

# HW3

學號:0716049

姓名:詹凱傑

## Task1- Soft Link and Hard Link

### Hard Link:

```
user@user-VirtualBox:~/Documents/OS_hw3$ touch file1.txt
user@user-VirtualBox:~/Documents/OS_hw3$ ls -li file1.txt
3014687 file1.txt
user@user-VirtualBox:~/Documents/OS_hw3$ ln file1.txt file2.txt
user@user-VirtualBox:~/Documents/OS_hw3$ ls -li
3014687 file1.txt 3014687 file2.txt      2 file_system
user@user-VirtualBox:~/Documents/OS_hw3$ echo "Hello World" >> file1.txt
user@user-VirtualBox:~/Documents/OS_hw3$ cat file1.txt
Hello World
user@user-VirtualBox:~/Documents/OS_hw3$ cat file2.txt
Hello World
user@user-VirtualBox:~/Documents/OS_hw3$ rm file1.txt
user@user-VirtualBox:~/Documents/OS_hw3$ cat file2.txt
Hello World
user@user-VirtualBox:~/Documents/OS_hw3$
```

新增一個 file1.txt

```
$ touch file1.txt
```

查看 file1.txt 的 inode number

```
$ ls -li file1.txt
```

建立 hard link

```
$ ln file1.txt file2.txt
```

```
$ ls -li ==> file1.txt 和 file2.txt 的 inode number 相同
```

編輯 file1.txt

```
$ echo "Hello World" >> file1.txt
```

```
$ cat file1.txt
```

```
$ cat file2.txt
```

將 file1.txt 刪除、並查看 file2.txt

```
$ rm file1.txt
```

```
$ cat file2.txt
```

### Sort Link

```
user@user-VirtualBox:~/Documents/OS_hw3$ touch file1.txt
user@user-VirtualBox:~/Documents/OS_hw3$ ls -li file1.txt
3014687 file1.txt
user@user-VirtualBox:~/Documents/OS_hw3$ ln -s file1.txt file3.txt
user@user-VirtualBox:~/Documents/OS_hw3$ ls -li
3014687 file1.txt 3014801 file3.txt      2 file_system
user@user-VirtualBox:~/Documents/OS_hw3$ echo "Hello World" >> file1.txt
user@user-VirtualBox:~/Documents/OS_hw3$ cat file1.txt
Hello World
user@user-VirtualBox:~/Documents/OS_hw3$ cat file3.txt
Hello World
user@user-VirtualBox:~/Documents/OS_hw3$ rm file1.txt
user@user-VirtualBox:~/Documents/OS_hw3$ cat file3.txt
cat: file3.txt: No such file or directory
user@user-VirtualBox:~/Documents/OS_hw3$ ls
file3.txt file_system
```

新增一個 **file1.txt**

```
$ touch file1.txt
```

查看 **file1.txt** 的 **inode number**

```
$ ls -l file1.txt
```

建立 **soft link**

```
$ ln -s file1.txt file3.txt
```

```
$ ls -l      ==>  file1.txt 和 file3.txt 的 inode number 不同相同
```

編輯 **file1.txt**

```
$ echo "Hello World" >> file1.txt
```

```
$ cat file1.txt
```

```
$ cat file3.txt
```

將 **file1.txt** 刪除、並查看 **file3.txt**

```
$ rm file1.txt
```

```
$ cat file3.txt
```

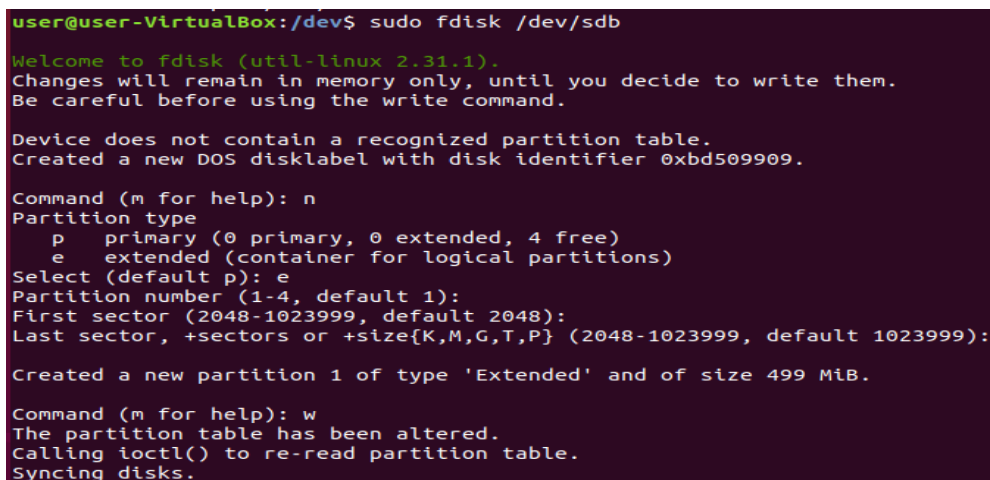
說明:

