OS HW3 0711529 陳冠儒

Task1 - Soft Link and Hard Link (30%)

1. Hard link

- 理論
  - hard link 原理
  - 每個檔案都會占用一個 inode，檔案內容由 inode 的紀錄來指向。
  - 想要讀取該檔案，必須要經過目錄紀錄的檔名來指向到正確的 inode 號碼才能讀取。

  故 hard link 就是在某個目錄下新增一筆檔名連結到某 inode 號碼的關聯紀錄。

  - hard link 優點
  - 安全：因為多個檔名指向相同內容，所以即使將任一檔名刪除，inode 跟 block 都還是存在

  - hard link 限制
  - 不能跨 Filesystem
  - 不能 link 目錄

  - hard link 空間使用

    不會改變，因為只是在某目錄 block 中多寫入一個資料而已，故不會增加 node 也不會耗用 block 數量，除非該目錄 block 剛好被填滿，才需要新加一個 block。

- 實做
  - hard link from file1.txt to file2.tx

```
ubuntu@ip-172-31-46-217:~/work/os$ echo hello > file1.txt
ubuntu@ip-172-31-46-217:~/work/os$ cat file1.txt
hello
ubuntu@ip-172-31-46-217:~/work/os$ ll -i ./file1.txt
516659 -rw-rw-r-- 1 ubuntu ubuntu 6 Jan  8 13:56 ./file1.txt
```

首先先建立一個 file1.txt，並且內容為 hello。觀察他的資訊可以看到他的 inode 為 516659。

```
ubuntu@ip-172-31-46-217:~/work/os$ ln ./file1.txt file2.txt
```

接著建立一個 hard link from file1.txt to file2.txt

```
ubuntu@ip-172-31-46-217:~/work/os$ ll -i file1.txt file2.txt
516659 -rw-rw-r-- 2 ubuntu ubuntu 6 Jan  8 13:56 file1.txt
516659 -rw-rw-r-- 2 ubuntu ubuntu 6 Jan  8 13:56 file2.txt
ubuntu@ip-172-31-46-217:~/work/os$ cat file2.txt
hello
```

我們可以發現 file2.txt 的 inode 會變得跟 file1.txt 一樣都是 516659，所以兩個的內容會是一樣的，並且可以看到第二欄的 link number 從原本的 1 變成了 2。

```
ubuntu@ip-172-31-46-217:~/work/os$ echo This is file2! > file2.txt
ubuntu@ip-172-31-46-217:~/work/os$ cat file1.txt
This is file2!
```

最後我們試著改變 file2.txt 的內容，可以發現 file1.txt 的內容亦會跟著改變，因為 file1.txt 跟 file2.txt 指向的是同一個 inode，而 inode 會指向一塊 block，因此 file1.txt 和 file2.txt 是使用相同 inode 與 block。

```
ubuntu@ip-172-31-46-217:~/work/os$ rm file1.txt
ubuntu@ip-172-31-46-217:~/work/os$ ll -i
total 12
516655 drwxrwxr-x 2 ubuntu ubuntu 4096 Jan  8 13:58 ./
256112 drwxrwxr-x 4 ubuntu ubuntu 4096 Jan  3 03:27 ../
516659 -rw-rw-r-- 1 ubuntu ubuntu   15 Jan  8 13:57 file2.txt
ubuntu@ip-172-31-46-217:~/work/os$ cat file2.txt
This is file2!
```

最後將 file1.txt 刪除後，可以看到 file2.txt 依然存在並且 link number 變回了 1，可以證實 hard link 的「安全性」，即使將一個檔名刪除，他的 inode 和 block 都還會存在。

- hard link 目錄

```
ubuntu@ip-172-31-46-217:~/work/os$ mkdir dir1
ubuntu@ip-172-31-46-217:~/work/os$ ln dir1 dir2
ln: dir1: hard link not allowed for directory
```

