

作業系統概論 HW3 Report

0716032 林佑鑫

Task1 – Soft Link and Hard Link

```
linyouxin@ubuntu:~/Desktop/hw3$ sudo vim file1.txt
linyouxin@ubuntu:~/Desktop/hw3$ ln file1.txt file2.txt
ln: failed to create hard link 'file2.txt' => 'file1.txt': Operation not permitted
linyouxin@ubuntu:~/Desktop/hw3$ sudo !!
sudo ln file1.txt file2.txt
linyouxin@ubuntu:~/Desktop/hw3$ sudo ln -s file1.txt file3.txt
```

1. 一開始 link 完後 edit/delete 前 content 都相同。

```
linyouxin@ubuntu:~/Desktop/hw3$ cat file1.txt
This is content of file1.
linyouxin@ubuntu:~/Desktop/hw3$ cat file2.txt
This is content of file1.
linyouxin@ubuntu:~/Desktop/hw3$ cat file3.txt
This is content of file1.
linyouxin@ubuntu:~/Desktop/hw3$ ls -li
total 8
411946 -rw-r--r-- 2 root root 26 Jan  9 07:08 file1.txt
411946 -rw-r--r-- 2 root root 26 Jan  9 07:08 file2.txt
400819 lrwxrwxrwx 1 root root  9 Jan  9 07:09 file3.txt -> file1.txt
```

inode 的值：用 hard link 時 file2.txt 的 inode 與原檔案 file1.txt 的 inode 是用同一個值 411946，然後 inode 會指到一樣的 data blocks，所以 content 一樣，symbolic link 會用另一個 inode 400819 到它的 data blocks，然後這個 data blocks 裡面會指到要 link 的檔案，所以 content 也一樣。

2. edit 完後，三個 content 還是一樣，inode 值也沒變。

```
linyouxin@ubuntu:~/Desktop/hw3$ cat file1.txt
This is new content of file1!
linyouxin@ubuntu:~/Desktop/hw3$ cat file2.txt
This is new content of file1!
linyouxin@ubuntu:~/Desktop/hw3$ cat file3.txt
This is new content of file1!
linyouxin@ubuntu:~/Desktop/hw3$ ls -li
total 8
411946 -rw-r--r-- 2 root root 30 Jan  9 07:22 file1.txt
411946 -rw-r--r-- 2 root root 30 Jan  9 07:22 file2.txt
400819 lrwxrwxrwx 1 root root  9 Jan  9 07:09 file3.txt -> file1.txt
```

edit 改的是 data blocks 裡面的東西，inode 不會改變，所以三個檔案都會一起改。

3. delete 後因為 file2.txt 用 hard link 所以 inode 值還在，還能直接指到 data blocks 所以資料不會改變。而 file3.txt 用 soft link，它的 data blocks 要指到 file1.txt，但被刪掉了所以找不到，才會有 No such file or directory

```
linyouxin@ubuntu:~/Desktop/hw3$ rm file1.txt
rm: remove write-protected regular file 'file1.txt'? y
linyouxin@ubuntu:~/Desktop/hw3$ cat file2.txt
This is new content of file1!
linyouxin@ubuntu:~/Desktop/hw3$ cat file3.txt
cat: file3.txt: No such file or directory
linyouxin@ubuntu:~/Desktop/hw3$ ls -li
total 4
411946 -rw-r--r-- 1 root root 30 Jan  9 07:22 file2.txt
400819 lrwxrwxrwx 1 root root  9 Jan  9 07:09 file3.txt -> file1.txt
```

Task2 – Creating and mounting file system

```
sudo fdisk -l /dev/sdb
Disk /dev/sdb: 1 GiB, 1073741824 bytes, 2097152 sectors
Disk model: VMware Virtual S
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: 0xf811b6b7

Device      Boot Start      End Sectors  Size Id Type
/dev/sdb1           2048 1026047 1024000  500M 83 Linux
```

