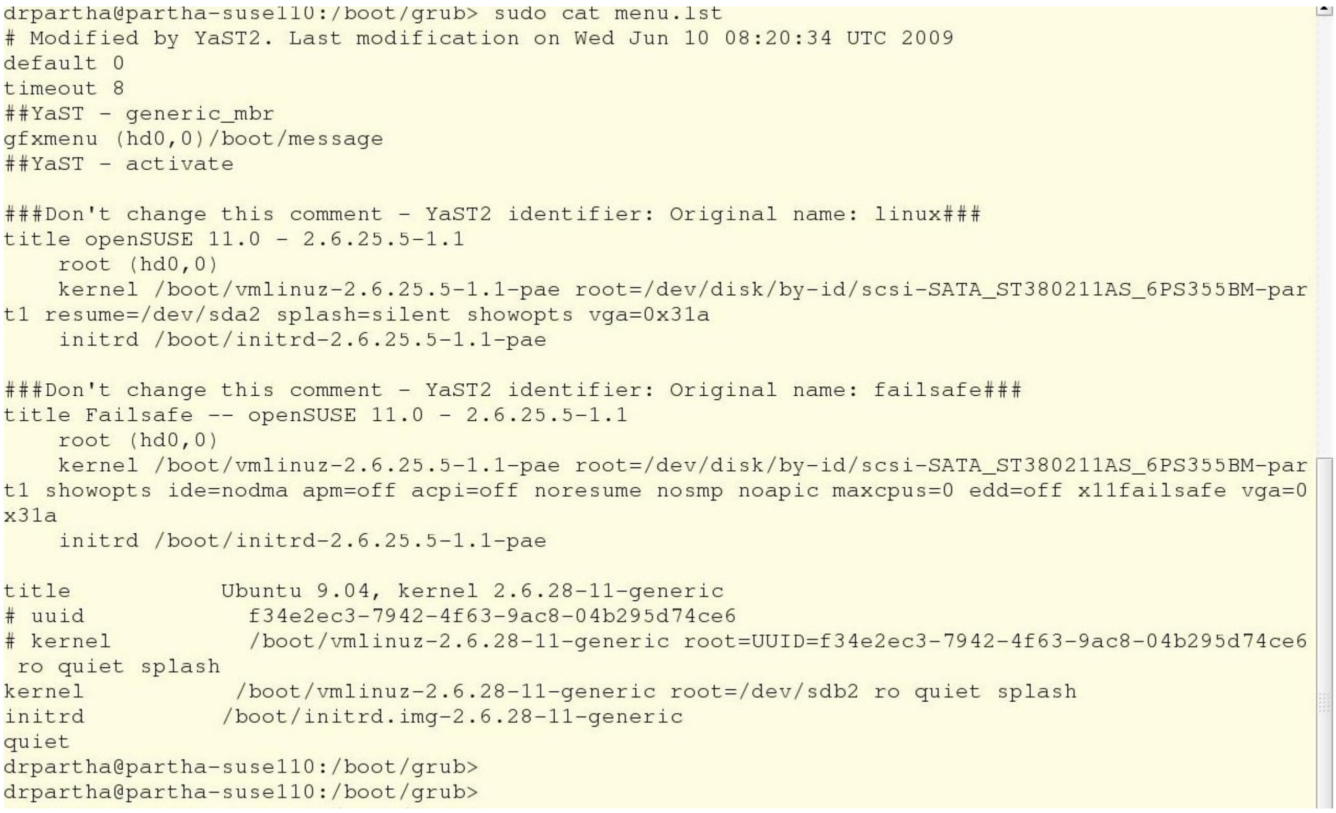
```
drpartha@partha-suse110:~> ls /home
drsuse  drubuntu lost+found
drpartha@partha-suse110:~>
```

Like the names indicate, drsuse is the user on Suse, and drubuntu is the user on ubuntu.

Modify GRUB : Linux uses a boot loader called GRUB (GRand Unified Bootloader), for helping you decide at boot time, which OS should run. The behaviour of GRUB is defined in the file : /boot/grub/menu.lst.