可以看出無法 hard link 目錄，因為如果要 hard link 目錄，那底下的資料也都要 hard link 一次，會有很大的複雜度。

- hard link 空間使用

```
ubuntu@ip-172-31-46-217:~/work/os$ touch file1.txt
ubuntu@ip-172-31-46-217:~/work/os$ du -sb; df -i .
4096    .
Filesystem      Inodes  IUsed  IFree IUse% Mounted on
/dev/root      1024000 100965 923035   10% /
ubuntu@ip-172-31-46-217:~/work/os$ ln file1.txt file2.txt
ubuntu@ip-172-31-46-217:~/work/os$ du -sb; df -i .
4096    .
Filesystem      Inodes  IUsed  IFree IUse% Mounted on
/dev/root      1024000 100965 923035   10% /
```

使用 hard link 絕大部分都不會增加額外空間(額外 indoe 和 block)。

2. Soft link
   - 理論
     - soft link 原理

       Linux 中的 soft link 又稱為 symbolic link，他就是再建立一個獨立的檔案，而這個檔案會讓資料的讀取指向他 link 的那個檔案的檔名。

- soft link 空間使用

  Symbolic link 與 Windows 的捷徑幾乎是相同的，故由 Symbolic link 所建立的檔案為一個獨立的新檔案，所以會占用掉 inode 與 block。

- 實做
  - soft link from file1.txt to file3.txt

```
ubuntu@ip-172-31-46-217:~/work/os$ echo Welcome to file1! > file1.txt
ubuntu@ip-172-31-46-217:~/work/os$ ln -s file1.txt file3.txt
ubuntu@ip-172-31-46-217:~/work/os$ cat file3.txt
Welcome to file1!
ubuntu@ip-172-31-46-217:~/work/os$ echo Welcome to file3! > file3.txt
ubuntu@ip-172-31-46-217:~/work/os$ cat file1.txt
Welcome to file3!
```

建立一個 file1.txt，內容為「Welcome to file1!」，並建立一個 file3.txt 的 soft link 指向 file1.txt，可以 read/write file1.txt 的內容。

```
ubuntu@ip-172-31-46-217:~/work/os$ ll -i file1.txt  file3.txt
516680 -rw-rw-r-- 1 ubuntu ubuntu 18 Jan  8 16:55 file1.txt
516681 lrwxrwxrwx 1 ubuntu ubuntu  9 Jan  8 16:55 file3.txt -> file1.txt
```

file1.txt 和 file3.txt 的 inode 是不一樣的，並且用了 soft link 後，他們的 link number 都不會增加，還是保持著 1。而可以注意到 file3.txt 的檔案大小為 9bytes，是因為「file3.txt」這個檔名有 9 個英文字，一個英文字佔了一個 bytes。

```
ubuntu@ip-172-31-46-217:~/work/os$ rm file1.txt
ubuntu@ip-172-31-46-217:~/work/os$ ll -i file3.txt
516681 lrwxrwxrwx 1 ubuntu ubuntu 9 Jan  8 16:55 file3.txt -> file1.txt
ubuntu@ip-172-31-46-217:~/work/os$ cat file3.txt
cat: file3.txt: No such file or directory
```

將 file1.txt 刪除後，file3.txt 會失效，因為他指向的 file1.txt 的檔名不見了。

  - soft link 目錄

```
ubuntu@ip-172-31-46-217:~/work/os$ mkdir dir1
ubuntu@ip-172-31-46-217:~/work/os$ ln -s dir1 dir3
ubuntu@ip-172-31-46-217:~/work/os$ ll -i
total 12
516655 drwxrwxr-x 3 ubuntu ubuntu 4096 Jan  8 17:07 ./
256112 drwxrwxr-x 4 ubuntu ubuntu 4096 Jan  3 03:27 ../
516680 drwxrwxr-x 2 ubuntu ubuntu 4096 Jan  8 17:07 dir1/
516681 lrwxrwxrwx 1 ubuntu ubuntu    4 Jan  8 17:07 dir3 -> dir1/
```

soft link 可以 link 目錄。

  - soft link 空間使用

```
ubuntu@ip-172-31-46-217:~/work/os$ echo Welcome to file1! > file1.txt
ubuntu@ip-172-31-46-217:~/work/os$ du -sb; df -i .
4114    .
Filesystem      Inodes  IUsed  IFree IUse% Mounted on
/dev/root      1024000 100965 923035   10% /
ubuntu@ip-172-31-46-217:~/work/os$ ln -s file1.txt file3.txt
ubuntu@ip-172-31-46-217:~/work/os$ du -sb; df -i .
4123    .
Filesystem      Inodes  IUsed  IFree IUse% Mounted on
/dev/root      1024000 100966 923034   10% /
```

