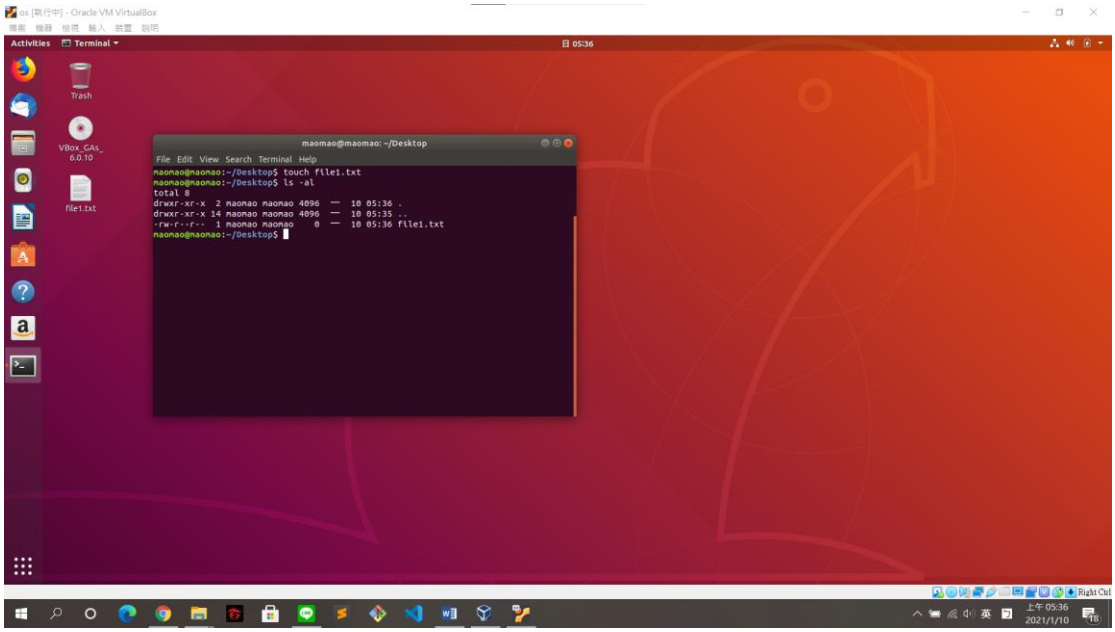
HW3 File system

0616098 黃秉茂

Task1 - Soft Link and Hard Link

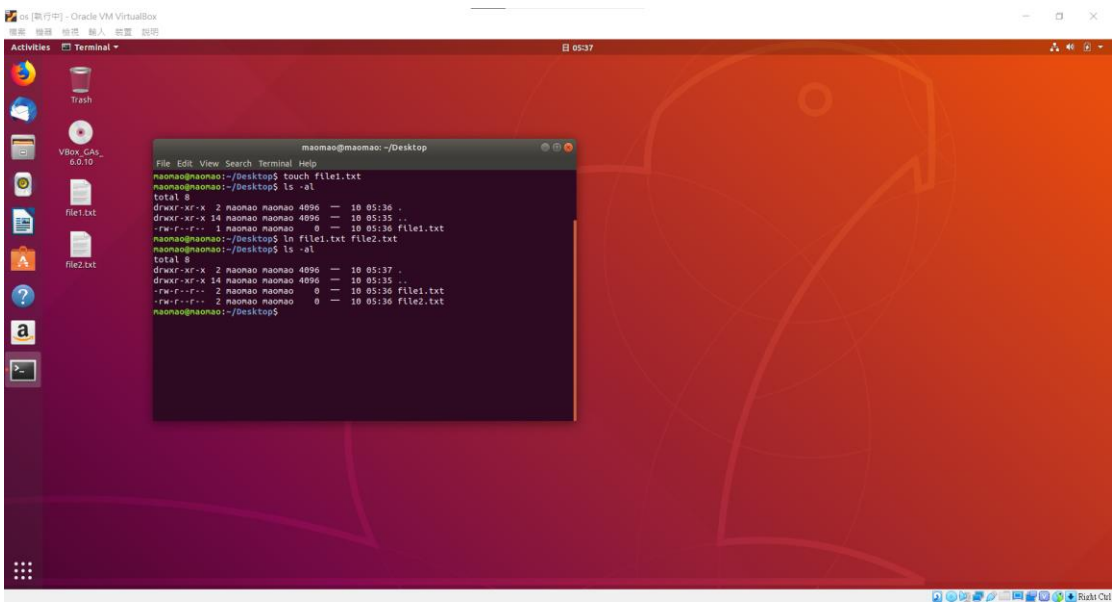
1. Create a text file as file1.txt

$ touch file1.txt



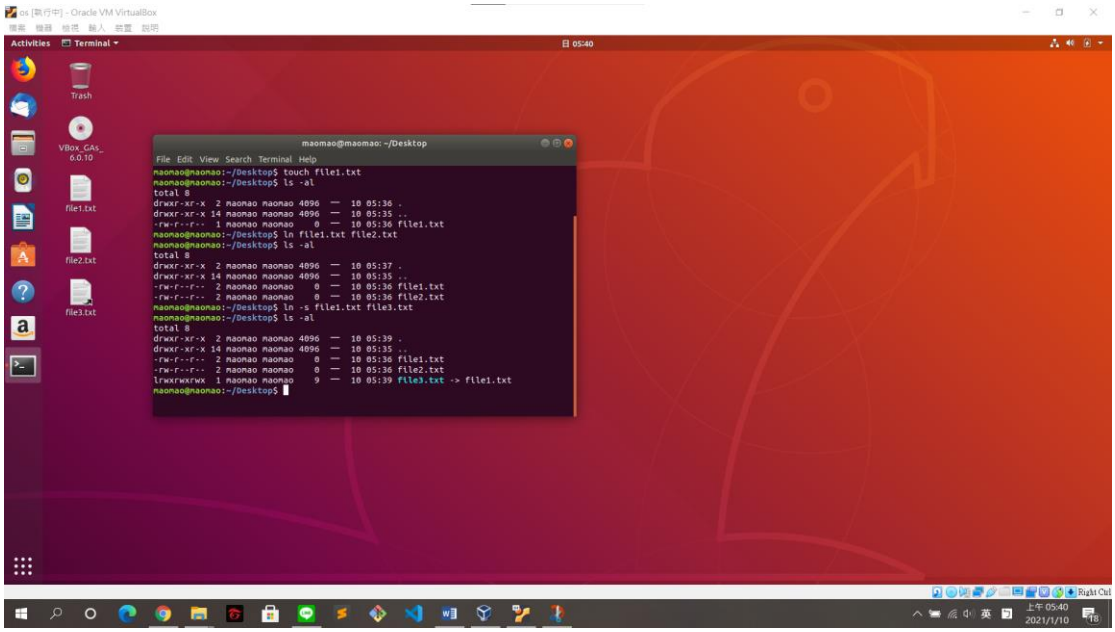
2. Create a hard link from file1.txt to file2.txt

$ ln file1.txt file2.txt



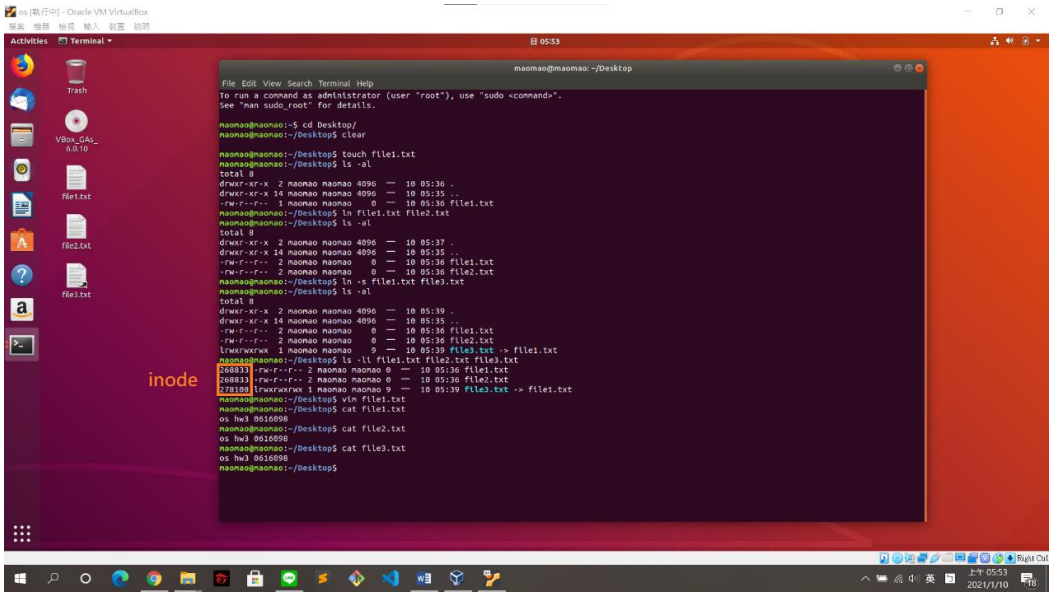
3. Create a soft link from file1.txt to file3.txt

$ ln -s file1.txt file3.txt



What are the inode values of each file?