新建一個硬碟，然後將它分割出 500M。

```

mke2fs 1.45.5 (07-Jan-2020)
/dev/sdb contains a ext4 file system
    created on Sat Jan  9 08:02:54 2021
Proceed anyway? (y,N) y
Creating filesystem with 262144 4k blocks and 896 inodes
Filesystem UUID: 6dba06d0-a90b-40ed-a5d1-75e0bc03be9c
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376

Allocating group tables: done
Writing inode tables: done
Creating journal (8192 blocks): done
Writing superblocks and filesystem accounting information: done

```

格式化成 ext4 file system 每個 block size 4096bytes(4KB)，共有 896 個（800 補滿後） inodes。

```
/dev/sdb: UUID="6dba06d0-a90b-40ed-a5d1-75e0bc03be9c" TYPE="ext4"
```

先用 sudo blkid 查出剛剛新增硬碟的 UUID。

```
UUID="6dba06d0-a90b-40ed-a5d1-75e0bc03be9c" /newmount ext4 defaults 0 0
```

用查到的 UUID 在/etc/fstab 加上這行，讓系統開機時會自動掛載剛剛建立的硬碟/dev/sdb 在 /newmount。

```

llyouxin@ubuntu:~$ df
Filesystem      1K-blocks    Used Available Use% Mounted on
udev            6092008         0   6092008   0% /dev
tmpfs           1224084      1920   1222164   1% /run
/dev/sda5       19992176 10188092   8765492  54% /
tmpfs           6120416         0   6120416   0% /dev/shm
tmpfs           5120          4      5116    1% /run/lock
tmpfs           6120416         0   6120416   0% /sys/fs/cgroup
/dev/loop0       9344        9344         0 100% /snap/canonical-livepatch/95
/dev/loop2      100224     100224         0 100% /snap/core/10577
/dev/loop3      224256     224256         0 100% /snap/gnome-3-34-1804/66
/dev/loop4      100352     100352         0 100% /snap/core/10583
/dev/loop5      223232     223232         0 100% /snap/gnome-3-34-1804/60
/dev/loop1      147328     147328         0 100% /snap/code/52
/dev/loop6       31872      31872         0 100% /snap/snapd/10492
/dev/loop7       56704      56704         0 100% /snap/core18/1932
/dev/loop8       31872      31872         0 100% /snap/snapd/10707
/dev/loop9       52352      52352         0 100% /snap/snap-store/498
/dev/loop10      56832      56832         0 100% /snap/core18/1944
/dev/loop11      65920      65920         0 100% /snap/gtk-common-themes/1513
/dev/loop12      52352      52352         0 100% /snap/snap-store/518
/dev/loop13      66432      66432         0 100% /snap/gtk-common-themes/1514
/dev/sdb         1015480      2564    944104   1% /newmount
/dev/sda1        523248         4    523244   1% /boot/efi
tmpfs           1224080         24   1224056   1% /run/user/1000

```

reboot 後能用 df 看到 mount 成功了/dev/sdb 成功 mount 在/newmount（反白那行）。

Task3 — Inode and block

1. ext4 filesystem 每一個 directory 需要一個 inode 值，而 Task2 開的 inode 有 896 個，系統保留有 10 個，所以有最多 885 個可創建的 directory，下面是 mkdir 創建的實作，只能開 886 個。

```
sudo mkdir test{1..900}
mkdir: cannot create directory 'test887': No space left on device
mkdir: cannot create directory 'test888': No space left on device
mkdir: cannot create directory 'test889': No space left on device
mkdir: cannot create directory 'test890': No space left on device
mkdir: cannot create directory 'test891': No space left on device
mkdir: cannot create directory 'test892': No space left on device
mkdir: cannot create directory 'test893': No space left on device
mkdir: cannot create directory 'test894': No space left on device
mkdir: cannot create directory 'test895': No space left on device
mkdir: cannot create directory 'test896': No space left on device
mkdir: cannot create directory 'test897': No space left on device
mkdir: cannot create directory 'test898': No space left on device
mkdir: cannot create directory 'test899': No space left on device
mkdir: cannot create directory 'test900': No space left on device
```

