LAB 01 - LINUX BACKUP

Adrien Barth, Jeremy Zerbib

TASK 1: PREPARE THE BACKUP DISK

1.

Which disks and which partitions on these disks are visible?

`/dev/sda` and `/dev/sda1` are visible as `ll /dev/hd* /dev/sd*`.

As for the `hd*` part, `ll` could not ready anything as there is no disk in IDE plugged in.

```
ls: cannot access '/dev/hd*': No such file or directory
brw-rw---- 1 root disk 8, 0 Sep 25 22:02  /dev/sda
brw-rw---- 1 root disk 8, 1 Sep 25 22:02  /dev/sda1
```

Which partitions are mounted? Use the command `mount` without parameters to find out.

```
root@ubuntu:/home/adrien# mount | grep /dev/sd
/dev/sda1 on / type ext4 (rw,relatime,errors=remount-ro)
```

2.

Which new files appeared?

`dev/sdb/`

3.

```
root@ubuntu:/home/adrien# parted /dev/sdb
GNU Parted 3.2
Using /dev/sdb
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted) print
Error: /dev/sdb: unrecognised disk label
Model: VMware, VMware Virtual S (scsi)
Disk /dev/sdb: 21.5GB
Sector size (logical/physical): 512B/512B
Partition Table: unknown
Disk Flags:
(parted) mktable
New disk label type? msdos
(parted) print free
Model: VMware, VMware Virtual S (scsi)
Disk /dev/sdb: 21.5GB
```

```
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:

Number  Start   End     Size    Type  File system  Flags
        32.3kB  21.5GB  21.5GB        Free Space


(parted) mkpart
Partition type?  primary/extended? primary
File system type?  [ext2]? fat32
Start? 0
End? 10000
Warning: The resulting partition is not properly aligned for best performance.
Ignore/Cancel? ignore
(parted) mkpart
Partition type?  primary/extended? primary
File system type?  [ext2]? ext4
Start? 10001
End? 21000
(parted) q
Information: You may need to update /etc/fstab.
```

4.

```
root@ubuntu:~# mkfs.vfat /dev/sdb1
mkfs.fat 4.1 (2017-01-24)
root@ubuntu:~# mkfs.ext4 /dev/sdb2
mke2fs 1.44.1 (24-Mar-2018)
Creating filesystem with 2685184 4k blocks and 671744 inodes
Filesystem UUID: d9402135-6cbc-4cd3-984a-c2c6482c0094
Superblock backups stored on blocks:
  32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208

Allocating group tables: done
Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done

root@ubuntu:~# mkdir /mnt/backup1 /mnt/backup2
root@ubuntu:~# mount /dev/sdb1 /mnt/backup1
root@ubuntu:~# mount /dev/sdb2 /mnt/backup2
```

5.

```
root@ubuntu:/home/adrien# ll /dev/sd*
brw-rw---- 1 root disk 8,  0 Sep 25 06:37 /dev/sda
brw-rw---- 1 root disk 8,  1 Sep 25 06:37 /dev/sda1
brw-rw---- 1 root disk 8, 16 Sep 25 06:48 /dev/sdb
brw-rw---- 1 root disk 8, 17 Sep 25 06:48 /dev/sdb1
brw-rw---- 1 root disk 8, 18 Sep 25 06:48 /dev/sdb2
```

6.

```
/dev/sdb1        9.4G  8.0K  9.4G   1% /mnt/backup1
/dev/sdb2         11G   41M  9.5G   1% /mnt/backup2
```

TASK 2: PERFORM BACKUPS USING TAR AND ZIP

1.

```
cd /mnt/backup1/
tar -czf backup.tar.gz /home
```

```
zip -r backup.zip /home
```

Do the files in the archive have a relative path so that you can restore them later to any place?

```
tar: Removing leading `/' from member names
```

We can see that during the making of the archive, `tar` removed the leading "/" to make the path relative.

Regarding the `zip` command, it seems that the relative path is stored.

2.

```
cd /mnt/backup1
tar -tf backup.tar.gz
unzip -l backup.zip
```

3.

```
cd /tmp
tar -xf /mnt/backup1/backup.tar.gz
unzip /mnt/backup1/backup.zip -d zip
```

4.

```
find /home/ -newermt "2016-09-23 10:42:33" | tar cz -T - -f
incrementalBackup1.tar.gz
```

TASK 3: BACKUP OF FILE METADATA

Using `tar`

We first started to create a temporary directory and add a file in it using `touch`:

```
cd Desktop/
mkdir test testRestore
cd test
touch test
ls -al #Using this will show the owner of the file
stat test
```

Using the `stat` command, you can see all the metadata you need (last modified, etc.) . Then we created a new user in order to change the ownership of that file.

```
sudo adduser adrien
sudo chown adrien test
```

After the change of ownership, you want to backup and restore the file using the command `tar`

```
ls -al #Using this will show the new owner of the file
stat test
```



```
cd ..
tar -czf /mnt/backup1/test.tar.gz test/
cd testRestore
tar -xf /mnt/backup1/test.tar.gz
stat test/test
```
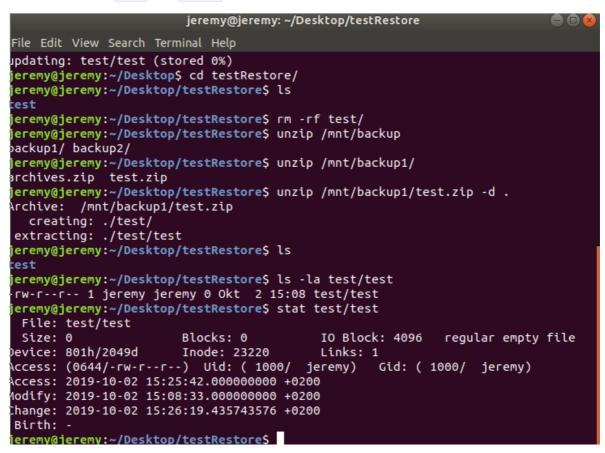
Everything is kept the way it was prior to the backup except the user that is changed to the current owner of the directory `testRestore`.

Using `zip`

The same steps can be repeated for this part and you can see that every metadata is kept the way it was except for the owner that was changed to `backup1`'s owner.

Before executing the `zip` command, we ran the same commands that we did with `tar` and we checked that the `stat` and `ls -la` were the same values as before.

TASK 4: SYMBOLIC AND HARD LINKS

Using `tar`