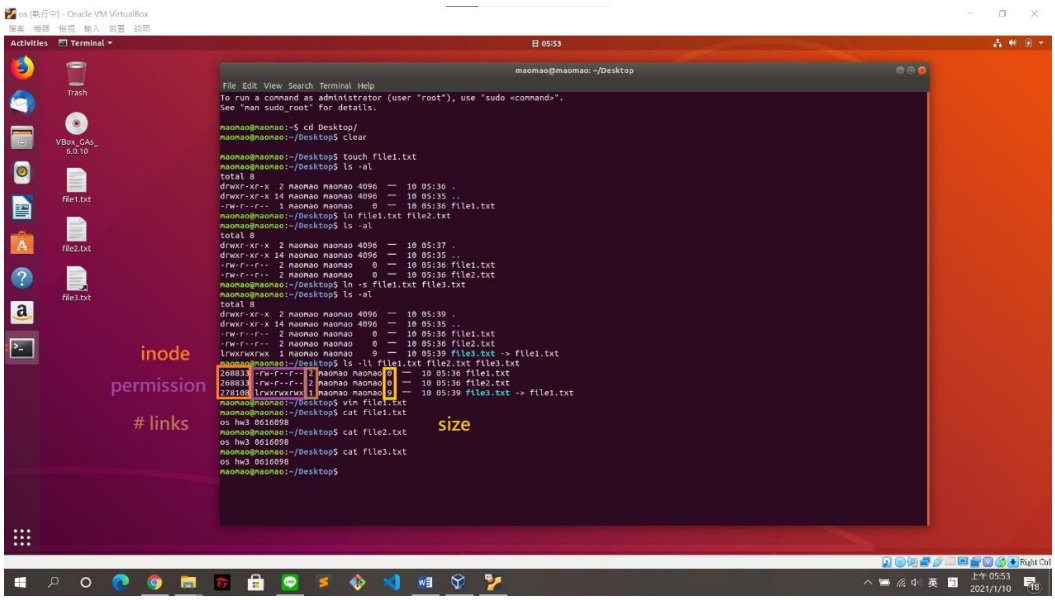
$ ls -li file1.txt file2.txt file3.txt



The inode of hard link (file2.txt) is 268833, which is same as the file(file1.txt).

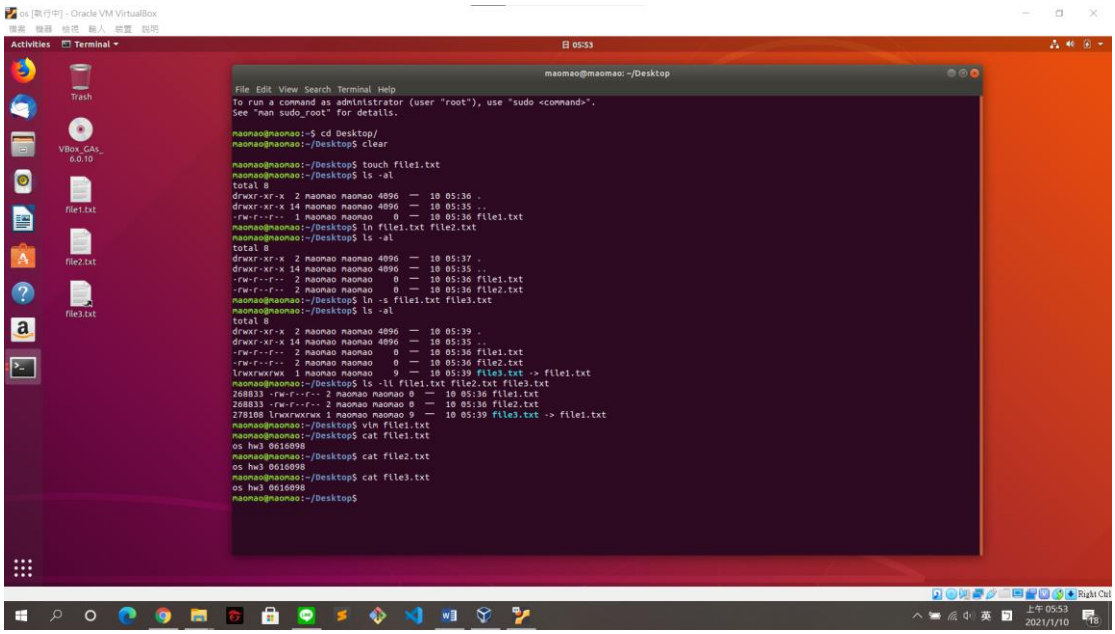
The inode of soft link (file3.txt) is 278108, which is different from the file(file1.txt).

Do they have the same content?



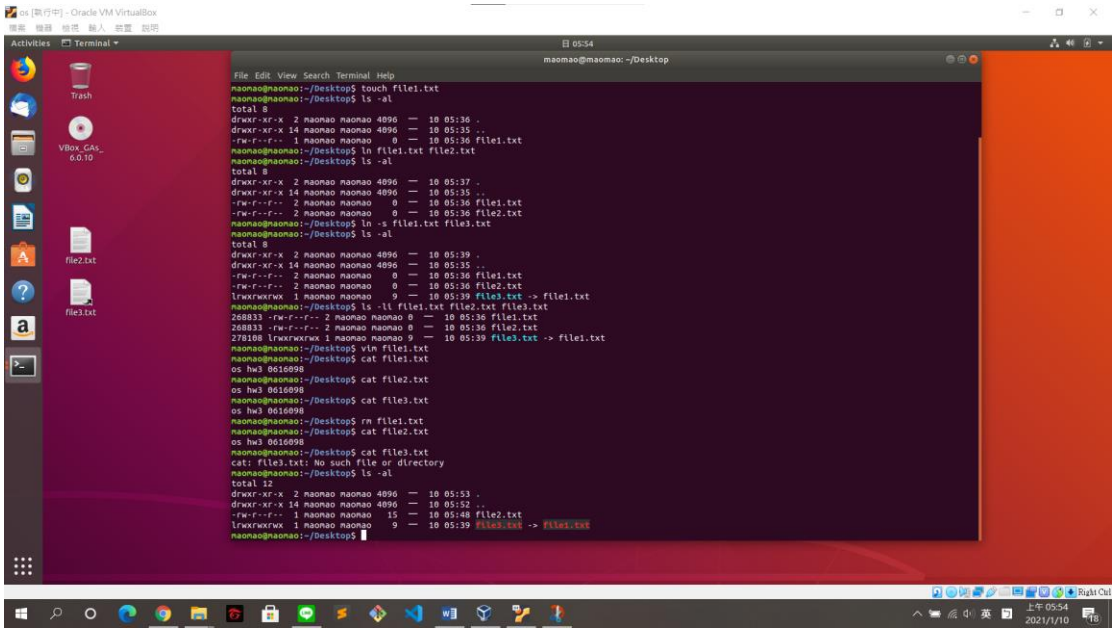
The attribute of hard link (file2.txt) is same as the file(file1.txt), e.g., Inode, permission, # links, size. And it is a file not a link.

The attribute of soft link (file3.txt) is different from the file(file1.txt), e.g., Inode, permission, # links, size. And it is a link not a file.



