Kristin Adachi
CS431
2/27/17

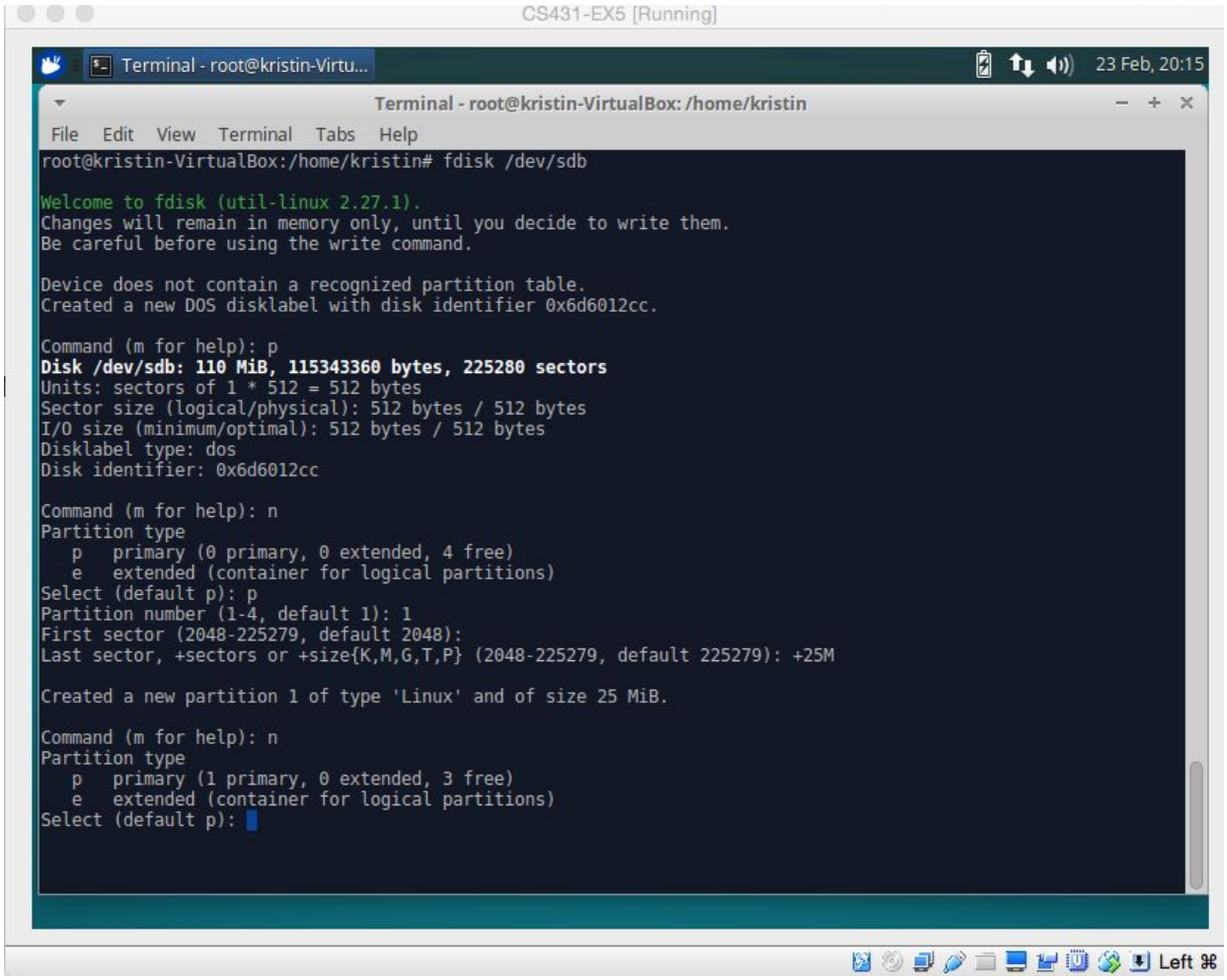
Exercise 5

Adding a Disk:

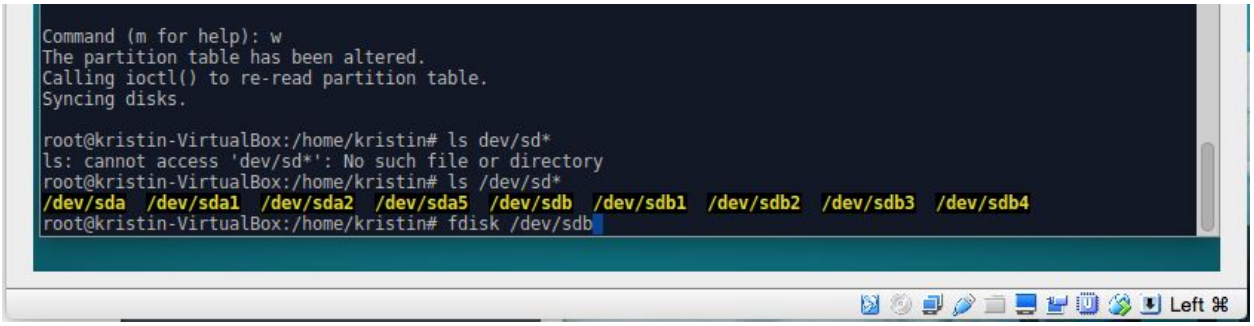
After setting up the virtual machine, I added a disk of size 110MB.

Creating Partitions:

The `fdisk` command was used to create four partitions of size 25MB each on the disk I created.

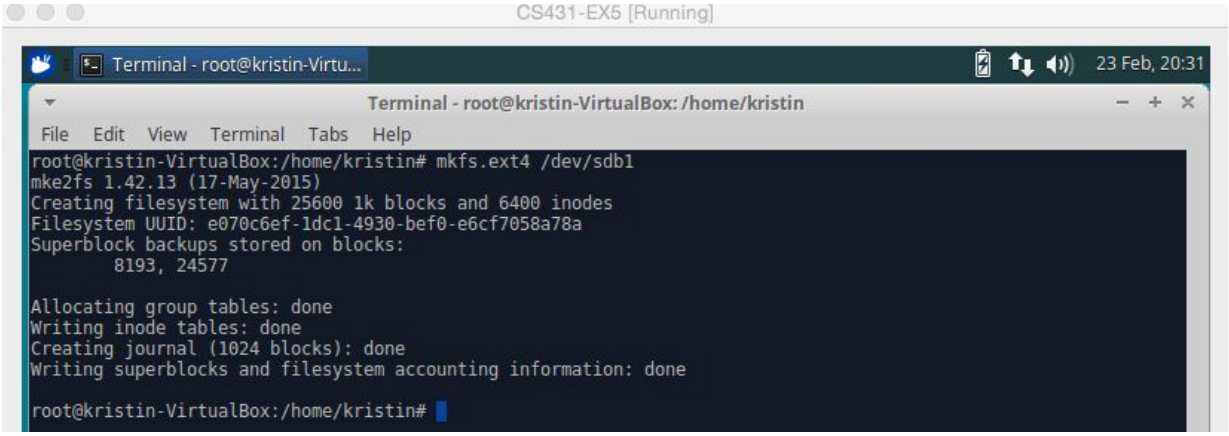


After adding the 4 partitions, the table was written to the disk and the `fdisk` program was exited. After running `ls /dev/sd*` again, it produced this output:

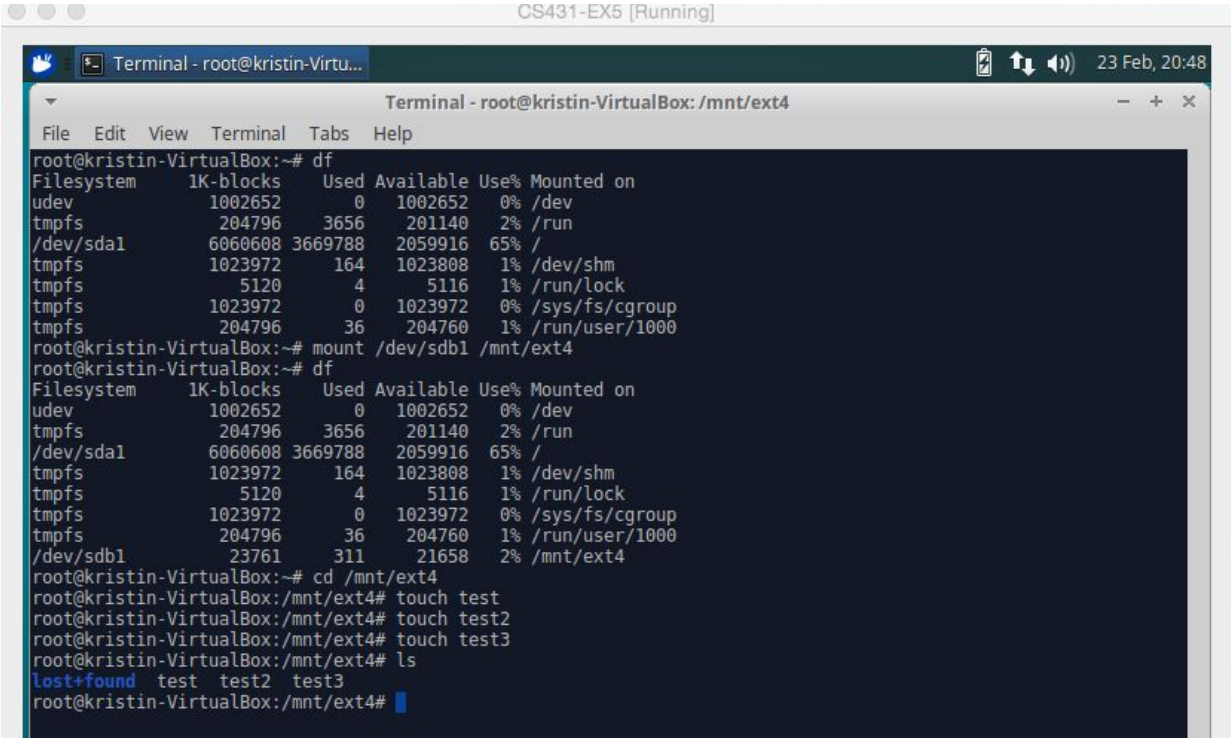


Creating ext4 File System:

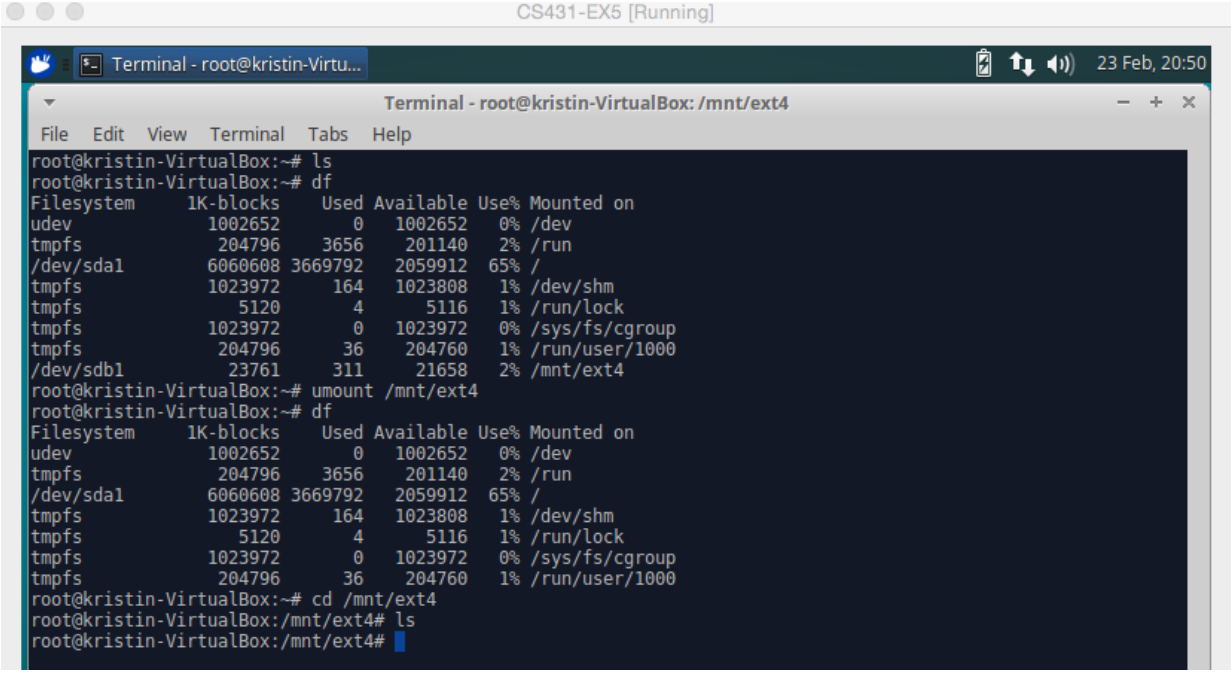
The `mkfs.ext4` command was used to create a new `ext4` file system on `/dev/sdb1`.



After it was created, the file system was mounted onto the OS's directory tree to be accessed. I made a directory called `/mnt/ext4` and used the `mount` command to mount the file system.