2. 就算裡面只存了 1byte 的資料，每個 file/directory 需要至少一個 inode，所以最多 file 數量一樣為 Task2 開的 inode 數 886 個。同時會有很多空間是空的，因為每 4096bytes 只用了 1byte，以下是我寫的 shell script，讓每次 for 都會 fallocate 出一個 1Byte 大小的檔案(for 裡面是 270000 是因為我以為會是 file blocks 的大小)。

```
#!/bin/bash
for i in {1..270000}
do
    sudo fallocate -l 1 ${i}
done
```

```

linyouxin@ubuntu:/newmount$ sudo ./create.sh
fallocate: cannot open 886: No space left on device
fallocate: cannot open 887: No space left on device
fallocate: cannot open 888: No space left on device
fallocate: cannot open 889: No space left on device
fallocate: cannot open 890: No space left on device
fallocate: cannot open 891: No space left on device
fallocate: cannot open 892: No space left on device
fallocate: cannot open 893: No space left on device
fallocate: cannot open 894: No space left on device
fallocate: cannot open 895: No space left on device
fallocate: cannot open 896: No space left on device
fallocate: cannot open 897: No space left on device
fallocate: cannot open 898: No space left on device
fallocate: cannot open 899: No space left on device
fallocate: cannot open 900: No space left on device

```

(上圖 886 就無法創建是因為裡面還有一個 create.sh，但其實是可以創 886 個檔案的)

- ext4 filesystem 理論上的 max file size 16TiB，遠遠超過我在 Task2 所開的空間，所以在 Task2 的情況下最大的檔案能佔滿我所開的所有空間，為約 500MB。

```

linyouxin@ubuntu:/newmount$ sudo fallocate -l 500M test
linyouxin@ubuntu:/newmount$ ls -l
total 996532
-rw-r--r-- 1 root root 1020444672 Jan 10 06:46 test
linyouxin@ubuntu:/newmount$ df -h

```

Filesystem	Size	Used	Avail	Use%	Mounted on
udev	5.9G	0	5.9G	0%	/dev
tmpfs	1.2G	1.9M	1.2G	1%	/run
/dev/sda5	20G	9.8G	8.4G	54%	/
tmpfs	5.9G	0	5.9G	0%	/dev/shm
tmpfs	5.0M	4.0K	5.0M	1%	/run/lock
tmpfs	5.9G	0	5.9G	0%	/sys/fs/cgroup
/dev/loop0	9.2M	9.2M	0	100%	/snap/canonical-livepatch/95
/dev/loop3	98M	98M	0	100%	/snap/core/10577
/dev/loop2	98M	98M	0	100%	/snap/core/10583
/dev/loop4	56M	56M	0	100%	/snap/core18/1944
/dev/loop1	144M	144M	0	100%	/snap/code/52
/dev/loop5	56M	56M	0	100%	/snap/core18/1932
/dev/loop6	219M	219M	0	100%	/snap/gnome-3-34-1804/66
/dev/loop7	65M	65M	0	100%	/snap/gtk-common-themes/1513
/dev/loop8	218M	218M	0	100%	/snap/gnome-3-34-1804/60
/dev/loop9	52M	52M	0	100%	/snap/snap-store/518
/dev/loop10	32M	32M	0	100%	/snap/snapd/10707
/dev/loop11	65M	65M	0	100%	/snap/gtk-common-themes/1514
/dev/loop12	52M	52M	0	100%	/snap/snap-store/498
/dev/loop13	32M	32M	0	100%	/snap/snapd/10492
/dev/sdb	992M	976M	0	100%	/newmount
/dev/sda1	511M	4.0K	511M	1%	/boot/efi
tmpfs	1.2G	28K	1.2G	1%	/run/user/1000

(上圖中 500MB 的檔案佔滿了我 mount 的硬碟，反白那行)