Inode 是用來儲存檔案的資訊，inode number 用來識別描述這個檔案的 inode 是哪一個。Hard Link 建立的檔案會對應到同一個 inode number，所以修改其中一個檔案時，所有連結到相同 inode number 的檔案都會被改，因為本質上是同一個檔案，若將原本的檔案刪除後，由於還有其他檔案指到這個 inode number，所以這個檔案還是存在。Soft Link 的部分，類似建立一個捷徑，兩個檔案 inode number 不同，是建立一個連結。所以當把被連結的檔案刪除後，這個檔案就不見了，連結只會指到一個不存在的檔案。

## Task2- Creating and Mounting File System

### 1. 新增一個 500MB 的邏輯分割區: (新增的磁碟是/dev/sdb)

```
$ sudo fdisk /dev/sdb
```



```
user@user-VirtualBox:/dev$ sudo fdisk /dev/sdb

Welcome to fdisk (util-linux 2.31.1).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Device does not contain a recognized partition table.
Created a new DOS disklabel with disk identifier 0xbd509909.

Command (m for help): n
Partition type
   p   primary (0 primary, 0 extended, 4 free)
   e   extended (container for logical partitions)
Select (default p): e
Partition number (1-4, default 1):
First sector (2048-1023999, default 2048):
Last sector, +sectors or +size{K,M,G,T,P} (2048-1023999, default 1023999):

Created a new partition 1 of type 'Extended' and of size 499 MiB.

Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.
```

## 2. 查看新的分割去有沒有被新增

\$ sudo fdisk -l

```
Disk /dev/sdb: 500 MiB, 524288000 bytes, 1024000 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0xbd509909
```

## 3. 格式化，ext4 file system、800 inodes、block size 4096 B

\$ sudo mkfs.ext4 -N 800 -b 4096 /dev/sdb

```
user@user-VirtualBox:/dev$ sudo mkfs.ext4 -N 800 -b 4096 /dev/sdb
mke2fs 1.44.1 (24-Mar-2018)
Found a dos partition table in /dev/sdb
Proceed anyway? (y,N) y
Creating filesystem with 128000 4k blocks and 896 inodes
Filesystem UUID: 6cf1b512-f15f-4411-9ae6-67274c4fb6ea
Superblock backups stored on blocks:
    32768, 98304

Allocating group tables: done
Writing inode tables: done
Creating journal (4096 blocks): done
Writing superblocks and filesystem accounting information: done
```

## 4. 編輯/etc/fstab，讓系統一開機就自動 Mount

先查看/dev/sdb 的 UUID:

\$ sudo blkid | grep /dev/sdb

```
user@user-VirtualBox:/dev$ sudo blkid | grep /dev/sdb
/dev/sdb: UUID="6cf1b512-f15f-4411-9ae6-67274c4fb6ea" TYPE="ext4"
user@user-VirtualBox:/dev$
```

再去修改/etc/fstab

```
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point> <type> <options> <dump> <pass>
# / was on /dev/sda1 during installation
UUID=39fe5aa0-ccb2-443c-ad87-920239ddd38d / ext4 errors=remount-ro 0
1
/swapfile none swap sw 0
0
UUID=6cf1b512-f15f-4411-9ae6-67274c4fb6ea /home/user/Documents/OS_hw3/file_system ext4
defaults 0
```

## 5. 重開機，確認 Mound 有成功

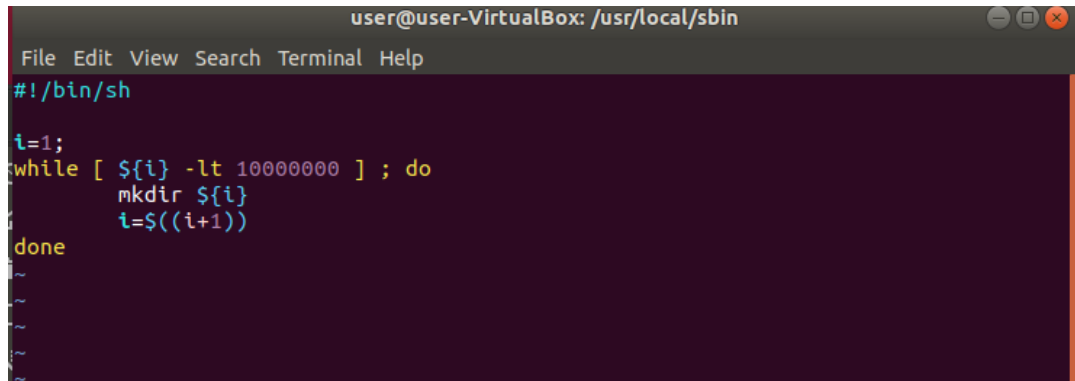
\$ df -h | grep /dev/sdb

```
ser@user-VirtualBox:~$ df -h | grep /dev/sdb
dev/sdb      484M 768K 449M  1% /home/user/Documents/OS_hw3/file_system
ser@user-VirtualBox:~$
```