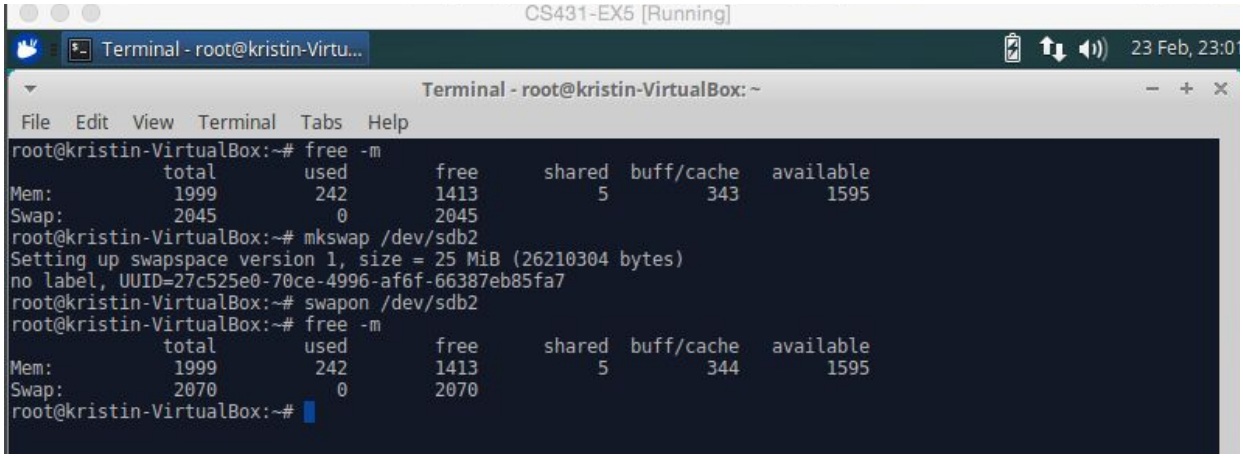


After mounting, I changed directory into the new file system and created files `test`, `test2`, and `test3`. Then, I dismounted the file system using the `umount` command.

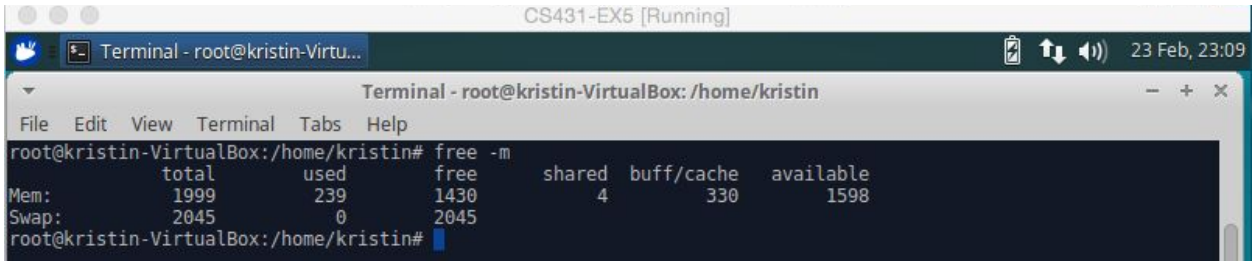


Creating Swap Space:

A swap space was created by running using the `mkswap` command and enabled by using the `swapon` command. I ran `free -m` before and after to show the change in free memory.



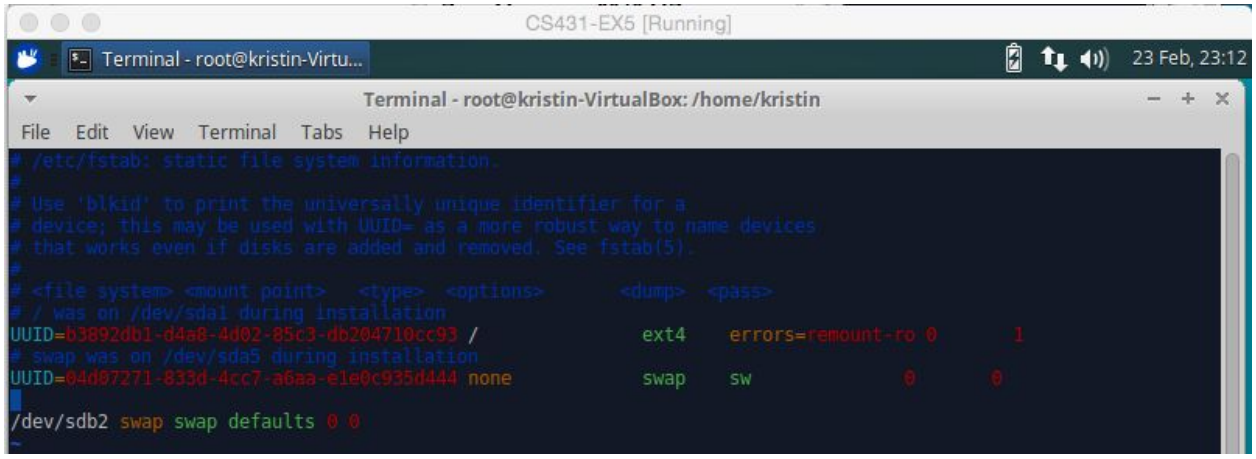
After restarting and running `free -m` again, the swap space is gone.



