Musab Mehadi

mmehadi@jacobs-university.de

Sheet 11

Problem 1)

A)

In RAID 0, the system fails if one of the discs come to failure as the data is stored in only one of the discs and there is no copy of it on the other discs. Hence, the probability of failure of RAID 0 is equal to the probability of at least one of the n discs failing.

F0(p,n) =1- probability(no disc fails)

=1- ((1-p)^n)

B) In RAID 1, since the data is written on all n discs ,failure happens if both discs fail at the same time. Hence, the probability of failure of RAID 1 is equal to the probability of all n discs failing.

F1(p,n) = p^n

C) Assume we have 6 discs with failure probabilities of 0.1 and hence according to RAID 10 arrangement which gives us n/2 = 3 groups of discs arranged in RAID 1 to be arranged as RAID 0.

In this arrangement ,we lose our data if we lose both discs within the same group.

Hence, then probability of failure is equal to the probability of atleast one group failing.

In this specific case,

A	A
B	B

C	C
D	D

E	E
F	F

Note: Each table represents one group, the groups are arranged together in RAID 0.

- **Within each group the discs (columns in each table) are arranged in RAID 1**

F10(p,n) = 1- probability(no group fails)

probability(no group fails)= (1-(P)^2)^(n/2)

probability(no group fails)=(1-(0.1)^2)^(6/2)

F10(p,n) =1-0.97=0.03=3%

Hence, the general formula is F10(p,n) =1-(1-(P)^2)^(n/2).

D)Again, Assuming we have 6 discs and hence according to RAID 01 arrangement we will have two groups hence each group will have n/2 =3 discs.

We lose our data in this arrangement if BOTH discs carrying the same data from different groups fail.

In this specific case,

A	C	E
B	D	F

A	C	E
B	D	F

Note: Each table represents one group, the groups are arranged together in RAID 1.

- **Within each group the discs (columns in each table) are arranged in RAID 0.**

F01(p,n) = probability(both group fails)

probability(both group fails)= (1-(1-P)^(n/2))^(2)

probability(no group fails)=(1-(1-0.1)^(6/2))^(2)=0.073=7.3%

F10(p,n) =7.3%

Hence, the general equation is,

F01(p,n)=(1-(1-p)^(n/2))^2

In this arrangement ,we lose our data if discs that carry similar data from both groups fail.

5)

Although both are similar in their performance and their storage capacity,I prefer to use the RAID 10 configuration as it has less probability of failure as compared to RAID 01 with the same number of discs and failure rates in the examples shown above.

Problem 2)

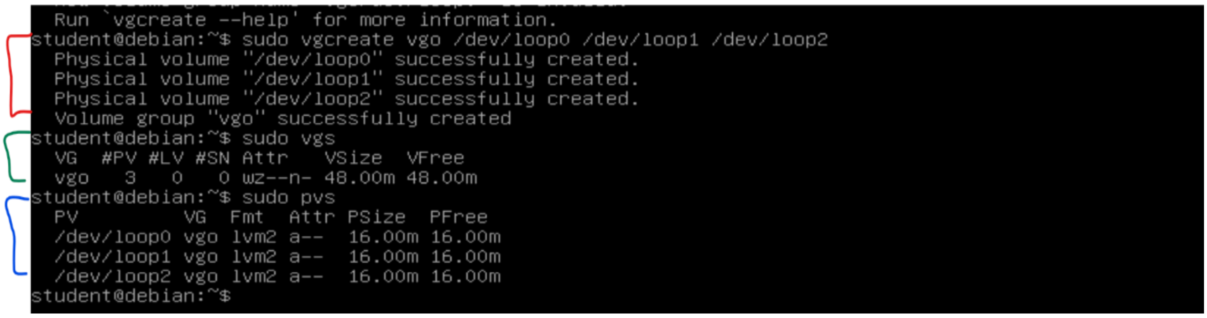
The volume group is created using the command

sudo vgcreate vgo /dev/loop0 /dev/loop1 /dev/loop2

which prints the confirmation of the creation (shown in red)

