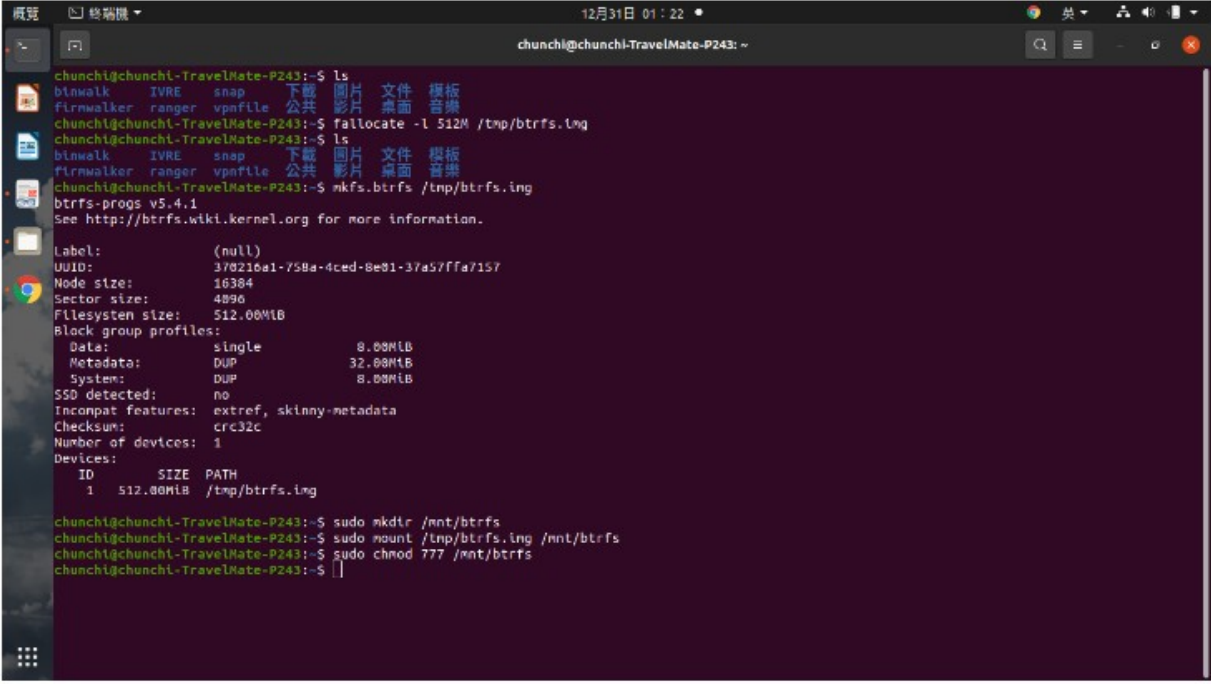


1.先建立一個 subvolume，當我們建立快照時可以發現存在兩個 subvolume。在建立的 snapshot 文件裡，可以發現我們看得到 dir1 裡的東西，卻無法看到 sub1 裡的東西，因為 sub1 為一個 subvolume。而我們對 subvolume 做快照時，並不會將 subvolume 裡的東西也做快照。而當我們對 sub1 進行快照時，可以看到 sub1 的東西出現在快照內。最終 snap-sub1 以及 sub1 底下個建立一個文件，都不會互相影響。