可以看到 inode 使用量多了 1，且空間使用從 4114 bytes 變成了 4123 bytes，增加了 9 bytes，即為 file3.txt 檔名的大小。

Task2 - Creating and mounting file system (30%)

```
ubuntu@ip-172-31-89-222:~$ sudo fdisk -l
Disk /dev/loop0: 97.8 MiB, 102486016 bytes, 200168 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


Disk /dev/loop1: 28.1 MiB, 29454336 bytes, 57528 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


Disk /dev/xvda: 8 GiB, 8589934592 bytes, 16777216 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: 0x197df13e

Device     Boot Start      End  Sectors Size Id Type
/dev/xvda1 *     2048 16777182 16775135   8G 83 Linux


Disk /dev/xvdb: 8 GiB, 8589934592 bytes, 16777216 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


Disk /dev/xvdc: 8 GiB, 8589934592 bytes, 16777216 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
```

首先，我們用 sudo fdisk -l 觀察要選哪一個磁碟做分割，可以發現 /dev/xvdb 跟/dev/xvdc 都還沒被分割過，故我們選用這兩個其中的 /dev/xvdb 來進行分割。

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

```
ubuntu@ip-172-31-89-222:~$ sudo fdisk /dev/xvdb

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 0x0f13d6fb.

Command (m for help): p
Disk /dev/xvdb: 8 GiB, 8589934592 bytes, 16777216 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: 0x0f13d6fb
```

我們先用指令「p」看看這個磁碟的分割情況，上述已經提過他尚未被分割。

```
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 (2048-16777215, default 2048):
Last sector, +sectors or +size{K,M,G,T,P} (2048-16777215, default 16777215):

Created a new partition 1 of type 'Extended' and of size 8 GiB.

Command (m for help): p
Disk /dev/xvdb: 8 GiB, 8589934592 bytes, 16777216 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: 0x0f13d6fb

Device     Boot Start      End  Sectors Size Id Type
/dev/xvdb1       2048 16777215 16775168   8G  5 Extended
```

接著在建立 logical partition 前，我們先用指令「n」建立一個 Extended partition，Extended partition 可以被切成很多個 logical partition。

```
Command (m for help): n
All space for primary partitions is in use.
Adding logical partition 5
First sector (4096-16777215, default 4096):
Last sector, +sectors or +size{K,M,G,T,P} (4096-16777215, default 16777215): +500M

Created a new partition 5 of type 'Linux' and of size 500 MiB.

Command (m for help): p
Disk /dev/xvdb: 8 GiB, 8589934592 bytes, 16777216 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: 0x0f13d6fb

Device     Boot Start      End  Sectors  Size Id Type
/dev/xvdb1       2048 16777215 16775168    8G  5 Extended
/dev/xvdb5       4096  1028095  1024000  500M 83 Linux

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

用指令「n」建立一個 500M 的 logical partition，其他不用更改的地方，都用 default 值，指令「p」確認已建立後，輸入「w」讓他 write to disk

and exit。

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

```
ubuntu@ip-172-31-89-222:~$ sudo fdisk -l /dev/xvdb
Disk /dev/xvdb: 8 GiB, 8589934592 bytes, 16777216 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: 0x0f13d6fb

Device     Boot Start      End  Sectors  Size Id Type
/dev/xvdb1       2048 16777215 16775168    8G  5 Extended
/dev/xvdb5       4096  1028095  1024000  500M 83 Linux
```

用 sudo fdisk -l 可以看到剛剛建立好的 Extended partition /dev/xvdb1 跟 logical partition /dev/xvdb5。

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