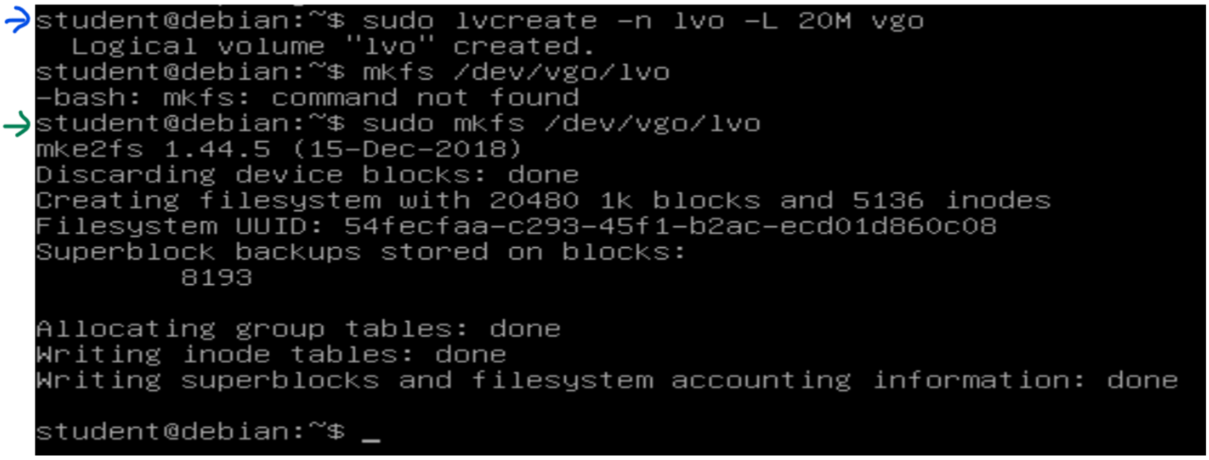
The sudo vgs (in green) command shows the size of the volume group which is 48MB

The sudo pvs command(in blue) shows the size contributions of /dev/loop0,/dev/loop1 and /dev/loop2 which is 16MB each as we can see from PSize. This shows that 4MB of space is used by each loop from the original 20MB



```
  Run `vgcreate --help' for more information.
student@debian:~$ sudo vgcreate vgo /dev/loop0 /dev/loop1 /dev/loop2
  Physical volume "/dev/loop0" successfully created.
  Physical volume "/dev/loop1" successfully created.
  Physical volume "/dev/loop2" successfully created.
  Volume group "vgo" successfully created
student@debian:~$ sudo vgs
  VG   #PV #LV #SN Attr   VSize  VFree
  vgo    3   0   0 wz--n- 48.00m 48.00m
student@debian:~$ sudo pvs
  PV         VG  Fmt  Attr PSize  PFree
  /dev/loop0 vgo lvm2 a--  16.00m 16.00m
  /dev/loop1 vgo lvm2 a--  16.00m 16.00m
  /dev/loop2 vgo lvm2 a--  16.00m 16.00m
student@debian:~$
```

B)



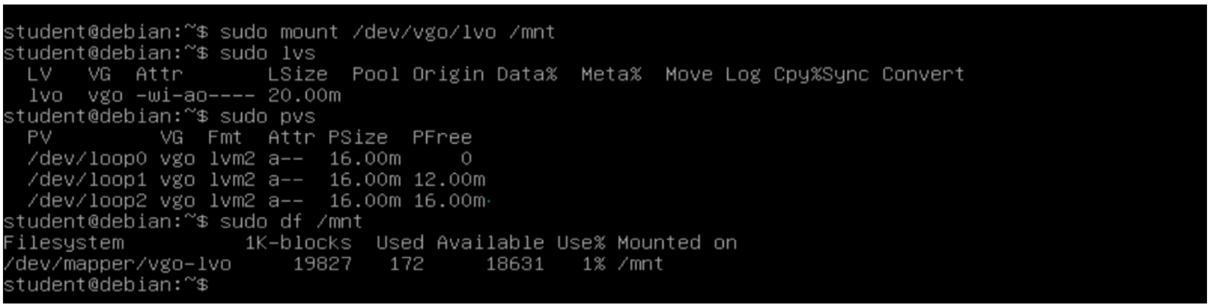
```
student@debian:~$ sudo lvcreate -n lvo -L 20M vgo
  Logical volume "lvo" created.
student@debian:~$ mkfs /dev/vgo/lvo
-bash: mkfs: command not found
student@debian:~$ sudo mkfs /dev/vgo/lvo
mke2fs 1.44.5 (15-Dec-2018)
Discarding device blocks: done
Creating filesystem with 20480 1k blocks and 5136 inodes
Filesystem UUID: 54fecfaa-c293-45f1-b2ac-ecd01d860c08
Superblock backups stored on blocks:
        8193

Allocating group tables: done
Writing inode tables: done
Writing superblocks and filesystem accounting information: done

student@debian:~$ _
```

- From the figure above,the command shown by the blue arrow creates a logical volume of 20MB size.