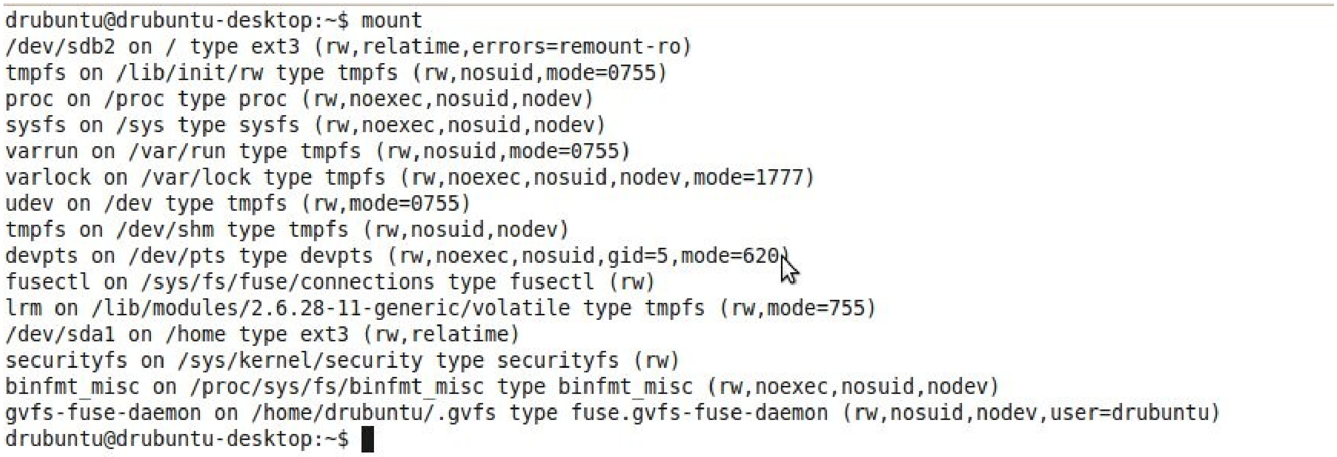
- The first Linux should install GRUB on the MBR
- The second installed Linux should install GRUB on its root partition

Here is a screenshot of the menu.lst file on my first Linux (Suse Linux):



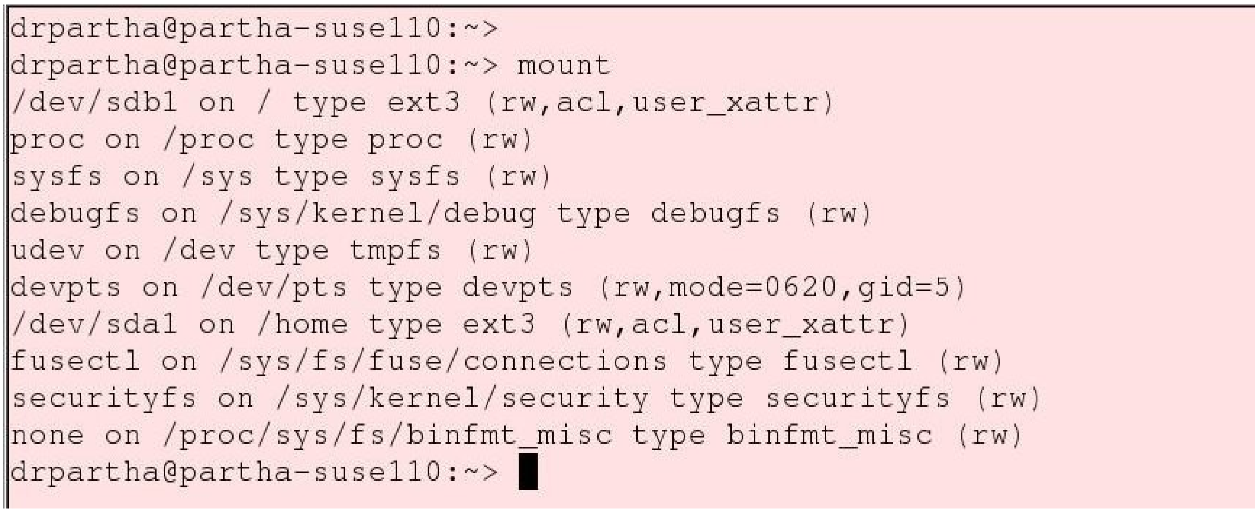
```
drpartha@partha-suse110:/boot/grub> sudo cat menu.lst
# Modified by YaST2. Last modification on Wed Jun 10 08:20:34 UTC 2009
default 0
timeout 8
##YaST - generic_mbr
gfxmenu (hd0,0)/boot/message
##YaST - activate

###Don't change this comment - YaST2 identifier: Original name: linux###
title openSUSE 11.0 - 2.6.25.5-1.1
    root (hd0,0)
    kernel /boot/vmlinuz-2.6.25.5-1.1-pae root=/dev/disk/by-id/scsi-SATA_ST380211AS_6PS355BM-par
t1 resume=/dev/sda2 splash=silent showopts vga=0x31a
    initrd /boot/initrd-2.6.25.5-1.1-pae

###Don't change this comment - YaST2 identifier: Original name: failsafe###
title Failsafe -- openSUSE 11.0 - 2.6.25.5-1.1
    root (hd0,0)
    kernel /boot/vmlinuz-2.6.25.5-1.1-pae root=/dev/disk/by-id/scsi-SATA_ST380211AS_6PS355BM-par
t1 showopts ide=nodma apm=off acpi=off noresume nosmp noapic maxcpus=0 edd=off x11failsafe vga=0
x31a
    initrd /boot/initrd-2.6.25.5-1.1-pae

title           Ubuntu 9.04, kernel 2.6.28-11-generic
# uuid            f34e2ec3-7942-4f63-9ac8-04b295d74ce6
# kernel          /boot/vmlinuz-2.6.28-11-generic root=UUID=f34e2ec3-7942-4f63-9ac8-04b295d74ce6
 ro quiet splash
kernel           /boot/vmlinuz-2.6.28-11-generic root=/dev/sdb2 ro quiet splash
initrd          /boot/initrd.img-2.6.28-11-generic
quiet
drpartha@partha-suse110:/boot/grub>
drpartha@partha-suse110:/boot/grub>
```

