Stefano Franzoni

mat: 253061

REPORT LAB 5

EXERCISE 1

-- FIRST PART --

> **dd if=/dev/zero of=hd.img bs=512 count=1 seek=$((kbytes*1024))**

dd{disk dump} command create a disk image(hd.img), given the input file(/dev/zero), of kbytes dimension(passed as argument) setted in seek argument, in bs argument was specified the block size. The disk image has a single partition (count=1)

> **losetup /dev/loop1 hd.img**

losetup is used to associate loop devices(/dev/loop1) with disk image (hd.img). in this way i can use all the commands that i uses on devices

```
> cat <<EOF > fdisk.input
x
h
16
s
63
c
EOF
```

fdisk is used to create and manipulate partitions. So we create an fdisk input file giving the commands

cat: in this case is redirected on fdisk.input

x: to enter in expert mode

h: to set the head to 16

s: set the number of sectors/track to 63

```
> cat <<EOF >> fdisk.input
r
n
p
1



a
1
w
EOF
```

n: new partition

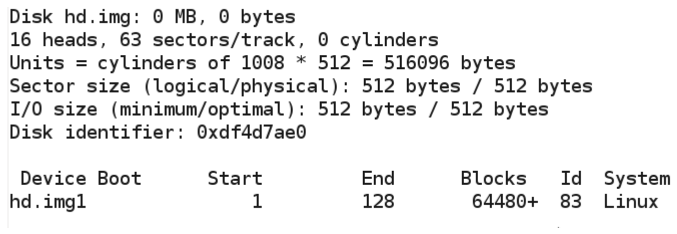
p: partition number 1

a: toggle a bootable flag (so we make bootable the partition number 1)

w: so we write table to disk

> **fdisk hd.img < fdisk.input 1>>pack1.txt 2>>pack2.txt**

After losetup both hd.img and /dev/loop1 are the same thing, so giving as input fdisk.input will produce the effects of this virtual partition. So the file hd.img is like a partition with the features that we are given through fdisk. Redirect the standard output on files pack1.txt and pack2.txt



```
Disk hd.img: 0 MB, 0 bytes
16 heads, 63 sectors/track, 0 cylinders
Units = cylinders of 1008 * 512 = 516096 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0xdf4d7ae0

 Device Boot      Start         End      Blocks   Id  System
hd.img1               1         128       64480+  83  Linux
```

> **fdisk -l -u /dev/loop1 1>>pack1.txt 2>>pack2.txt**

Get the numbers of cylinders

> **blocks=$(fdisk -u -l /dev/loop1|tail -1|tr -s " "|cut -d " " -f 5|cut -d "+" -f 1)**

Take the blocks number

> **losetup -d /dev/loop1**

Detach the /dev/loop1

> **losetup -o $((63*512)) --sizelimit $(($blocks*1024)) /dev/loop0 hd.img 1>>pack1.txt 2>>pack2.txt**

Associate again the /dev/loop0 at hd.img but starting at offset $((63*512)) to leave the space for masterboot record, and we set the sizelimit to $(($blocks*1024))

> **mkfs.ext2 /dev/loop0**

This command is used to create a file system on /dev/loop0 (than on hd.img)

-- SECOND PART --

> **cat <<EOF > menu.lst**

title minimal linux like kernel

kernel /boot/minimal_linux_like_kernel root=/dev/hda ro quiet splash

# initrd /boot/minimal_linux_like_kernel_initrd (not necessary because we don't create a # file system inside the kernel)

quiet

EOF

We create menu list for the GRUB multiboot loader

The kernel is in the directory /boot/minimal_linux_like_kernel

The root is /dev/hda

It is mounted ro(read only)

```
> mkdir mount_point 1>>pack1.txt 2>>pack2.txt
mount /dev/loop0 mount_point 1>>pack1.txt 2>>pack2.txt
```

mount /dev/loop0 to mount_point

```
> mkdir -p mount_point/boot/grub
cp ../grub/stage1 ../grub/stage2 menu.lst mount_point/boot/grub 1>>pack1.txt 2>>pack2.txt
```

Copy grub files(from local hard drive) in the menu list in the directory grub

```
> cp -v src/kernel mount_point/boot/minimal_linux_like_kernel 1>>pack1.txt 2>>pack2.txt
```

Copy the kernel from souce to mount_point/boot/minimal_linux_like_kernel

```
> umount mount_point 1>>pack1.txt 2>>pack2.txt
rm -r mount_point 1>>pack1.txt 2>>pack2.txt
```

unmount the mount_point because we want to unmount the virtual hard disk

and remove everything that was unmount point because it has been copied in the hd.img

```
> losetup -d /dev/loop0  1>>pack1.txt 2>>pack2.txt
```

detach /dev/loop0

```
> cat <<EOF > grub.input
device (hd0,0) hd.img
EOF
echo geometry \(hd0\) $kbytes 16 63 >> grub.input
cat <<EOF >> grub.input
root (hd0,0)
```

```
setup (hd0)
quit
EOF
```

creates the grub.input file for grub

the device is in the hd0(first partition) and the name is hd.img

than we give same geometry parameters

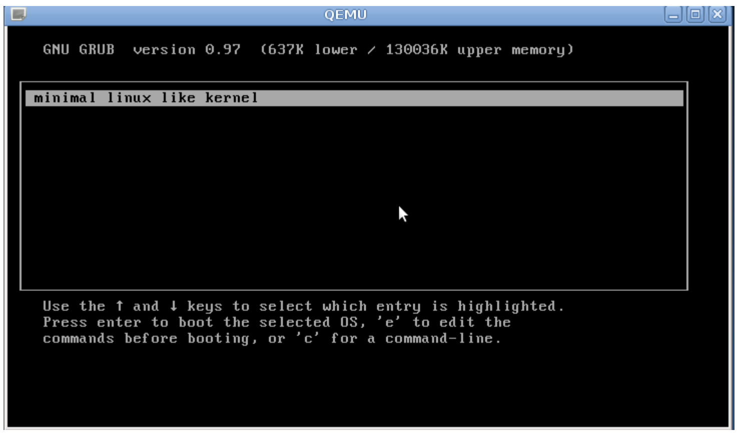
root is in the hd0 and setup to start

```
> ../grub/grub.bin --device-map=/dev/null < grub.input 1>>pack1.txt 2>>pack2.txt
```

Give the input grub.input1 to the grub.bin and copy this two stages of the grub in hd.img so that it can be bootable

```
> chown ${var_user}:${var_group} pack1.txt
chown ${var_user}:${var_group} pack2.txt
chown ${var_user}:${var_group} hd.img
chown ${var_user}:${var_group} menu.lst
```

all this instructions are runned in sudo mode as a superuser so here is chaged the owner



```
GNU GRUB  version 0.97  (637K lower / 130036K upper memory)

 minimal linux like kernel

Use the ↑ and ↓ keys to select which entry is highlighted.
Press enter to boot the selected OS, 'e' to edit the
commands before booting, or 'c' for a command-line.
```