- The command pointed by the green arrow creates a file system in lvo.



```
student@debian:~$ sudo mount /dev/vgo/lvo /mnt
student@debian:~$ sudo lvs
  LV   VG   Attr       LSize  Pool Origin Data%  Meta%  Move Log Cpy%Sync Convert
  lvo  vgo -wi-ao---- 20.00m
student@debian:~$ sudo pvs
  PV         VG   Fmt  Attr PSize  PFree
  /dev/loop0 vgo lvm2 a--   16.00m     0
  /dev/loop1 vgo lvm2 a--   16.00m 12.00m
  /dev/loop2 vgo lvm2 a--   16.00m 16.00m
student@debian:~$ sudo df /mnt
Filesystem          1K-blocks  Used Available Use% Mounted on
/dev/mapper/vgo-lvo     19827   172     18631   1% /mnt
student@debian:~$
```

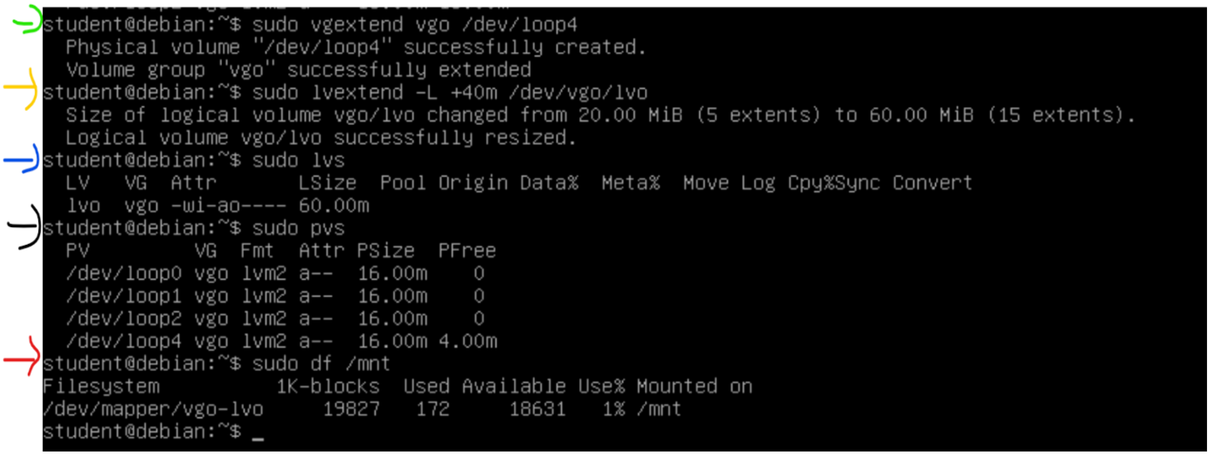
- From the above figure, the command **sudo mount /dev/vgo/lvo /mnt** mounts lvo on it.

-**sudo lvs** shows the size of the logical volume lvo.

- From the outcome of **sudo pvs**, we can see that 16MB and 4MB are used from loop 0 and loop 1 respectively (which make up the volume group) to give space given to lvo.

-sudo df /mnt shows that lvo is mounted on /mnt .

C)



```
student@debian:~$ sudo vgextend vgo /dev/loop4
  Physical volume "/dev/loop4" successfully created.
  Volume group "vgo" successfully extended
student@debian:~$ sudo lvextend -L +40m /dev/vgo/lvo
  Size of logical volume vgo/lvo changed from 20.00 MiB (5 extents) to 60.00 MiB (15 extents).
  Logical volume vgo/lvo successfully resized.
student@debian:~$ sudo lvs
  LV   VG   Attr       LSize  Pool Origin Data%  Meta%  Move Log Cpy%Sync Convert
  lvo  vgo -wi-ao---- 60.00m
student@debian:~$ sudo pvs
  PV         VG   Fmt  Attr PSize  PFree
  /dev/loop0 vgo lvm2 a--  16.00m    0
  /dev/loop1 vgo lvm2 a--  16.00m    0
  /dev/loop2 vgo lvm2 a--  16.00m    0
  /dev/loop4 vgo lvm2 a--  16.00m 4.00m
student@debian:~$ sudo df /mnt
Filesystem          1K-blocks  Used Available Use% Mounted on
/dev/mapper/vgo-lvo     19827   172     18631   1% /mnt
student@debian:~$ _
```

Since we want to grow our file system lvo to 60MB,we will need to add more space from the volume group vgo.However,since vgo is only 48MB big, we will first extend it by adding loop4.Thus, vgo will have enough space to give for lvo.