Verify using mount or /etc/fstab : The “mount” command is a very useful command. You can call it without any arguments, in which case it will display all the mount points and their respective devices. By examining the mount output of both OS, you can confirm that they are indeed using the partitions as you inteded them to use. Here is a mount output of my second distro (Ubuntu): The file /etc/fstab keeps track of the various filesystems installed on the system. The file /etc/fstab is like the horoscope of your system. Each OS you install will have a /etc/fstab of its own. The /etc/fstab of Ubuntu is shown further below.



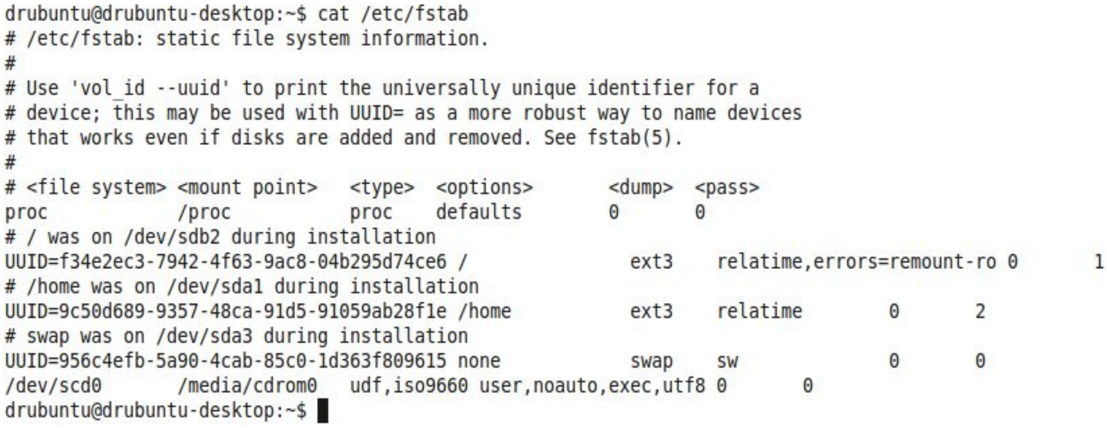
```
drubuntu@drubuntu-desktop:~$ mount
/dev/sdb2 on / type ext3 (rw,relatime,errors=remount-ro)
tmpfs on /lib/init/rw type tmpfs (rw,nosuid,mode=0755)
proc on /proc type proc (rw,noexec,nosuid,nodev)
sysfs on /sys type sysfs (rw,noexec,nosuid,nodev)
varrun on /var/run type tmpfs (rw,nosuid,mode=0755)
varlock on /var/lock type tmpfs (rw,noexec,nosuid,nodev,mode=1777)
udev on /dev type tmpfs (rw,mode=0755)
tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev)
devpts on /dev/pts type devpts (rw,noexec,nosuid,gid=5,mode=620)
fusectl on /sys/fs/fuse/connections type fusectl (rw)
lrm on /lib/modules/2.6.28-11-generic/volatile type tmpfs (rw,mode=755)
/dev/sda1 on /home type ext3 (rw,relatime)
securityfs on /sys/kernel/security type securityfs (rw)
binfmt_misc on /proc/sys/fs/binfmt_misc type binfmt_misc (rw,noexec,nosuid,nodev)
gvfs-fuse-daemon on /home/drubuntu/.gvfs type fuse.gvfs-fuse-daemon (rw,nosuid,nodev,user=drubuntu)
drubuntu@drubuntu-desktop:~$
```