The content of hard link (file2.txt) is same as the file(file1.txt).

The content of soft link (file3.txt) is same as the file(file1.txt).



When removing the file (file1.txt):

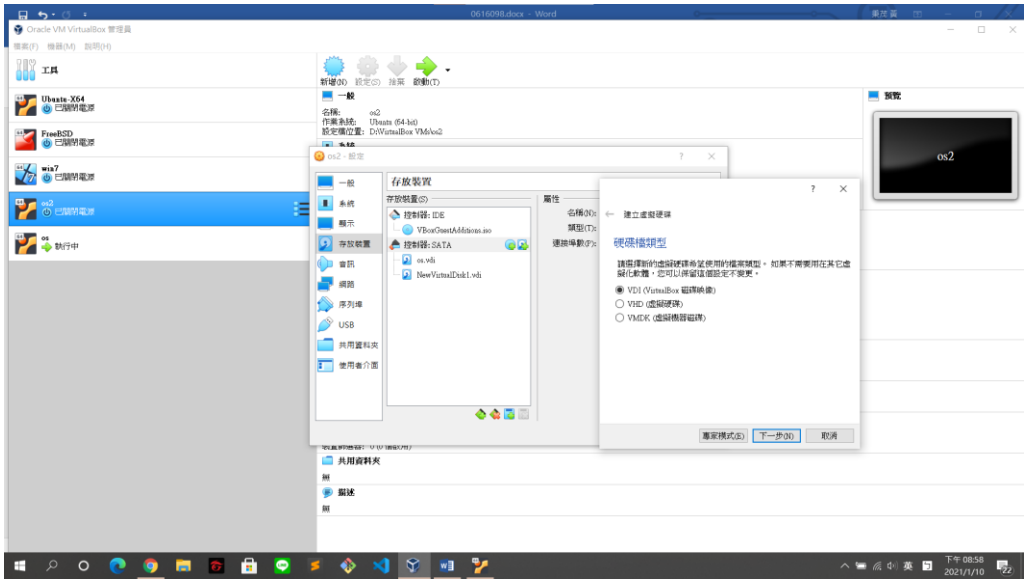
hard link (file2.txt) still exists, and the content doesn’ t changed.

soft link (file3.txt) is also removed.

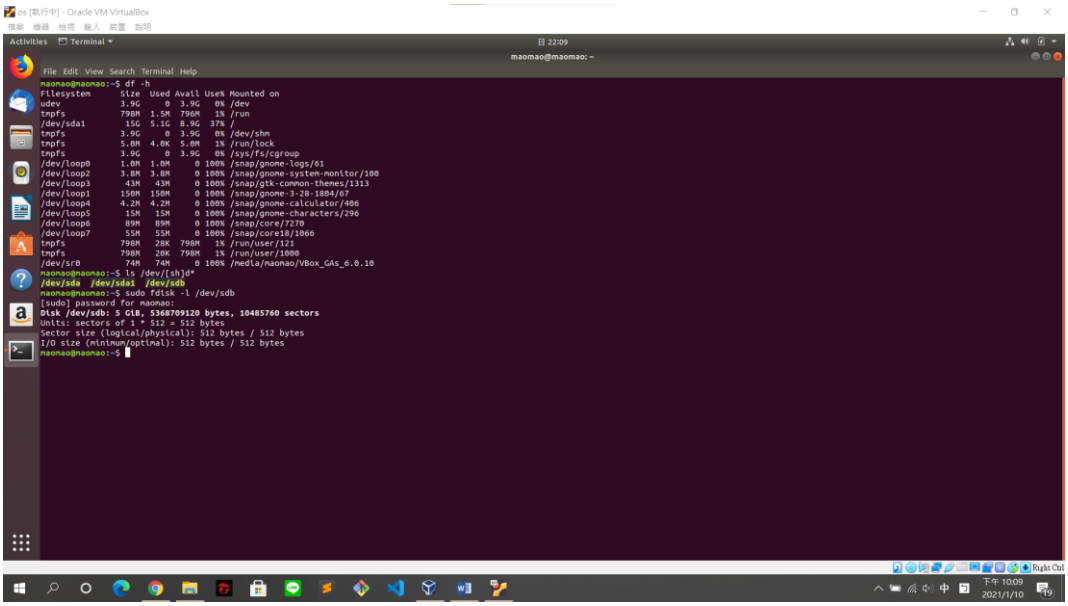
Task2 - Creating and mounting file system