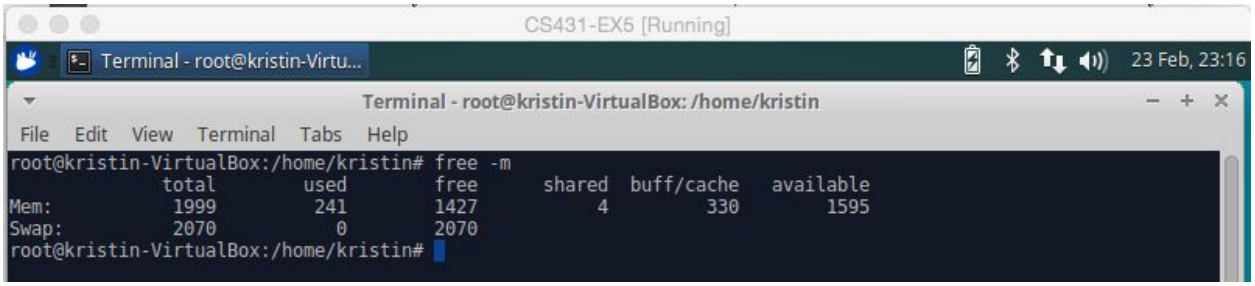
To fix this issue, I added a line to the bottom of the file system table file (`/etc/fstab`):

`/dev/sdb2 swap swap defaults 0 0`

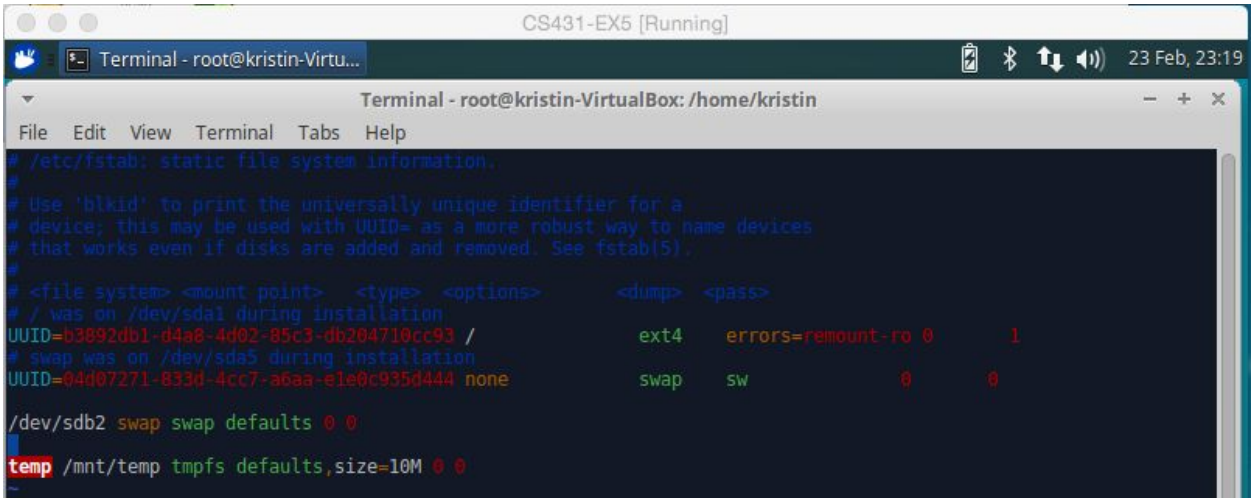
This refers to the file system, mount point, type, options, dump, and pass respectively.