```
ubuntu@ip-172-31-89-222:~$ sudo mkfs.ext4 -b 4096 -N 800 /dev/xvdb5
mke2fs 1.44.1 (24-Mar-2018)
Creating filesystem with 128000 4k blocks and 896 inodes
Filesystem UUID: 46cd300f-9b09-4a0b-838d-6a72bb4cec17
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
```

用 mkfs.ext4 去 format partition，參數-b <block size>、-N <iNode number>，但因為分成了四個 Block Group，而前 11 個 inode 為 Special Inode 被 ext4 保留以做些特殊的用途，剩下的要平均分配給內四個 Block Group，因為 800 不能被平均分配，所以才不會是準確剛好 800 個 inode。

```
ubuntu@ip-172-31-89-222:~$ sudo dumpe2fs /dev/xvdb5
dumpe2fs 1.44.1 (24-Mar-2018)
Filesystem volume name:   <none>
Last mounted on:          <not available>
Filesystem UUID:          46cd300f-9b09-4a0b-838d-6a72bb4cec17
Filesystem magic number:  0xEF53
Filesystem revision #:    1 (dynamic)
Filesystem features:      has_journal ext_attr resize_inode dir_index filetype extent 64bi
t flex_bg sparse_super large_file huge_file dir_nlink extra_isize metadata_csum
Filesystem flags:         signed_directory_hash
Default mount options:    user_xattr acl
Filesystem state:         clean
Errors behavior:          Continue
Filesystem OS type:       Linux
Inode count:              896
Block count:              128000
Reserved block count:     6400
Free blocks:              123670
Free inodes:              885
First block:              0
Block size:               4096
Fragment size:            4096
Group descriptor size:    64
Reserved GDT blocks:      62
Blocks per group:         32768
Fragments per group:      32768
Inodes per group:         224
Inode blocks per group:   7
Flex block group size:    16
Filesystem created:       Sat Jan  9 18:09:46 2021
Last mount time:          n/a
Last write time:          Sat Jan  9 18:09:46 2021
Mount count:              0
```

我們可以用 dumpe2fs 來確認該 partition 的 superblock 資訊。

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

```
ubuntu@ip-172-31-89-222:~$ sudo vi /etc/fstab
ubuntu@ip-172-31-89-222:~$ cat /etc/fstab
LABEL=cloudimg-rootfs   /        ext4   defaults,discard        0 0
/dev/xvdb5              /data2   ext4   defaults                0 0
```

用 vi 修改/etc/fstab，讓他每次開機都能自動掛載，我讓他掛載到/data2 這個地方。

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

```
ubuntu@ip-172-31-89-222:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            476M     0  476M   0% /dev
tmpfs            98M  780K   98M   1% /run
/dev/xvda1      7.7G  1.2G  6.6G  15% /
tmpfs           490M     0  490M   0% /dev/shm
tmpfs           5.0M     0  5.0M   0% /run/lock
tmpfs           490M     0  490M   0% /sys/fs/cgroup
/dev/loop0       29M   29M     0 100% /snap/amazon-ssm-agent/2012
/dev/loop1       98M   98M     0 100% /snap/core/10185
/dev/xvdb5      484M  768K  449M   1% /data2
tmpfs            98M     0   98M   0% /run/user/1000
```

最後利用 df 去確認他是否有正確被掛載上去，可以看到是正確的！

Task3 – Inode and block (40%)

According to the file system in the Task2, implement and answer the following questions

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)