1. Use the fdisk command to add a new 500MB logical partition to your hard drive.

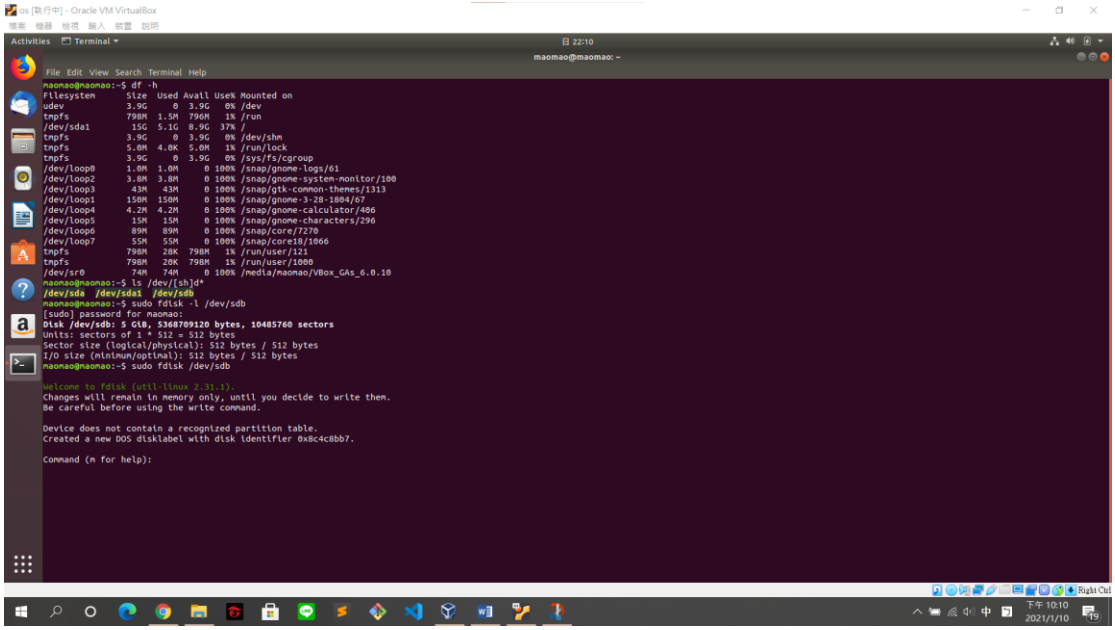
先新增虛擬硬碟以做實驗

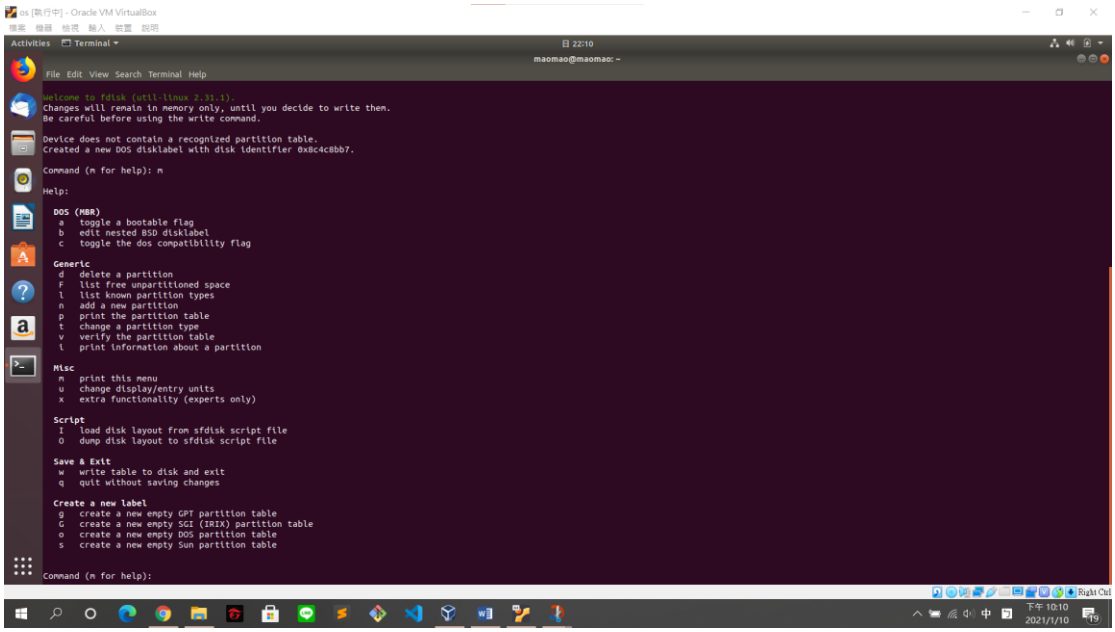


preceding operation



$ sudo fdisk /dev/sdb

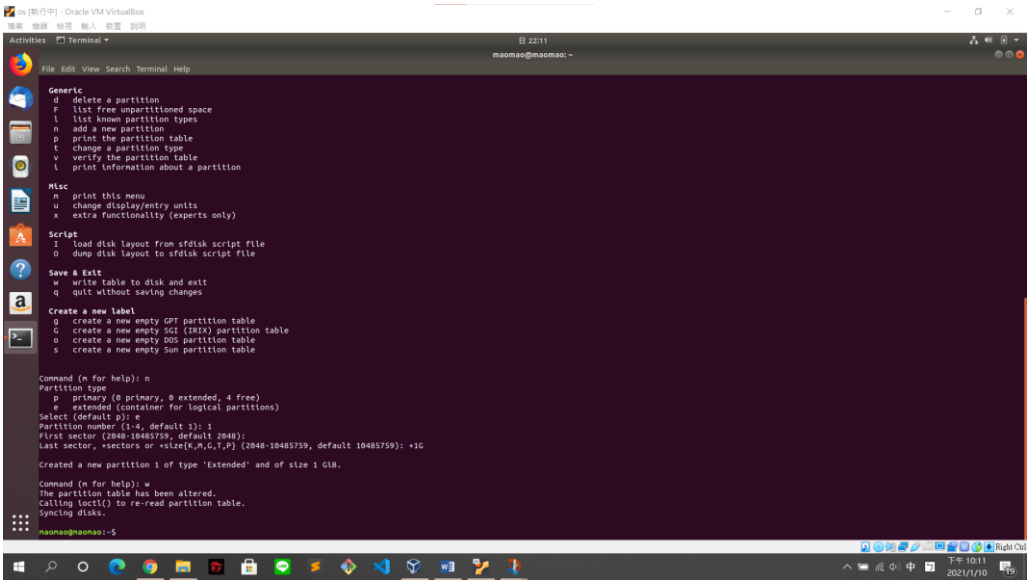




Create extended partition first, which is the

container for logical partitions

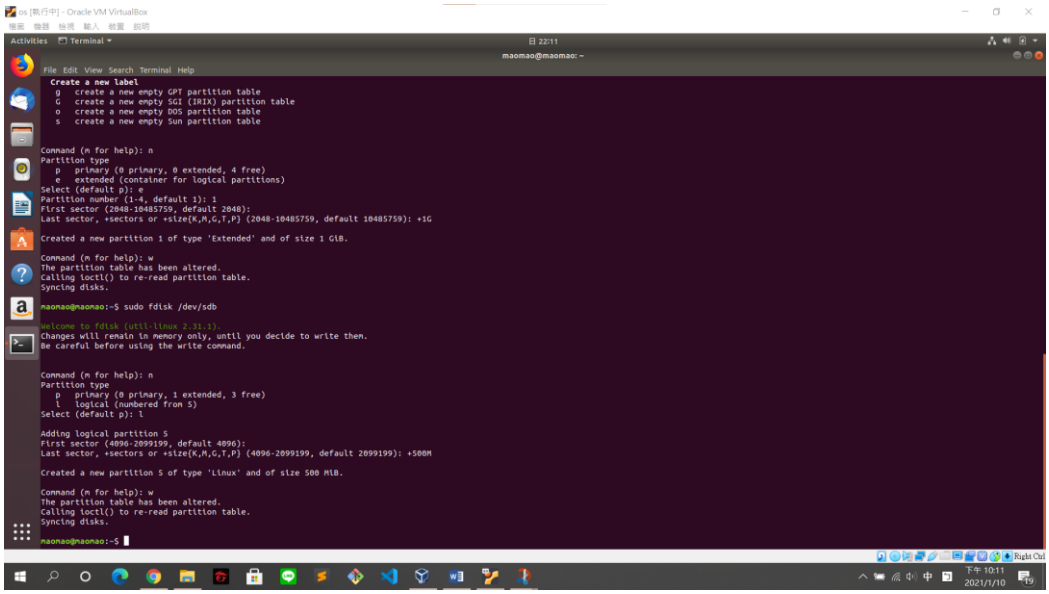
n -> e -> 1 -> “enter” -> +1G (must more than 500MB) -> w



$ sudo fdisk /dev/sdb

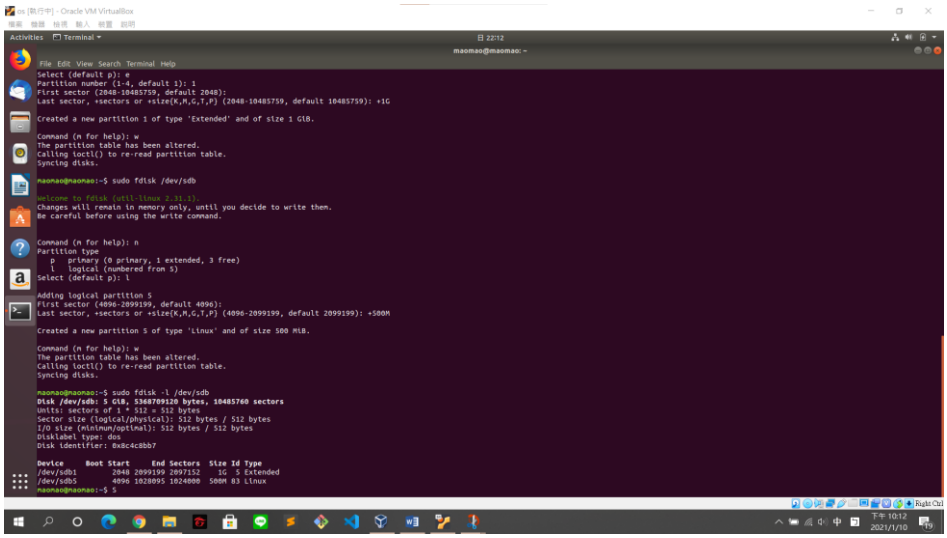
Create logical partition

n -> l -> “enter” -> +500M -> w



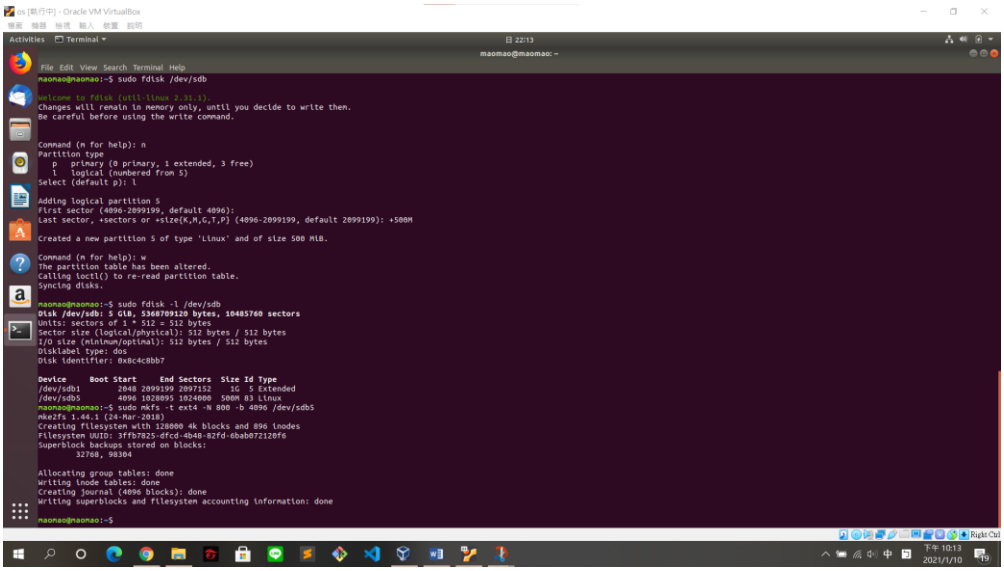
2. Use the fdisk -l command to verify that the new partition has been created.