搭建環境



```
chunchi@chunchi-TravelMate-P243:~$ ls
binwalk  IVRE  snap  下載  圖片  文件  模板
firmwalker  ranger  vpnfile  公共  影片  桌面  音樂
chunchi@chunchi-TravelMate-P243:~$ fallocation -l 512M /tmp/btrfs.img
chunchi@chunchi-TravelMate-P243:~$ ls
binwalk  IVRE  snap  下載  圖片  文件  模板
firmwalker  ranger  vpnfile  公共  影片  桌面  音樂
chunchi@chunchi-TravelMate-P243:~$ mkfs.btrfs /tmp/btrfs.img
btrfs-progs v5.4.1
See http://btrfs.wiki.kernel.org for more information.

Label:                (null)
UUID:                 378216a1-758a-4ced-8e81-37a57ffa7157
Node size:            16384
Sector size:          4096
Filesystem size:      512.00MiB
Block group profiles:
  Data:               single          8.00MiB
  Metadata:           DUP             32.00MiB
  System:             DUP             8.00MiB
SSD detected:         no
Incompat features:    extref, skinny-metadata
Checksum:             crc32c
Number of devices:    1
Devices:
  ID     SIZE  PATH
  1     512.00MiB  /tmp/btrfs.img

chunchi@chunchi-TravelMate-P243:~$ sudo mkdir /mnt/btrfs
chunchi@chunchi-TravelMate-P243:~$ sudo mount /tmp/btrfs.img /mnt/btrfs
chunchi@chunchi-TravelMate-P243:~$ sudo chmod 777 /mnt/btrfs
chunchi@chunchi-TravelMate-P243:~$
```

```
fallocation -l 512M /tmp/btrfs.img

mkfs.btrfs /tmp/btrfs.img

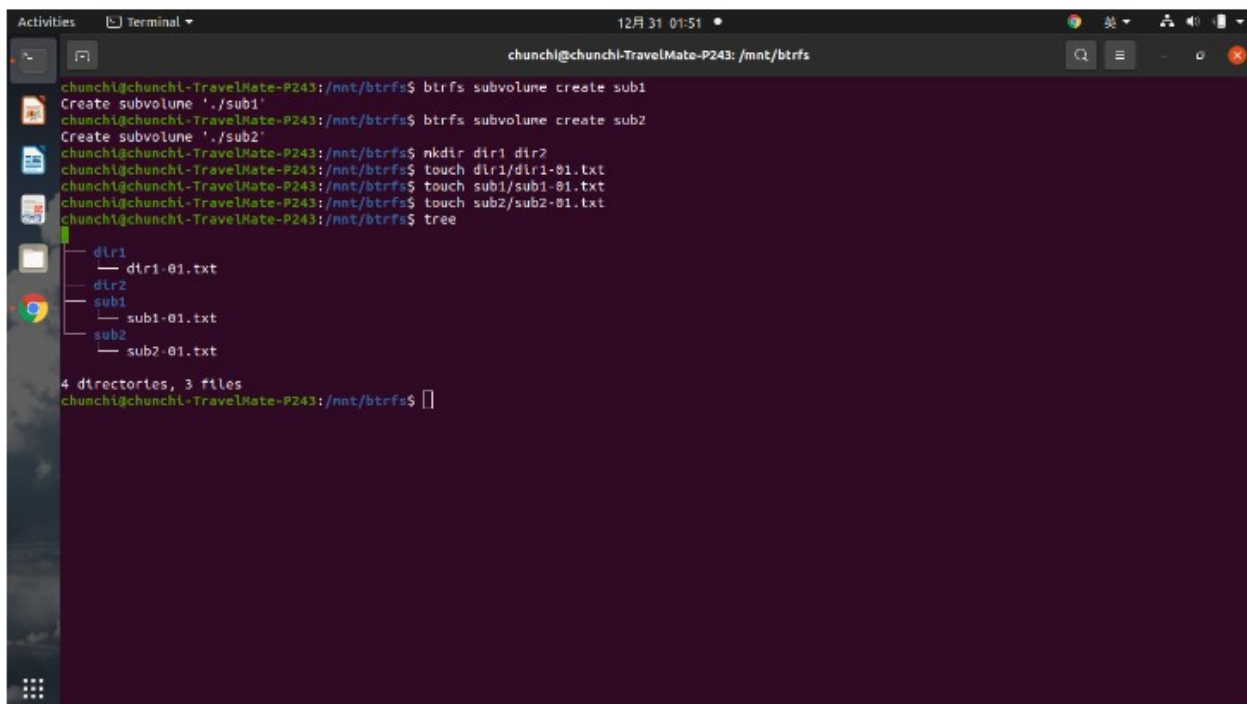
sudo mkdir /mnt/btrfs

sudo mount /tmp/btrfs.img /mnt/btrfs

sudo chmod 777 /mnt/btrfs

mount|grep btrfs
```

建立subvolume

A terminal window titled 'chunchi@chunchi-TravelMate-P243: /mnt/btrfs' showing the execution of several commands to create subvolumes and files. The commands are: 'btrfs subvolume create sub1', 'btrfs subvolume create sub2', 'mkdir dir1 dir2', 'touch dir1/dir1-01.txt', 'touch sub1/sub1-01.txt', and 'touch sub2/sub2-01.txt'. The 'tree' command is used to show the directory structure, which lists 'dir1', 'dir2', 'sub1', and 'sub2' as directories, each containing a file named '-01.txt'. The output indicates there are 4 directories and 3 files.

```
chunchi@chunchi-TravelMate-P243:/mnt/btrfs$ btrfs subvolume create sub1
Create subvolume './sub1'
chunchi@chunchi-TravelMate-P243:/mnt/btrfs$ btrfs subvolume create sub2
Create subvolume './sub2'
chunchi@chunchi-TravelMate-P243:/mnt/btrfs$ mkdir dir1 dir2
chunchi@chunchi-TravelMate-P243:/mnt/btrfs$ touch dir1/dir1-01.txt
chunchi@chunchi-TravelMate-P243:/mnt/btrfs$ touch sub1/sub1-01.txt
chunchi@chunchi-TravelMate-P243:/mnt/btrfs$ touch sub2/sub2-01.txt
chunchi@chunchi-TravelMate-P243:/mnt/btrfs$ tree
.
├── dir1
│   └── dir1-01.txt
├── dir2
├── sub1
│   └── sub1-01.txt
└── sub2
    └── sub2-01.txt

4 directories, 3 files
chunchi@chunchi-TravelMate-P243:/mnt/btrfs$
```

```
cd /mnt/btrfs

btrfs subvolume create sub1

btrfs subvolume create sub2

mkdir dir1 dir2

touch dir1/dir1-01.txt

touch sub1/sub1-01.txt

touch sub2/sub2-01.txt

tree
```

