File system Fundamentals

Main goals

- Understand basic concepts of filesystems, such as performance tuning and availability improvement.

- Partition management (format, mount, umount, join resize…)

- Extending File System size through logical volumes.

- Error protection (RAID systems) and recovery (Backup tools)

Session 1

1. **Create an initial snapshot for this part**. Next, add in the configuration of your virtual machine in VirtualBOX, a new disk to the SATA controller with 1GB of capacity.

2. Find out which inode number corresponds to the root (/) directory and which block number stores its content. Dump the content of the block to take a look at it[1]. Check if the i-node corresponding to etc directory matches with the info obtained with ls/stat command.

3. In your $HOME directory, create a text file (personal-info.txt) and write your personal information (Name, phone number and e-mail). Find and write down the i-node number for that new file and the extents associated to it (disk blocks storing the data). Now, delete the file (rm) and check which information of that file remains after deletion (i-node entry, directory block, data blocks, …).

4. Create, in your $HOME directory, a 512Mb file with random content (named random.txt). Once created, we are going to move this file to the directory "/home/alumno" in two different ways, and we want to measure the time consumed to perform both. The first solution consists of copying random.txt content to the new location, and then deleting the source file random.txt[2]. The second solution consists of directly using the command mv. Look at the time[3] consumed by both operations and try to explain the differences observed.

5. Create, in the new disk, 5 partitions of 200MB each and format them. One of them will have an **ext3** filesystem, other **reiserfs**[4], other **xfs**[5] and the remaining two **ext2** and **ext4** respectively**.** Mount the partitions in the directories "/disco{x}", where x=1...5. Check the real capacity of each partition after

---

[1] Take a look at this link to understand how directory information is encode in the block: https://www.kernel.org/doc/html/latest/filesystems/ext4/directory.html.

[2] Concatenate copy and remove commands in the same line.

[3] Use the command time for this operation

[4] install reiserfs software with command: apt install reiserfsprogs

[5] install xfs software with command: apt install xfsprogs

the creation & mount process. Is it 200MB? does every partition has the same capacity? Being every Filesystem empty, why do you think each one of them has a different %Used value?

Session 2

6. mount the last partition permanently in the directory "/home/alumno". Perform this task ensuring that user alumno can still access to its $HOME content (all the files that were in /home/alumno before mounting must be still there after this process)[6]. Through a second terminal, login as test user and try to exhaust FS resources for your assigned partition.

    a. First, create as many small files as possible until "something" breaks up. At this point check out how many files you have created and the capacity available in your filesystem. Find out why you are not allowed to create new files.

    b. Second, create a single file able to grow until reaching fs capacity. Once reached this point check out if any storage is available for the administrator (try to create a small file as root). Why is this possible? How much storage is left for root?

7. Join the first three partitions in a unique partition of ext3 type (the same way as we formatted before and without creating a new filesystem, extending the previously created in the first partition)[7]. Copy the content of the /var directory to the new partition you have just created[8]. Once this is done, resize the partition again, so the space in the partition is the smallest possible.

8. Mount the partition permanently in the directory /var. Reboot and check if your /var partition works properly[9]. Correct the problem.

9. Join the last two partitions in a unique partition and create an encrypted partition of LUKS type. That partition must be configured to include the /home directory content (similar to section 7). Add the required entry in the fstab file for automatically mounting your encrypted /home directory[10]. Try to measure the time overhead required to create a 128Mb file in an encrypted FS.

---

[6] Login from another terminal to check if the user works properly.

[7] Try to extend the existing ext3 filesystem to the whole capacity of the new partition.

[8] When copying complete directories, using the "-a" option of cp command is recommended.

[9] Try to install a new package with apt.

[10] Find information associated to an encrypted volume in fstab, you may need adding an entry in /etc/cryptab