$ sudo fdisk -l /dev/sdb



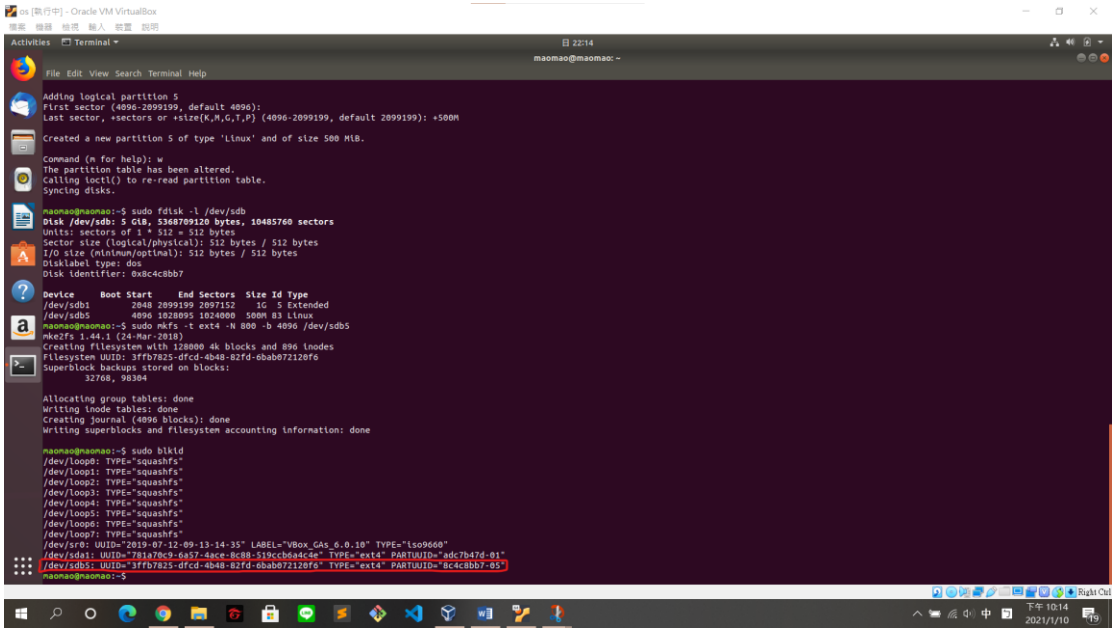
3. Format this partition with an ext4 file system that contains 800 inodes and block size is 4096 bytes.

$ sudo mkfs -t ext4 -N 800 -b 4096 /dev/sdb5

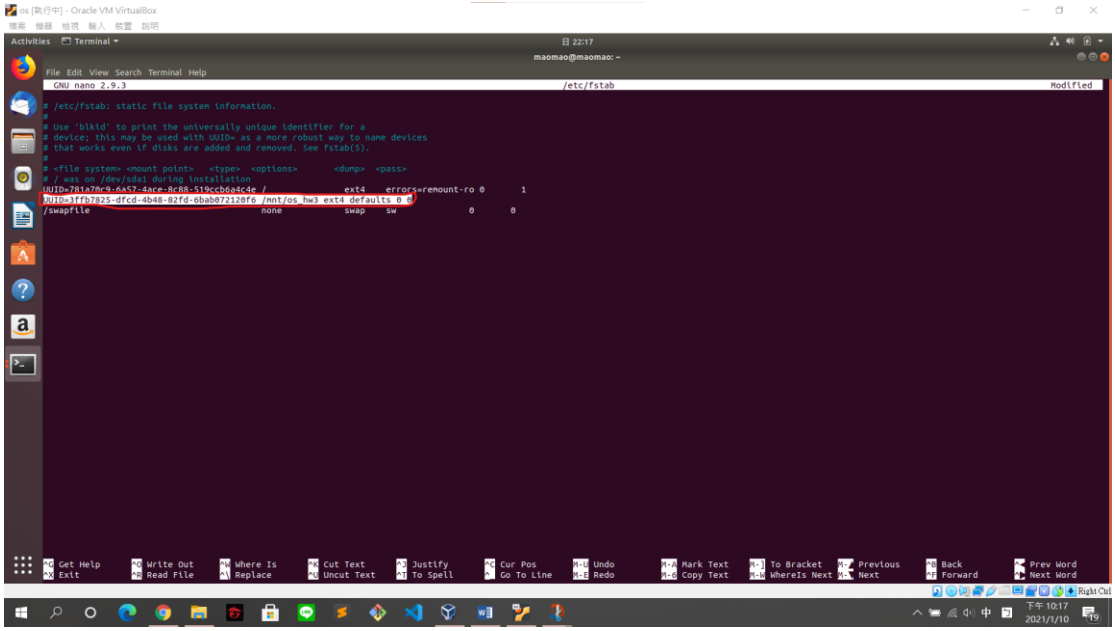


4. Edit /etc/fstab and reboot to mount file system.

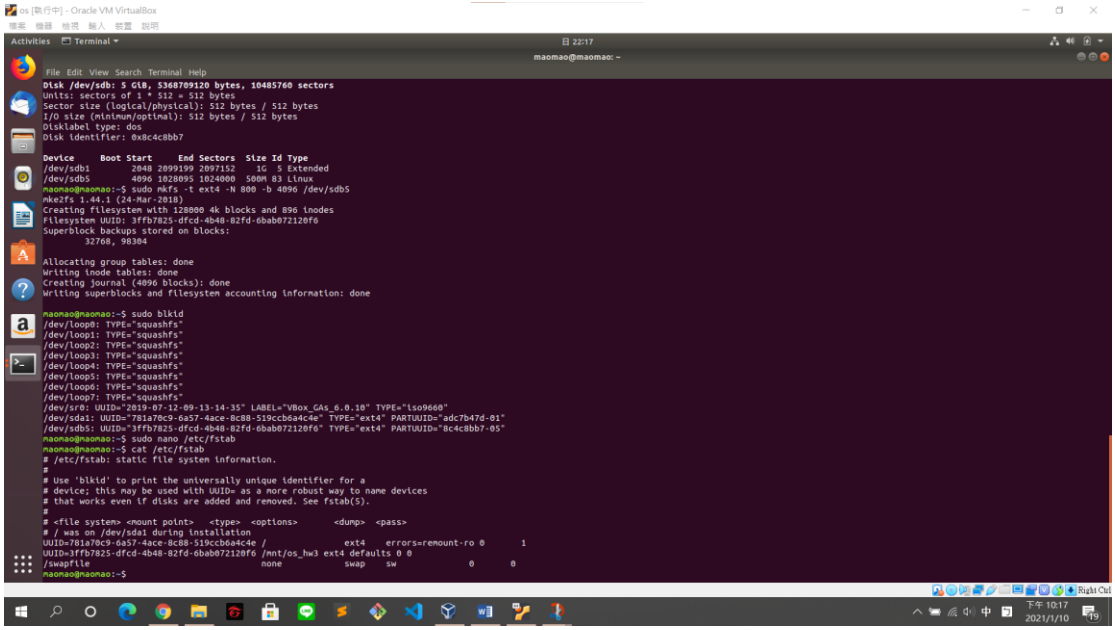
$ sudo blkid (check UUID)



