11 - Linux Filesystem

- Mounting of partitions and `/etc/fstab`
- Description of Kernel and its only hook "**/**"
- Description of Kernel Virtual file system (diagram)
- Description of */proc* and */dev* as windows to kernel and devices
- Description of *Journaling Filesystem eg. reiserfs, ext3,xfs,jfs*
- Description of `/lib` and the programming libraries
- Description of the File Hierarchy Standard.

	shareable	unshareable
static	/usr /opt	/etc /boot
variable	/var/mail /var/spool/news	/var/run /var/lock

- Linux File System Structure Standard (FHS)
  `http://www.pathname.com/fh/1.2/fsstnd-toc.html`
- Directories absolutely needed to boot:
  - `/boot`(interne or externe of root dir), `/etc`, `/bin`, `/sbin`, `/proc`(empty), `/dev`, `/lib`, `/tmp`.

- **Description of:**

`mkfs -t ext2`	(create an ext2 filesystem)
`e2fsck`	(check an ext2 filesystem)
`fdisk -l`	(list all partitions of PC)
`df -h`	(list free/used space on mounted partitions)
`du -sh /root`	(display amount of space used by /root dir.)
`df -i /dev/hda3`	(show amount of free inodes on `hda3`)

- **Description and source of the directory structure UNIX V**

  - **/**(root) , `root` user and **`/root`** directory clarification
  - Mounting points principle

- **Description of every content of standard Linux directories**
  see other pages for decription of dir. contents

- **Good programs to manage the file system**

- `mc`	- very good console program
- `konqueror`	- Part of KDE
- `nautilus`	- Part of GNOME
- `gmc`	- Part of GNOME
- `knc`	- Part of KDE-1
- `krusader`	- x-Program
- `xfm`	- X-windows prg. from ‚xap' packages on CD
- `kfm`	- X-windows KDE standard file manager

- **Disk ‚free space' programs**
  - `df` - standard UNIX command
  - `xdf` - X-Windows program
- **Directories used space program**
  - `du -sh /root`
  - `watch -n5 "du -sh /var"`
- **Kill all processes that uses a mounted filesystem to free it up before unmounting.**
  - `fuser -km /dev/hda5`
- **Convert an `ext2` partition to `ext3` Journaling filesystem.**
  The follwoing command can be issued for either mounted or unmounted partition: eg. `/dev/hda5`
  `tune2fs -j /dev/hda5`
  After issuing this command:
  1. If the partition was mounted then the `.journal` file will be created in the root directory of the partition. This file will be made hidden on next boot.
  2. If the partition was not mounted then a hidden journalling file will be created.
  Note: Remember to change the `/etc/fstab` to coincide with the new filesystem format for this partition.
- **To format a partition as ext3 filesystem**(witn bad blocks checking)**:**
  `mkfs -c -t ext3 /dev/hda5`

- **To display** all existing and recognized **Hard Disks** in the system:
  `fdisl -l`

- **Partitionen und Mountpoints**(german)

Verwendetet man viele Partitionen so verliert man eventuell den Ueberblick darueber, welche Partition zu welchem Mount Point gehoert. (Spaetestens dann, wenn man von seiner Notfalldiskette booten muss und die Eintrage in der fstab nicht vorhanden sind.) Durch das setzen des Volume Labels auf den Namen des Mountpoints gelingt kann man die Zuordnung erleichtern. Das setzten des Labels erfolgt ueber das Kommando:

`tune2fs -L <Name Partition>`

Das aktuelle Label kann man z.B. mit

`tune2fs -l <Partition>`

erfragen. Neuere Versionen des "mount" Kommandos verstehen auch den Parameter "**-L**" der es ermoeglicht eine Partition unter Angabe ihres Labels zu mounten.
Beispiel:
`mount -L <label> <Mountpoint>`

Tech Tip for ext2 file system

After an `ext2` file system has been mounted n number of times, an `fsck` is forced on the next mount. Typically n is set to `20`.

If you only mount at each reboot, it is typical to have a situation where for 20 reboots the `fsck` is skipped, but then at the 21st reboot, all the file systems are checked. The long wait on this 21st reboot gets irksome.

```
dumpe2fs /dev/hda7 | grep '[mM]ount count'
    dumpe2fs 1.19, 13-Jul-2000 for EXT2 FS 0.5b, 95/08/09
    Mount count:              7
    Maximum mount count:      20
```

This says that `/dev/hda7` has been mounted seven times since the last `fsck`, and the `fsck` will be skipped for 20 mounts.

If all your file systems have the same `Mount count' then they will all `fsck` together at the 20th reboot. It's easy to fix that. Say:

```
umount /dev/hda6
tune2fs -C 9 /dev/hda6
   tune2fs 1.19, 13-Jul-2000 for EXT2 FS 0.5b, 95/08/09
  Setting current mount count to 9
mount /dev/hda6
```

This changes the `Mount count' to 9.

WARNING WARNING:
ONLY RUN tune2fs ON AN UNMOUNTED FILESYSTEM.
`tune2fs` goes ahead and does it's job even if the file system is mounted.
Isuspect this is dangerous, so it's up to YOU to be careful.

Suppose you have four file systems. Then set the mount count to 1,5,10 and 15. This evenly spaces out the fsck for each of them.

By doing this, the expected cost of reboot remains the same. It's just the variance of boot time that's brought down. Hence this only helps.

The extent to which you like this better depends on how impatient you are; but it is strictly superior to the default arrangement.

If you're paranoid and like to have lots of `fscks`, or if you have more than 20 file systems, the `Maximum mount count' can also be changed by saying `tune2fs -c N`. The value N=-1 disables this feature. You should also know that `tune2fs -i 2` says "check every two days". This may be useful for machines such as notebooks.