And this is the mount output of my first distro (Suse) :

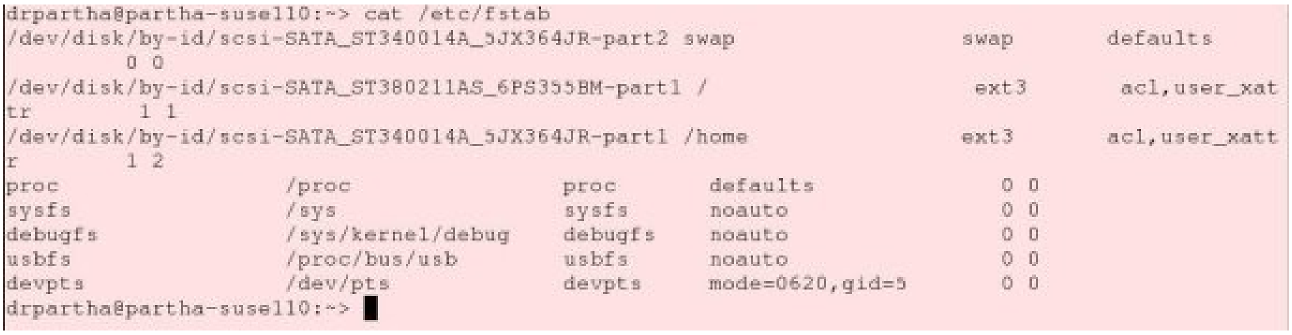


```
drpartha@partha-suse110:~>
drpartha@partha-suse110:~> mount
/dev/sdb1 on / type ext3 (rw,acl,user_xattr)
proc on /proc type proc (rw)
sysfs on /sys type sysfs (rw)
debugfs on /sys/kernel/debug type debugfs (rw)
udev on /dev type tmpfs (rw)
devpts on /dev/pts type devpts (rw,mode=0620,gid=5)
/dev/sda1 on /home type ext3 (rw,acl,user_xattr)
fusectl on /sys/fs/fuse/connections type fusectl (rw)
securityfs on /sys/kernel/security type securityfs (rw)
none on /proc/sys/fs/binfmt_misc type binfmt_misc (rw)
drpartha@partha-suse110:~> █
```

Notice that on both distros /home is mounted on /dev/sda1. The Suse / is at /dev/sdb1 and the Ubuntu / is on /dev/sdb2. Each distro has been given about 40 GB.



```
drubuntu@drubuntu-desktop:~$ cat /etc/fstab
# /etc/fstab: static file system information.
#
# Use 'vol_id --uuid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point>   <type>  <options>       <dump>  <pass>
proc            /proc           proc    defaults        0       0
# / was on /dev/sdb2 during installation
UUID=f34e2ec3-7942-4f63-9ac8-04b295d74ce6 /               ext3    relatime,errors=remount-ro 0       1
# /home was on /dev/sda1 during installation
UUID=9c50d689-9357-48ca-91d5-91059ab28f1e /home           ext3    relatime        0       2
# swap was on /dev/sda3 during installation
UUID=956c4efb-5a90-4cab-85c0-1d363f809615 none            swap    sw              0       0
/dev/scd0       /media/cdrom0   udf,iso9660 user,noauto,exec,utf8 0       0
drubuntu@drubuntu-desktop:~$ █
```



```
drpartha@partha-suse110:~> cat /etc/fstab
/dev/disk/by-id/scsi-SATA_ST340014A_5JX364JR-part2 swap                 swap       defaults
        0 0
/dev/disk/by-id/scsi-SATA_ST380211AS_6PS355BM-part1 /                    ext3       acl,user_xat
tr        1 1
/dev/disk/by-id/scsi-SATA_ST340014A_5JX364JR-part1 /home                ext3       acl,user_xatt
r        1 2
proc                 /proc                proc       defaults              0 0
sysfs                /sys                 sysfs      noauto                0 0
debugfs              /sys/kernel/debug    debugfs    noauto                0 0
usbfs                /proc/bus/usb        usbfs      noauto                0 0
devpts               /dev/pts             devpts     mode=0620,gid=5       0 0
drpartha@partha-suse110:~> █
```

We notice that the swap of Ubuntu is /dev/sda3 and the swap of Suse is /dev/sda2

Why to do this ?