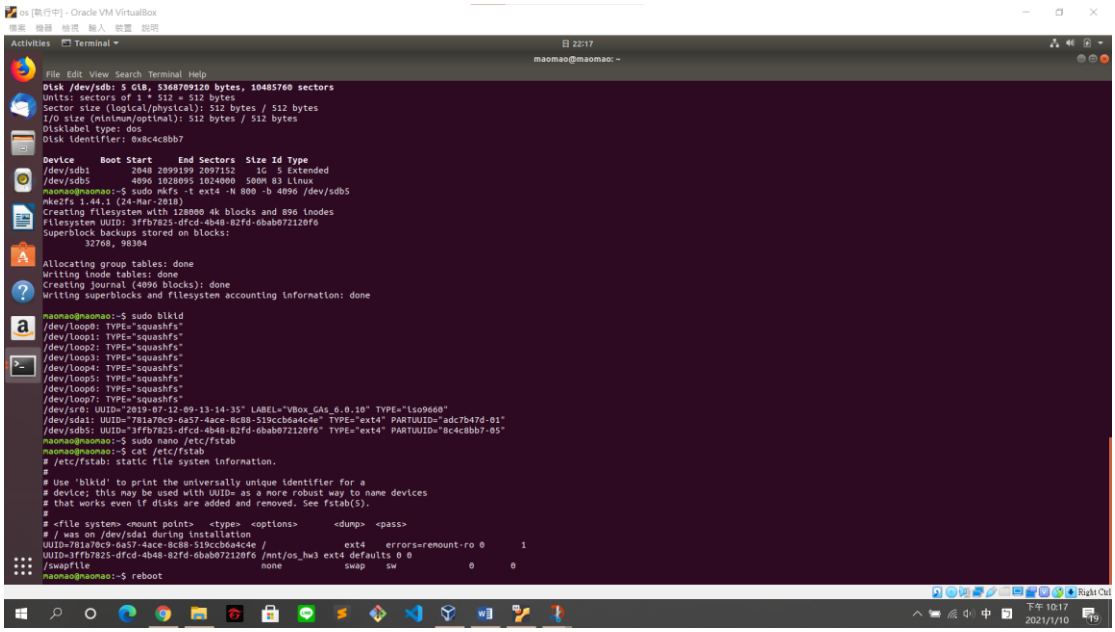
$ sudo nano /etc/fstab



check

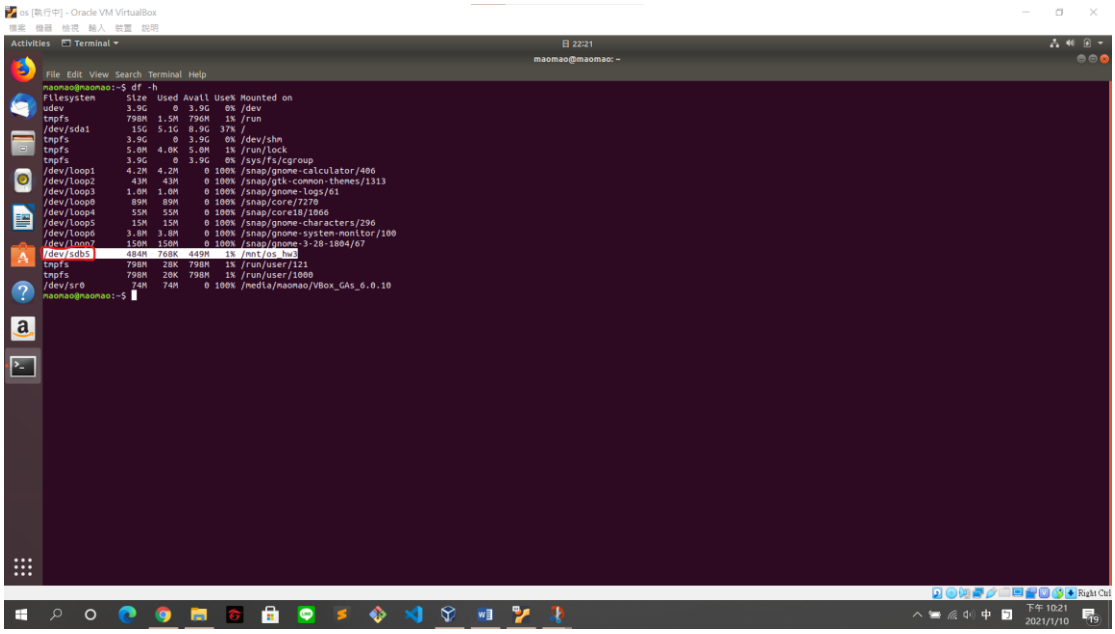


And then, reboot



5. Use the df command to confirm whether the mount is success

$ df -h

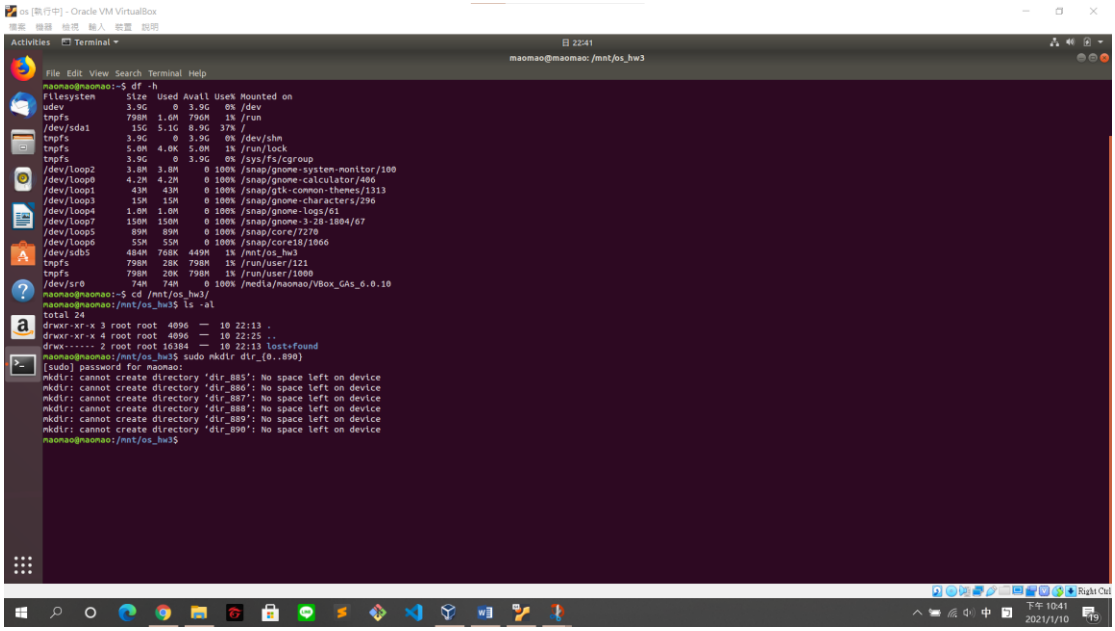


Task3 – Inode and block

1. Try to create directories in this file system as many as you can. How many directories can be created in this file system? Why? (Hint: inode)

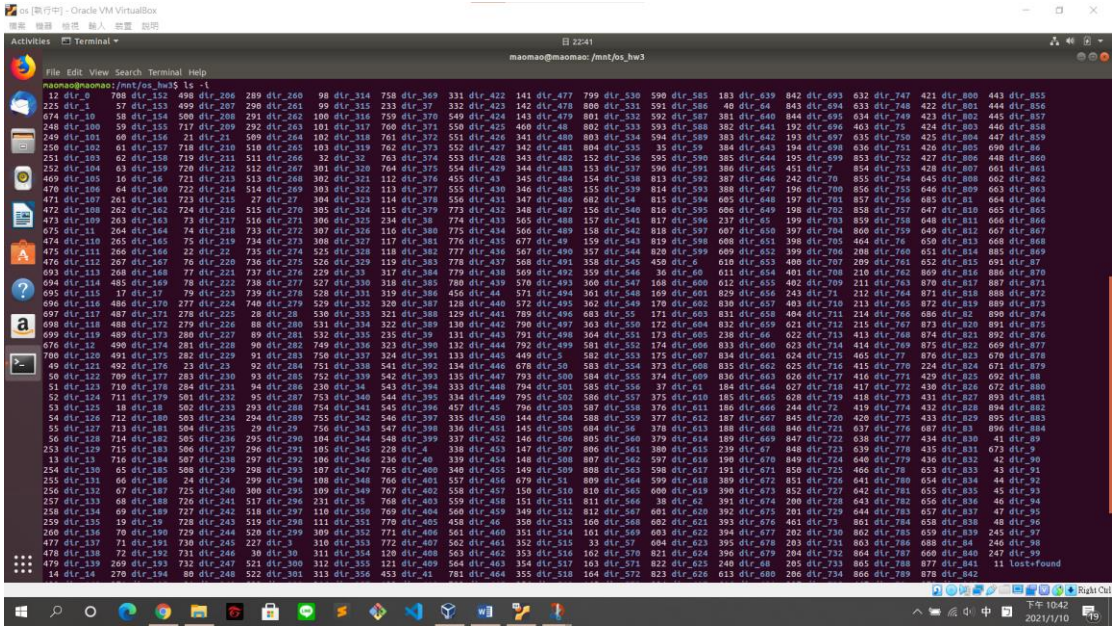
   $ sudo mkdir dir_{0..890}

   (create 891 directories)