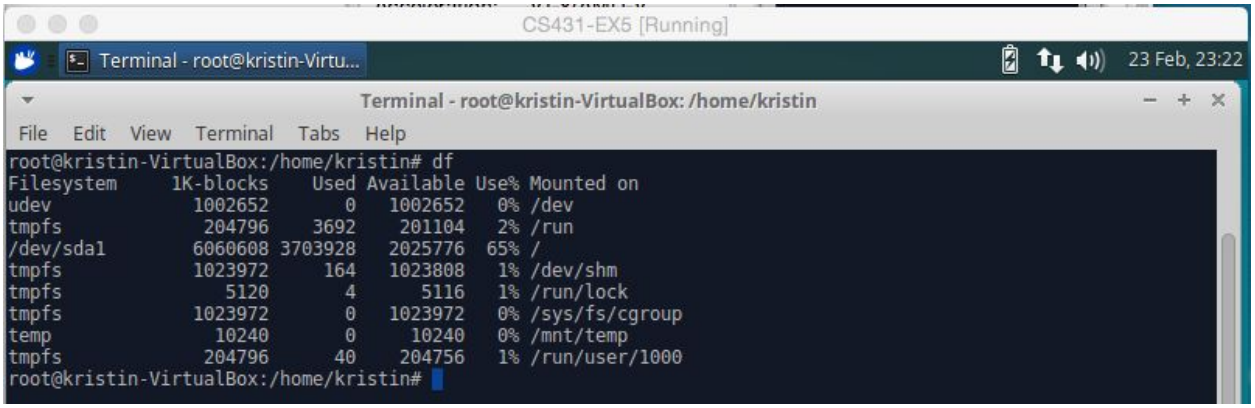
After rebooting and running `free -m` again, I noticed that the swap space remained after the line was added to the file system table file.



I used `fstab` to create a RAM disk by adding an additional line to the file system table file:
`temp /mnt/temp tmpfs defaults,size=10M 0 0`
This creates a RAM disk of size 10MB and mounts at `/mnt/temp` when the system is rebooted.

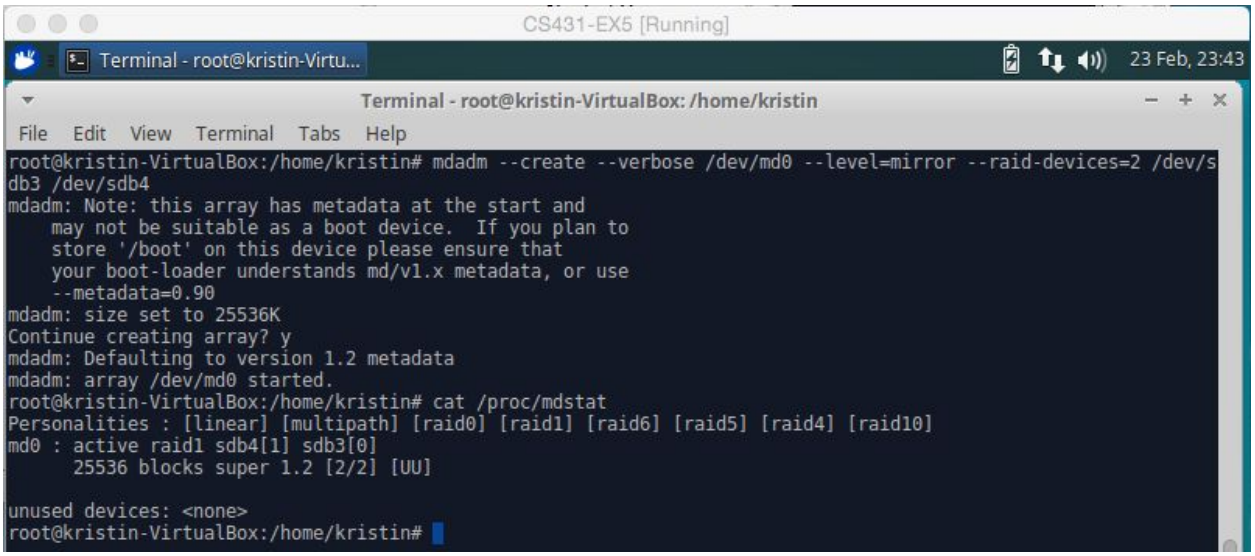


After rebooting, the RAM disk is created and automatically mounted:

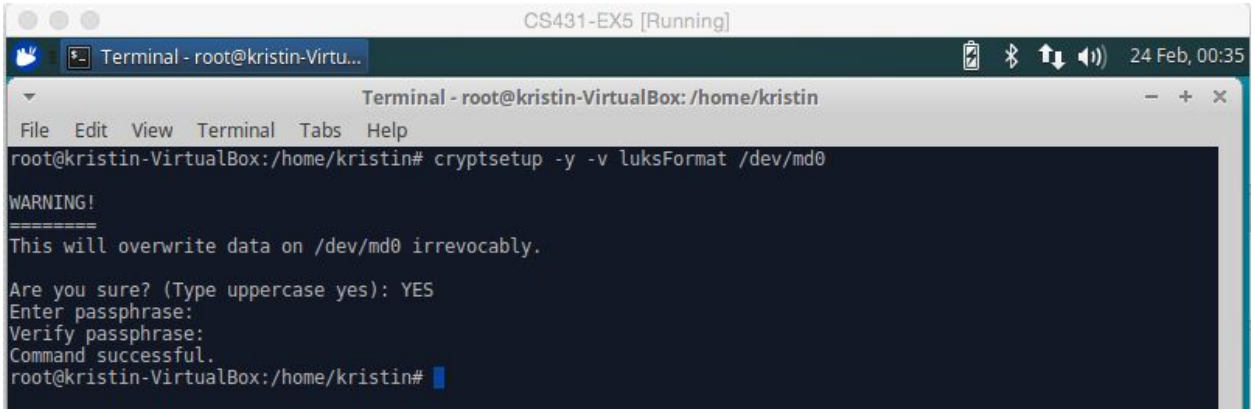


Creating a RAID1 Mirror:

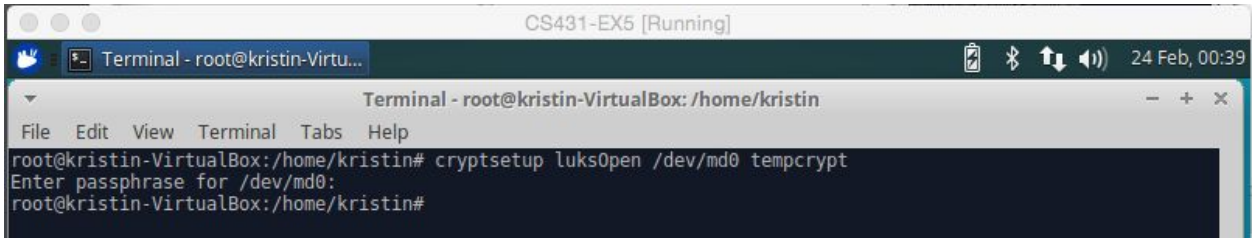
I used the `mdadm` to create a RAID1 mirror. After successfully creating it, I ran the command:
`cat /proc/mdstat`
This outputted the following:



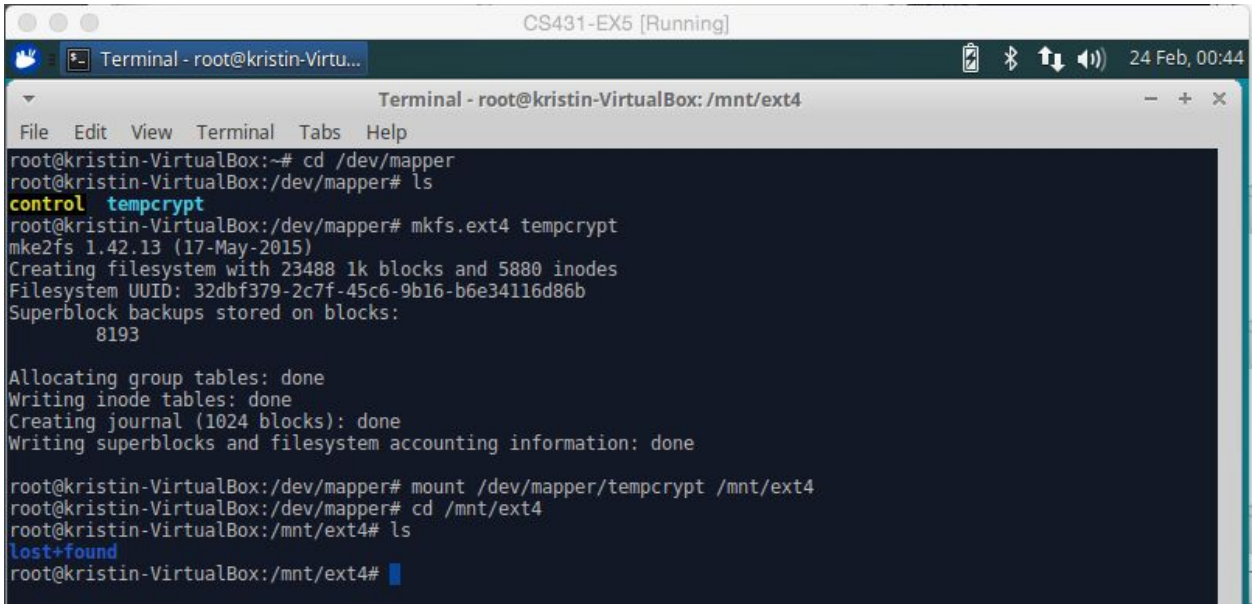
I used the LUKS (Linux Unified Key Setup) to create an encrypted file system on top of this RAID1 device. An encrypted file system was created on `/dev/md0` by using `cryptsetup luksFormat`.