It may sound strange to install two distributions of Linux on the same machine, and have two disks where one would do. There are situations when this is meaningful. For instance, one would like to be familiar with the finer details of various Linux distributions. A good Linux professional has to be conversant with various distributions. We can get confidence only by actually working on them. It may be a good idea, or even the only way out, to have both distros on the same machine. We can thus switch distros, without physically switching between machines. Each distribution has its own package management mechanism. For instance, Suse uses YaST, whereas Ubuntu and Debian look alikes use APT. Each distribution has its favourite desktop interface. On Suse, I use KDE, and on Ubuntu, I use the Gnome interface. These two interfaces are so different from each other. The internal structure, the file hierarchy, the startup and shutdown sequences and the various configuration parameters are all different between distros. All distros do not carry the same set of applications. It is also practical to keep one distro as an experimental distro, on which we can try our several “what-if” scenarios. And, keep one distro as a stable operational distro, for serious usage. So, it makes sense to have two distros on the same machine. On the other hand, keeping two hard disks is also useful. It would help maintain some kind of a backup mechanism, or a mirror, for fault-tolerant operations. The more ambitious user can also configure a RAID if he had two drives (conditions apply).

All Linux users know that /home is the directory where all users' files are stored. Each user gets a sub-directory under this /home directory. All his/her files are stored inside this sub-directory and very well protected from other users. In addition, there are several configuration files specific to each user (usually files with names starting with a period “.”), all of which are stored in the respective user directories. Each distro needs a /home for its users. But it makes sense to use a single /home shareable by both distros. You save space, management is easier, taking backups is faster. By putting the /home on a separate partition (common to both distros), you obtain some more benefits. You can reinstall or change the distros at will, without risking the /home. Disk access becomes faster since accesses to /home do not clash with accesses to other parts of the distro, and swap operations do not compete with other operations on the /home.

In a multitasking OS like Unix/Linux, your disk is constantly under pressure, to serve two competing and concurrent phenomena : process state transitions, and swapping (paging). .

> *Process state transitions :* In a multi-tasking OS, the OS keeps juggling between different tasks (processes), giving the user a feeling that all programs are running simultaneously. Processes are shuffled automatically and frequently between the main memory (RAM) and disk and vice-versa. Remember, this phenomenon can occur not just because of your own applications, it is also provoked by other background processes which are running in the background. If you do a "*pstree*" or a "*ps -ef*" command, you will be surprised to notice the amazing number of processes which are running. There is considerable disk activity going on because of this phenomenon.
>
> *Swapping (paging) :* At the same time, the disk is also required for another important function called "virtual memory". When the system is working on large chunks of data, or data which require huge disk space, the available RAM may not be adequate. This can occur in situations, like when you are trying to invert a very large matrix, say a 10000 x 10000 real matrix. You may be working with image processing applications which handle large images. If your RAM is not enough, the OS automagically and cleverly moves unused/unneeded data out to the hard disk, and brings this data back to RAM whenever needed. The data is temporarily parked on the disk, in a special area appropriately called the "swap area" (or swap).

Your system performance can be considerably reduced if the above two phenomena are too frequent and occur simultaneously and on the same disk. You will notice long periods of silence from your system as it recovers from the above two competing activities. In the arrangement explained in Figure 1, since the swap is on a different drive, access to the swap will not be affected by access to the disk due to process switching (and vice-versa).

Your system will appear to run faster !

Wrap-up

The author invites comments, queries, and suggestions from readers of this article. The author thanks his friends, P.S. Prasad and Suresh Ramasubramanian, for helping him out when he got into dead-ends in this experiment.

About the author

Parthasarathy is an aggressive supporter of FOSS. He teaches discrete mathematics, and preaches LaTeX and Linux, to students of Computer Science. His website : `http://algolog.tripod.com/nupartha.htm` will give more specific details about him. His contact address is : `drpartha@gmail.com`

***