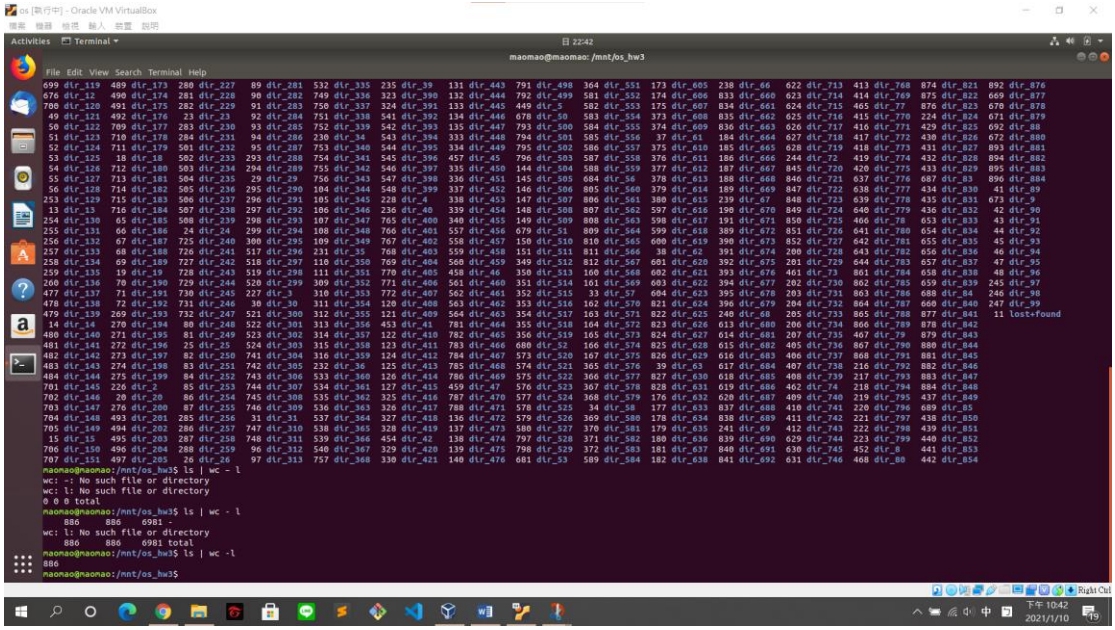


   (6 directories cannot be created)

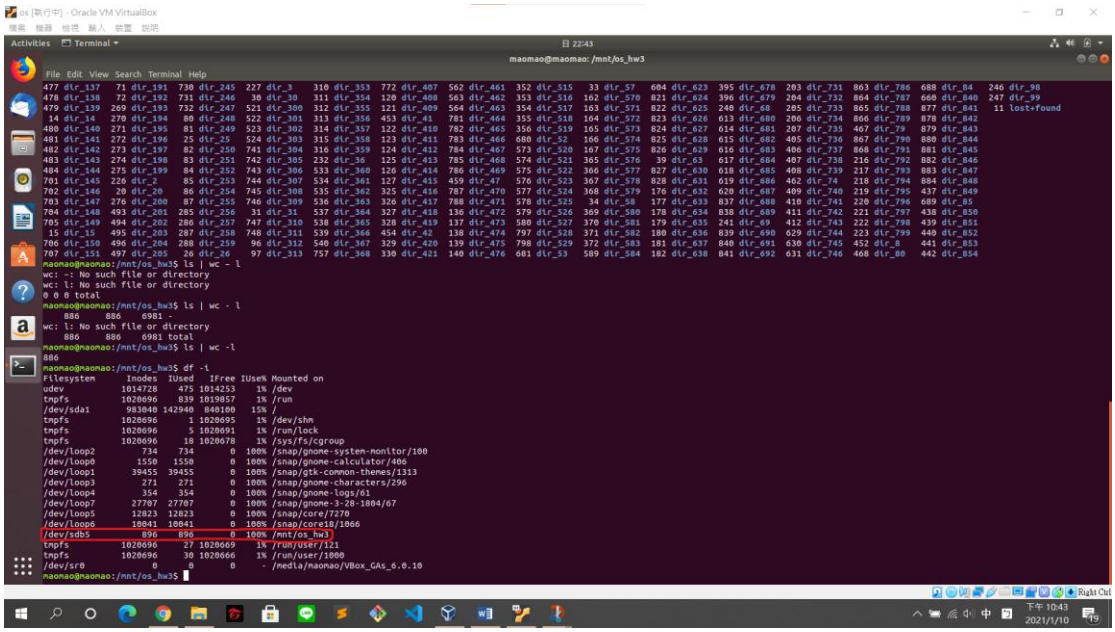
$ ls -i (show inodes)



$ ls | wc -l (count the numbers of directories)



885 directories can be created in this file system.



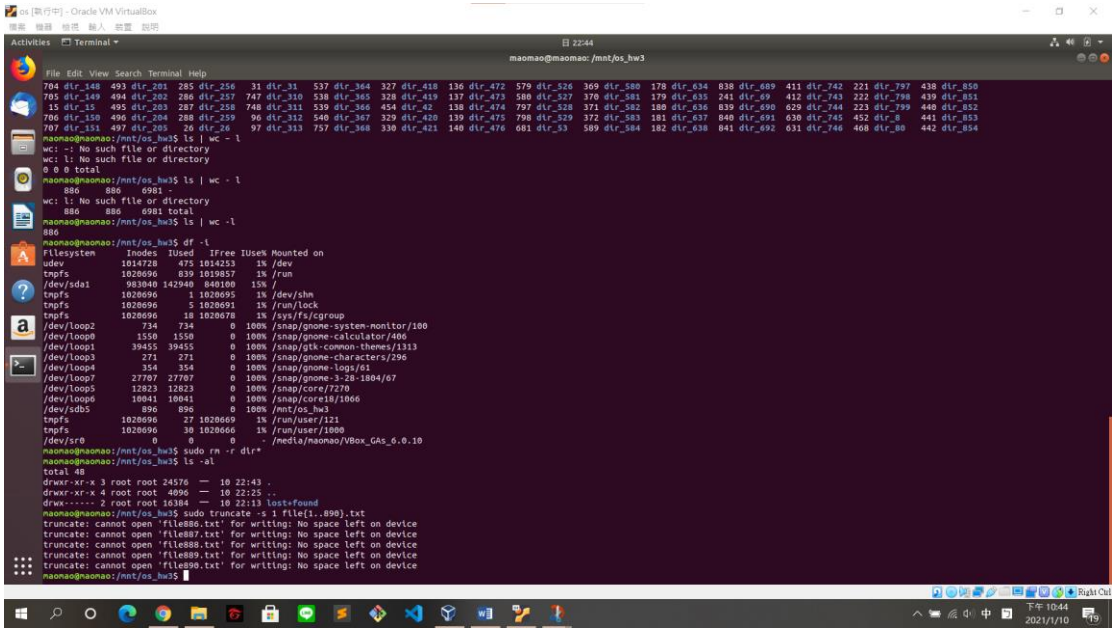
Only 885 directories can be created in this file system because inodes run out.