From the above figure,

-command indicated by the green arrow extends the vgo by adding /dev/ loop4.

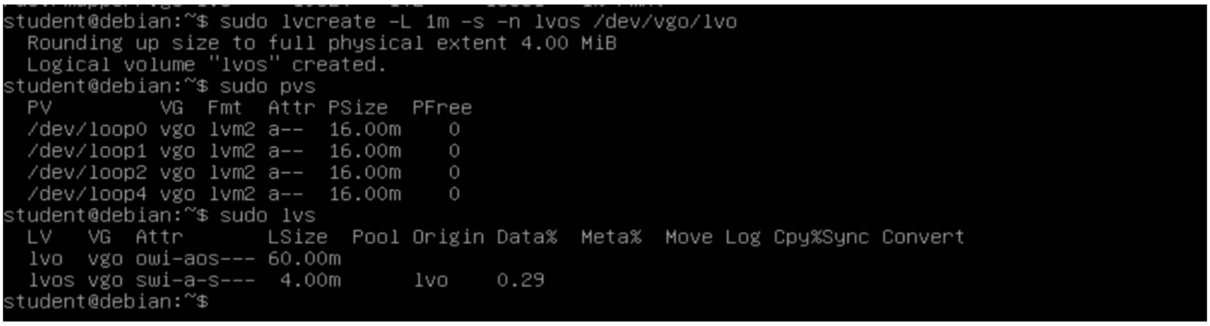
-command indicated by yellow arrow extends the lvo by 40MB.

-command indicated by blue arrow shows size of lvo which is now 60MB.

-command indicated by black arrow shows the size contributions of /dev/ loop0,/dev/ loop1,/dev/ loop2 and /dev/ loop4 (PSize) to lvo and how much is not used .

-command indicated by red arrow shows that lvo is mounted on /mnt .

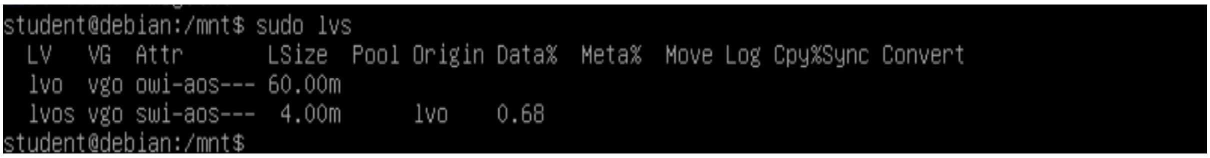
D)



```
student@debian:~$ sudo lvcreate -L 1m -s -n lvos /dev/vgo/lvo
  Rounding up size to full physical extent 4.00 MiB
  Logical volume "lvos" created.
student@debian:~$ sudo pvs
  PV         VG  Fmt  Attr PSize  PFree
  /dev/loop0 vgo lvm2 a--  16.00m    0
  /dev/loop1 vgo lvm2 a--  16.00m    0
  /dev/loop2 vgo lvm2 a--  16.00m    0
  /dev/loop4 vgo lvm2 a--  16.00m    0
student@debian:~$ sudo lvs
  LV   VG  Attr       LSize  Pool Origin Data%  Meta%  Move Log Cpy%Sync Convert
  lvo  vgo owi-aos--- 60.00m
  lvos vgo swi-a-s---  4.00m      lvo    0.29
student@debian:~$
```

-As we can see from the above figure, the **sudo lvcreate -L 1m -s -n lvos /dev/vgo/lvo** command creates the snapshot logical volume lvos of lvo.

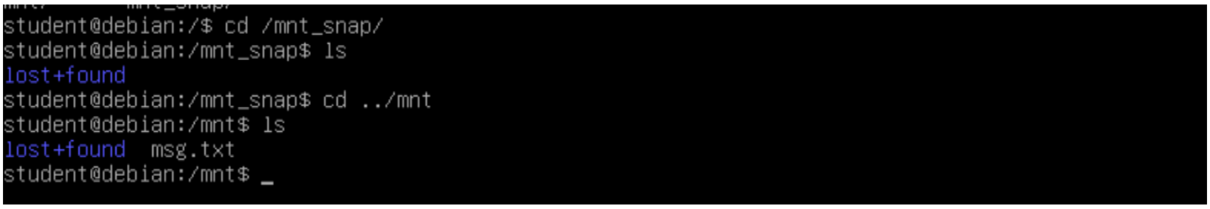
-I created a msg.txt in /mnt and wrote “I need to get full points for Homeworks”.



```
student@debian:/mnt$ sudo lvs
  LV   VG  Attr       LSize  Pool Origin Data%  Meta%  Move Log Cpy%Sync Convert
  lvo  vgo owi-aos--- 60.00m
  lvos vgo swi-aos---  4.00m      lvo    0.68
student@debian:/mnt$
```

- We observe a change in the data% from the above picture from 0.29% to 0.68% after msg.txt was created as shown in the above picture.