## Task3- Inode and Block

### 1. 可以新增 **directories** 的數量

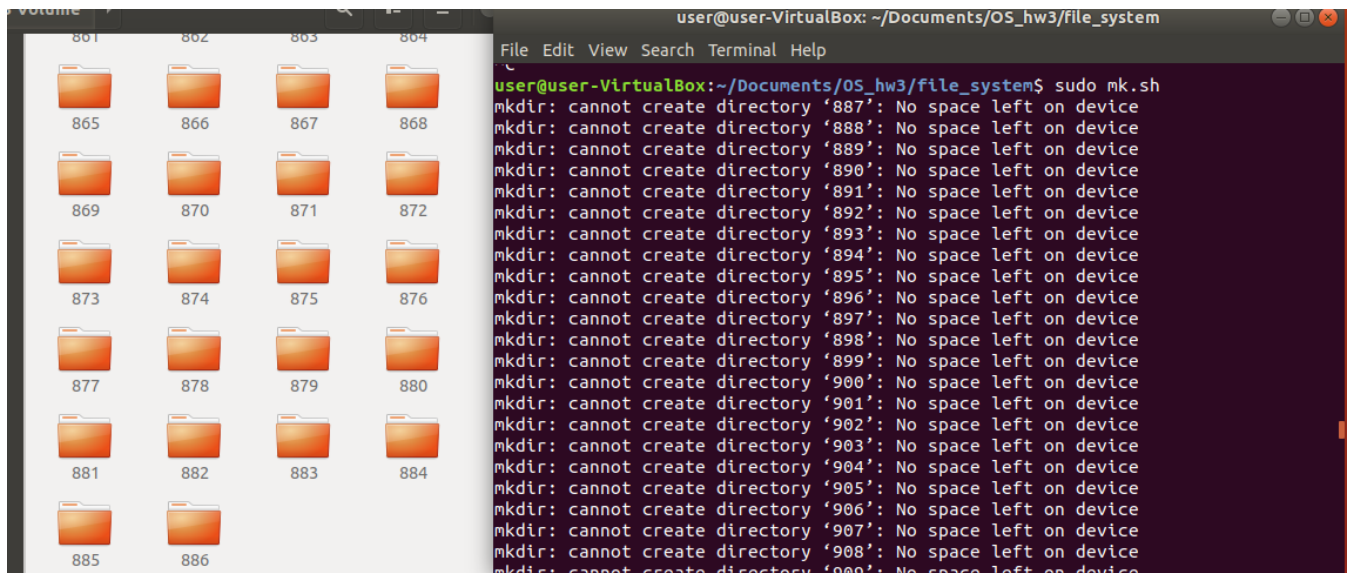
- a. 寫一個 script，自動生成資料夾



```
user@user-VirtualBox: /usr/local/sbin
File Edit View Search Terminal Help
#!/bin/sh

i=1;
while [ ${i} -lt 100000000 ] ; do
    mkdir ${i}
    i=$((i+1))
done
```

- b. 在 Task 2 的檔案系統中執行這個 script  
結果:



新增到 886 個資料夾後，就不能在新增了。

#### 說明:

我是寫一個 script 來自動產生資料夾並計數的，在生成 886 個目錄後就不能在新增了，由於每個目錄都要有一個 **inode** 紀錄資訊，所以可以新增目錄的數量就是 **inode** 的數量，雖然我在初始化的時候，設定的 **inode** 總數是 800，但初始完後看 **inode** 是總數有 800 多個，我覺得是跟檔案系統有關，所以才會比我設定的數量還多。