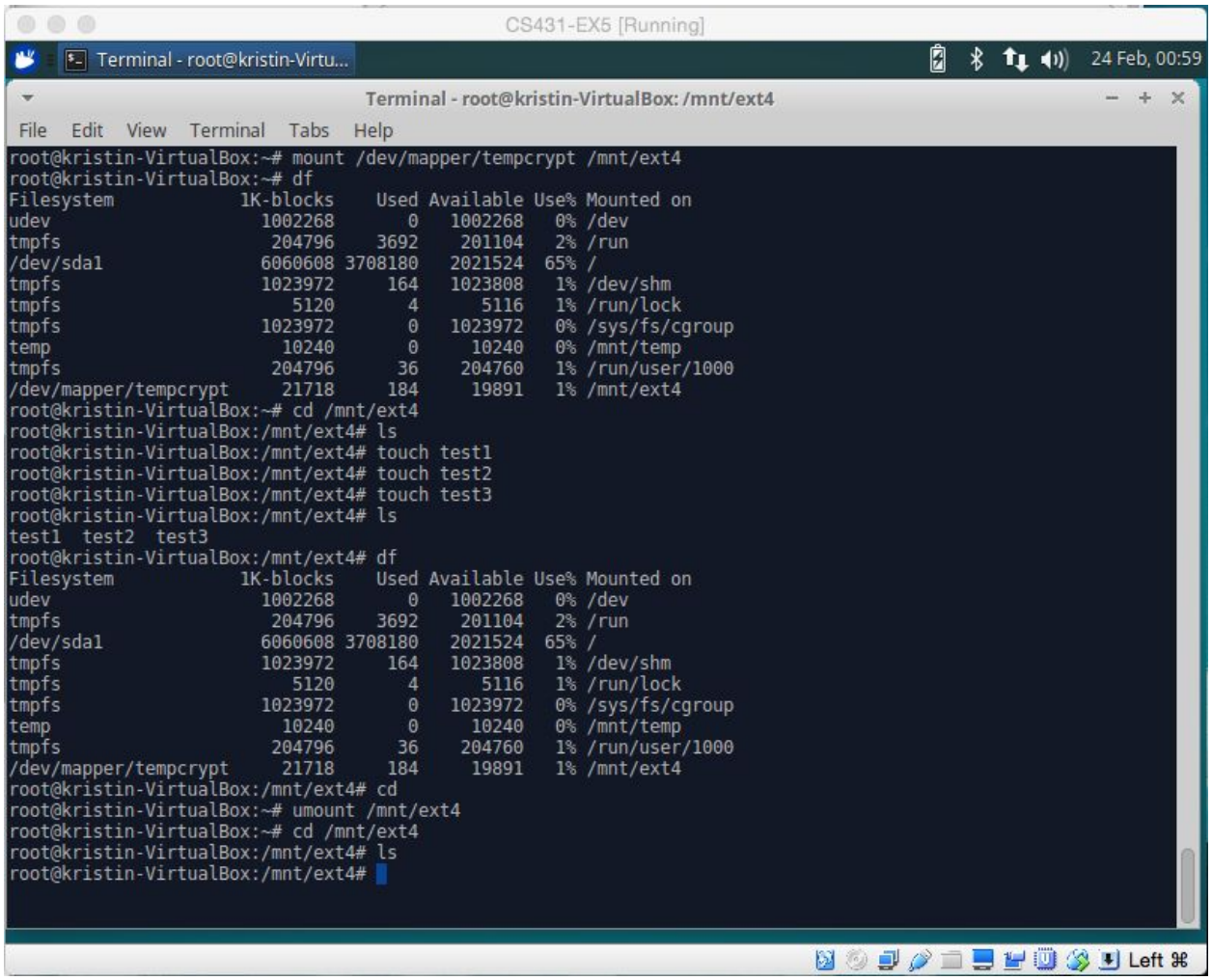
Then, `cryptsetup luksOpen` was used to open the encrypted file system and put a reference to it in `/dev/mapper`. I chose to open it as `tempcrypt`.



I created a file system on `/dev/mapper/tempcrypt` by using `mkfs.ext4` again. After, I mounted the newly created file system using the `mount` command.



Again, I created some files `test1`, `test2`, and `test3` and then dismounted the file system.



Finally, I closed the encrypted file system by using `cryptsetup luksClose` .