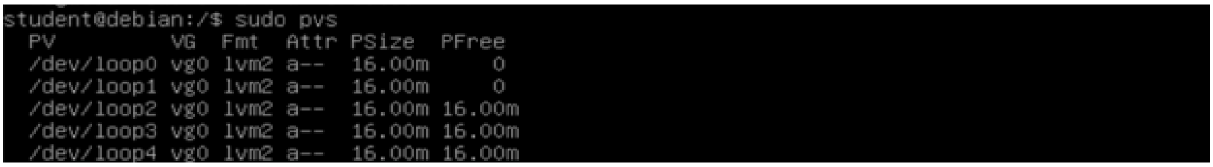
However, msg.txt will not exist in the snapshot file system since the text file in lvo was created after the snapshot lvos was created as we can see from figure below.



```
student@debian:/$ cd /mnt_snap/
student@debian:/mnt_snap$ ls
lost+found
student@debian:/mnt_snap$ cd ../mnt
student@debian:/mnt$ ls
lost+found  msg.txt
student@debian:/mnt$ _
```

E)

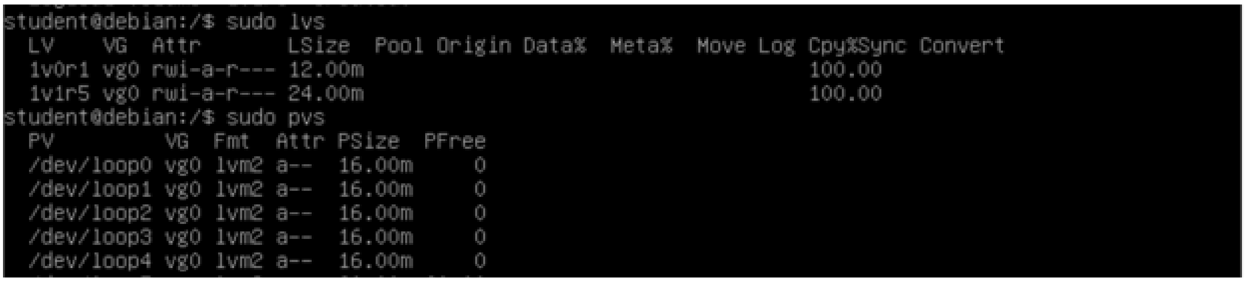
 - As seen from the figure below, loop0 and loop1 are used to store the data that belongs to lv0r1 as PFree is zero.



```
student@debian:/$ sudo pvs
  PV         VG  Fmt  Attr PSize  PFree
  /dev/loop0 vg0 lvm2 a--  16.00m      0
  /dev/loop1 vg0 lvm2 a--  16.00m      0
  /dev/loop2 vg0 lvm2 a--  16.00m 16.00m
  /dev/loop3 vg0 lvm2 a--  16.00m 16.00m
  /dev/loop4 vg0 lvm2 a--  16.00m 16.00m
```

-The output of the **sudo lvs** command shows the size of lv0r1 and lv1r5 which are 12MB and 24MB respectively.

-The physical volume providing storage for lv1r5 are loop 2,3 and 4 as shown from fig below(PFree value went to 0) which is 16+16+16=48MB .



```
student@debian:/$ sudo lvs
  LV     VG  Attr       LSize  Pool Origin Data%  Meta%  Move Log Cpy%Sync Convert
  lv0r1 vg0 rwi-a-r--- 12.00m                                     100.00
  lv1r5 vg0 rwi-a-r--- 24.00m                                     100.00
student@debian:/$ sudo pvs
  PV         VG  Fmt  Attr PSize  PFree
  /dev/loop0 vg0 lvm2 a--  16.00m     0
  /dev/loop1 vg0 lvm2 a--  16.00m     0
  /dev/loop2 vg0 lvm2 a--  16.00m     0
  /dev/loop3 vg0 lvm2 a--  16.00m     0
  /dev/loop4 vg0 lvm2 a--  16.00m     0
```