雖然 mkfs 是指定 800 inodes，但 inodes 的數量有一定的規則，所以會自動增加成 896 inodes，因為是離散的所以無法真的是 800 inodes。因為結構跟設計的關係所以只能用到快接近滿而非用掉 896

2. Try to create 1-byte files in this file system as many as you can. How many 1-byte files can be created in this file system? Can it completely use all space in this file system? (Hint: block size is 4096 bytes)

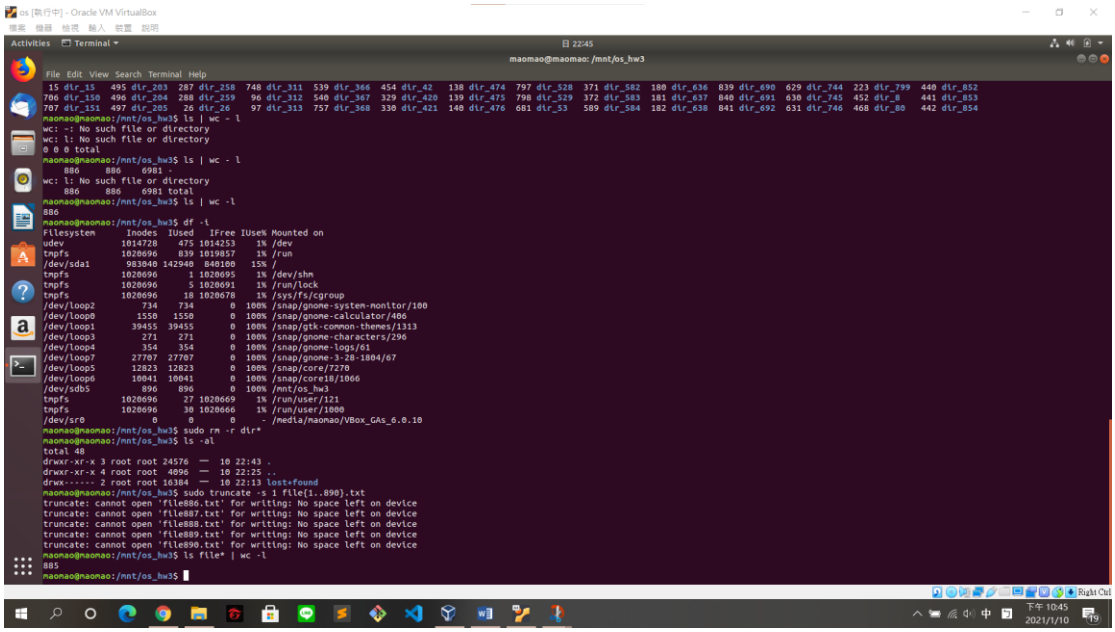
$ sudo truncate -s 1 file_{1..890}

(create 890 files)

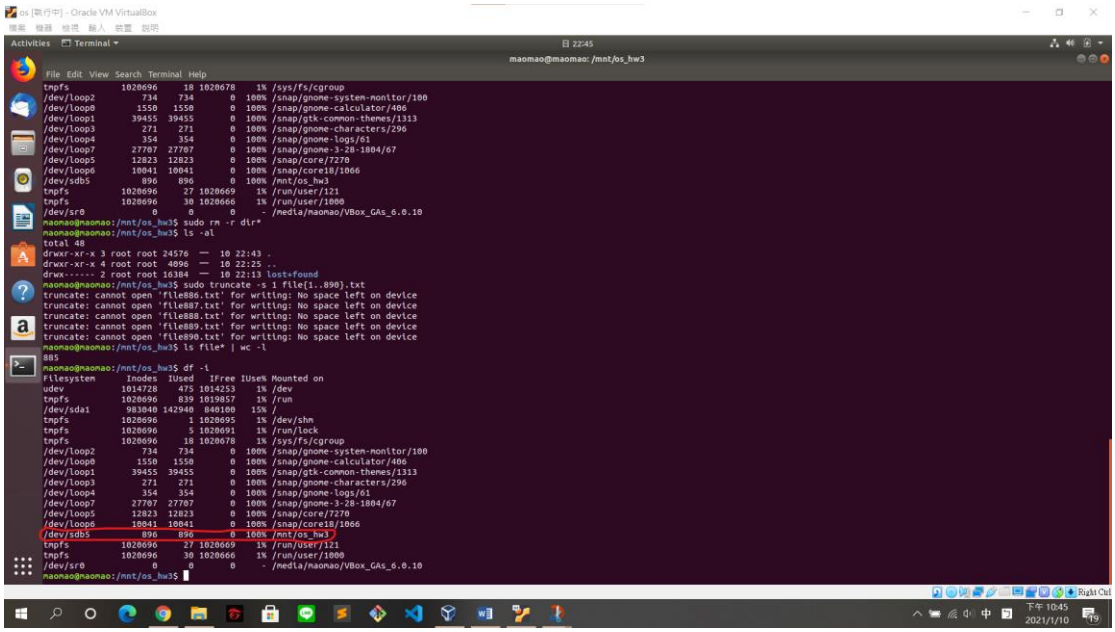


(5 files cannot be created)

$ ls file* | wc -l (count the numbers of files)



885 files can be created in this file system.

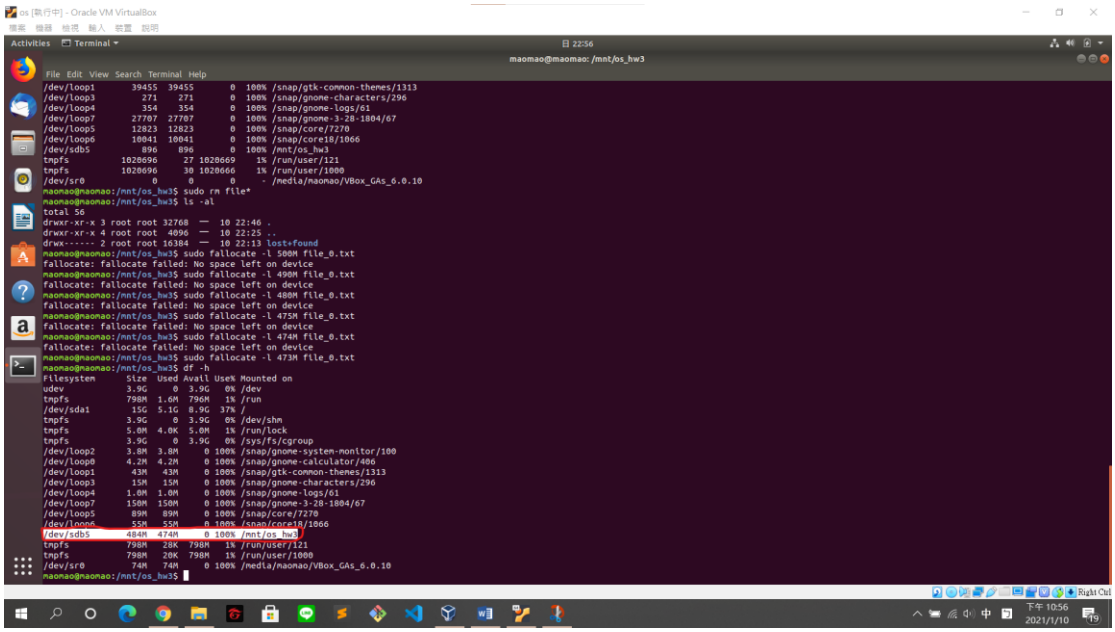


It cannot completely use all space in this file system because inodes run out.

雖然 mkfs 是指定 800 inodes，但 inodes 的數量有一定的規則，所以會自動增加成 896 inodes，因為是離散的所以無法真的是 800 inodes。因為結構跟設計的關係所以只能用到快接近滿而非用掉 896

3. Try to create a file which size as large as you can. What is the maximum file size? Can it completely use all space in this file system?

$ sudo fallocate -l 473M file_0.txt



The maximum file size is 473 MB.

It cannot completely use all space in this file system.

Some space for system not for data and the usage runs out.

By default, 5% of the filesystem blocks will be reserved for the super-user, to avoid fragmentation and "allow root-owned daemons to continue to function correctly after non-privileged processes are prevented from writing to the filesystem.