## 2. 可新增 1-byte files 的數量

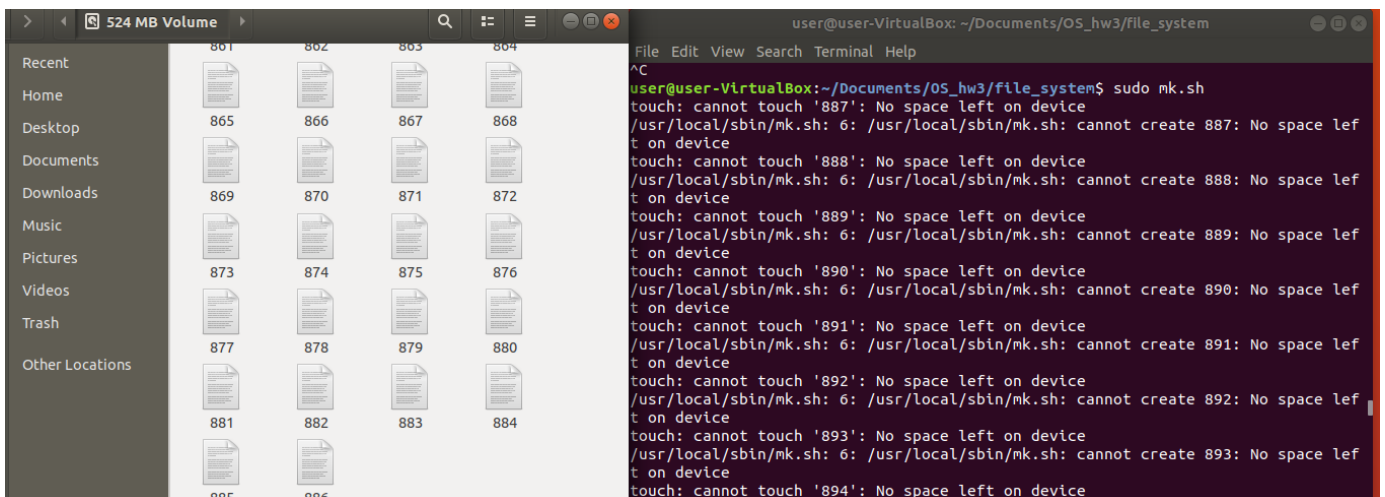
- a. 修改上一題的 script，改成生成很多 1- byte 的檔案

```
user@user-VirtualBox: ~/Documents/OS_hw3/file_system
File Edit View Search Terminal Help
#!/bin/sh

i=1;
while [ ${i} -lt 100000000 ] ; do
    touch ${i}
    echo "1" >> ${i}
    i=$((i+1))
done
```

- b. 在 Task 2 的檔案系統中執行這個 script

結果:



空間的使用量:

```
user@user-VirtualBox:~/Documents/OS_hw3/file_system$ df -h | grep /dev/sdb
/dev/sdb      484M  4.3M  445M   1% /home/user/Documents/OS_hw3/file_system
user@user-VirtualBox:~/Documents/OS_hw3/file_system$
```

說明:

跟上一題一樣，我用 script 自動生成檔案，並寫入一個字元，然後看可以生成多少檔案，而可以新增的檔案個數也跟上一題一樣，因為每個檔案也會用到 inode 來記錄資訊，所以我只能新增 886 個 1-byte 的檔案。不過雖然每個檔案我只有寫入 1byte 的資料，但因為 block size 是 4096 byte，所以每個檔案也用到 4096 個 byte。從硬碟的總使用量來看，可以看出使用量只有 1%左右，還有很多空間沒用到。

### 3. 可新增最大檔案的大小

- a. 修改上一題的 script，改成寫入資料到 test 中

```
#!/bin/sh

i=1;
while [ ${i} -lt 100000000 ] ; do
    echo "hello world" >> test
    i=$((i+1))
done
```

- b. 結果:

在執行幾次上述的 script 後，空間就滿了。

```
user@user-VirtualBox:~/Documents/OS_hw3/file_system$ sudo mk.sh
sh: echo: I/O error
sh: echo: I/O error
sh: echo: I/O error
sh: echo: I/O error
sh: echo: I/O error
sh: echo: I/O error
sh: echo: I/O error
sh: echo: I/O error

user@user-VirtualBox:~/Documents/OS_hw3/file_system$ ls -al
total 484464
drwxr-xr-x 2 root root    20480  7 15:29 .
drwxr-xr-x 3 user user     4096  6 23:56 ..
-rw-r--r-- 1 root root 496062464  7 15:56 test
user@user-VirtualBox:~/Documents/OS_hw3/file_system$
```

硬碟的使用量也到了 100%

```
user@user-VirtualBox:~/Documents/OS_hw3/file_system$ df -h | grep /dev/sdb
/dev/sdb      484M  474M    0 100% /home/user/Documents/OS_hw3/file_system
user@user-VirtualBox:~/Documents/OS_hw3/file_system$
```

#### 說明:

這個部分，我是用 script 自動寫入資料到 test 這個檔案中，看可以寫入多少，大概在我執行四次時，可能是因為硬碟快滿了，虛擬機突然變得很慢，然後就當機了，後來我從重開後，看硬碟的使用量已經達到 100%，且我又嘗試寫入後會有 error。在 ext4 的檔案系統下，單一檔案的大小可以超過 500MB 很多，所以我試起來用一個很大的檔案是可以用到快要全部的硬碟。