```
ubuntu@ip-172-31-89-222:/data2$ df -i /dev/xvdb5
Filesystem     Inodes IUsed IFree IUse% Mounted on
/dev/xvdb5        896    11   885    2% /data2
ubuntu@ip-172-31-89-222:/data2$ sudo mkdir dir{1..896}
mkdir: cannot create directory ‘dir886’: No space left on device
mkdir: cannot create directory ‘dir887’: No space left on device
mkdir: cannot create directory ‘dir888’: No space left on device
mkdir: cannot create directory ‘dir889’: No space left on device
mkdir: cannot create directory ‘dir890’: No space left on device
mkdir: cannot create directory ‘dir891’: No space left on device
mkdir: cannot create directory ‘dir892’: No space left on device
mkdir: cannot create directory ‘dir893’: No space left on device
mkdir: cannot create directory ‘dir894’: No space left on device
mkdir: cannot create directory ‘dir895’: No space left on device
mkdir: cannot create directory ‘dir896’: No space left on device
ubuntu@ip-172-31-89-222:/data2$ df -i /dev/xvdb5
Filesystem     Inodes IUsed IFree IUse% Mounted on
/dev/xvdb5        896   896     0  100% /data2
ubuntu@ip-172-31-89-222:/data2$
```

在建 directories 前，我先用 df -i /dev/xvdb5 看總共有多少 inode 數量和已經被用了多少。可以看到總共有 896 個 inode，有 11 個 special inodes 被使用，只剩下了 885 個未被使用的。
接著用 sudo mkdir dir{1...896}建立大量的 directories，可以看到最後總共建了 885 個，跟前述未被使用過的 inode 數量是一樣的。
最後再用 df 來檢查看看是不是所有 inode 都被使用了，此時 IFree 為 0，代表都被用掉了。

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)

```
ubuntu@ip-172-31-89-222:/data2$ df -h /dev/xvdb5
Filesystem      Size  Used Avail Use% Mounted on
/dev/xvdb5      484M  772K  449M   1% /data2
ubuntu@ip-172-31-89-222:/data2$ for i in {1..896}; do sudo fallocate -l 1 file_${i}; done;
fallocate: cannot open file_886: No space left on device
fallocate: cannot open file_887: No space left on device
fallocate: cannot open file_888: No space left on device
fallocate: cannot open file_889: No space left on device
fallocate: cannot open file_890: No space left on device
fallocate: cannot open file_891: No space left on device
fallocate: cannot open file_892: No space left on device
fallocate: cannot open file_893: No space left on device
fallocate: cannot open file_894: No space left on device
fallocate: cannot open file_895: No space left on device
fallocate: cannot open file_896: No space left on device
ubuntu@ip-172-31-89-222:/data2$ df -h /dev/xvdb5
Filesystem      Size  Used Avail Use% Mounted on
/dev/xvdb5      484M  4.3M  445M   1% /data2
ubuntu@ip-172-31-89-222:/data2$ df -i /dev/xvdb5
Filesystem     Inodes IUsed IFree IUse% Mounted on
/dev/xvdb5        896   896     0  100% /data2
ubuntu@ip-172-31-89-222:/data2$
```

我們先用 df -h /dev/xvdb5 來查看總共有多少可以使用的空間，目前我們還有 449M 可以使用，再用 for i in {1..896}; do sudo fallocate -l 1 file_${i}; done; 去建立 896 個 1-byte file，會發現最終只能建出 885

個，跟上面那題的結果是一樣的，因為一個檔案會占用一個 inode，而當 inode 被用完後就無法再建立檔案了，所以即使每個 block 裡面還有相當大的空間，但依然無法建立，這就是所謂的 internal fragmentation。

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?

```
ubuntu@ip-172-31-89-222:/data2$ df -h /dev/xvdb5
Filesystem      Size  Used Avail Use% Mounted on
/dev/xvdb5      484M  772K  449M   1% /data2
ubuntu@ip-172-31-89-222:/data2$ sudo fallocate -l 506552320 file
fallocate: fallocate failed: No space left on device
ubuntu@ip-172-31-89-222:/data2$ df -h /dev/xvdb5
Filesystem      Size  Used Avail Use% Mounted on
/dev/xvdb5      484M  474M     0 100% /data2
ubuntu@ip-172-31-89-222:/data2$ ll
total 484464
drwxr-xr-x  3 root root     20480 Jan 10 07:27 ./
drwxr-xr-x 24 root root      4096 Jan 10 06:34 ../
-rw-r--r--  1 root root 496058368 Jan 10 07:27 file
drwxr-xr-x  2 root root      4096 Jan 10 05:42 lost+found/
```

最大可以建立一個大小為 473M (496058368 bytes)的檔案，將所有可以使用的空間都使用完了，但是發現並不能使用掉所有 file system 的空間，總空間有 484M，總使用量為 474M，有 10M 是被特殊保留起來的。