建立snapshot

```
Activities Terminal 12月 31 01:53
chunchi@chunchi-TravelMate-P243: /mnt/btrfs

sub2-01.txt

4 directories, 3 files
chunchi@chunchi-TravelMate-P243:/mnt/btrfs$ sudo btrfs subvolume snapshot ./ ./snap-root
Create a snapshot of './' in './snap-root'
chunchi@chunchi-TravelMate-P243:/mnt/btrfs$ sudo btrfs subvolume list ./
ID 259 gen 21 top level 5 path sub1
ID 260 gen 21 top level 5 path sub2
ID 261 gen 22 top level 5 path snap-root
chunchi@chunchi-TravelMate-P243:/mnt/btrfs$ sudo btrfs subvolume list -s ./
ID 261 gen 22 cgen 22 top level 5 otime 2020-12-31 01:52:13 path snap-root
chunchi@chunchi-TravelMate-P243:/mnt/btrfs$ tree ./snap-root
tree
├── dir1
│   └── dir1-01.txt
├── dir2
├── sub1
└── sub2

4 directories, 1 file
chunchi@chunchi-TravelMate-P243:/mnt/btrfs$ sudo btrfs subvolume snapshot ./sub1/ ./snap-sub1
Create a snapshot of './sub1/' in './snap-sub1'
chunchi@chunchi-TravelMate-P243:/mnt/btrfs$ touch snap-sub1/snap-sub1-01.txt
chunchi@chunchi-TravelMate-P243:/mnt/btrfs$ touch sub1/sub1-02.txt
chunchi@chunchi-TravelMate-P243:/mnt/btrfs$ tree
tree
├── dir1
│   └── dir1-01.txt
├── dir2
├── snap-sub1
│   ├── snap-sub1-01.txt
│   └── sub1-01.txt
├── sub1
│   ├── sub1-01.txt
│   ├── sub1-02.txt
│   └── sub2
└── sub2
    └── sub2-01.txt

10 directories, 7 files
chunchi@chunchi-TravelMate-P243:/mnt/btrfs$
```

```
tree
├── dir1
│   └── dir1-01.txt
├── dir2
├── snap-sub1
│   ├── snap-sub1-01.txt
│   └── sub1-01.txt
├── sub1
│   ├── sub1-01.txt
│   ├── sub1-02.txt
│   └── sub2
└── sub2
    └── sub2-01.txt

10 directories, 7 files
chunchi@chunchi-TravelMate-P243:/mnt/btrfs$
```

```
sudo btrfs subvolume snapshot ./ ./snap-root

sudo btrfs subvolume list ./

sudo btrfs subvolume list -s ./

tree ./snap-root

sudo btrfs subvolume snapshot ./sub1/ ./snap-sub1

touch snap-sub1/snap-sub1-01.txt

touch sub1/sub1-02.txt

tree
```

2.寫一個程式，各對 fileA 還有 fileB 做開檔寫檔的動作，而讓它延遲 20 秒。導致他在 tar 備份的過程中，可能會發生僅有 fileA 做備份，但 fileB 為備份到的情況，即導致發生一致性的問題。

The image shows a terminal window with a dark background and light-colored text. The window title is "chunchi@chunchi-TravelMate-P243: ~/hw10". The terminal displays a C program that demonstrates file synchronization using `fopen`, `fwrite`, `fflush`, `fsync`, and `fclose`. The program uses file descriptors to manage buffers and file pointers. Comments in Chinese explain the process of writing to a buffer, flushing it, and then syncing it to the disk.

```
#include <stdio.h>
#include <string.h>
#include <unistd.h>
#include <stdlib.h>
int main(){
    FILE *fp1 = fopen("DIR/fileA", "a+");
    int fd1 = -1;
    for(long long int i = 1 ; i <= 10000000 ; i++){
        if(fp1){
            fwrite("input!", 1, 6, fp1);
            fflush(fp1); //寫到緩衝區
            fd1 = fileno(fp1); //由檔案流獲取檔案描述符
        }
    }
    if(fd1 != -1)
        fsync(fd1); //由緩衝區寫到磁碟
    fclose(fp1);

    FILE *fp2 = fopen("DIR/fileB", "a+");
    int fd2 = -1;
    for(long long int i = 1 ; i <= 10000000 ; i++){
        if(fp2){
            fwrite("input!", 1, 6, fp2);
            fflush(fp2); //寫到緩衝區
            fd2 = fileno(fp2); //由檔案流獲取檔案描述符
        }
    }
    if(fd2 != -1)
        fsync(fd2); //由緩衝區寫到磁碟
    fclose(fp2);

    return 0;
}
```

At the bottom of the terminal, the command `"insert.c" 33L, 753C` is visible, indicating the current line and column in the file.

A screenshot of a Linux desktop environment. The top panel shows the 'Activities' button, a 'Terminal' window icon, and the system clock displaying '12月 31 05:30'. On the right side of the top panel, there are icons for network, volume, and power. The desktop background is a dark purple color. A terminal window is open, showing the command prompt 'chunchi@chunchi-TravelMate-P243: ~/hw10'. The terminal contains the following C code:

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
int main(){
    int ret = system("./insert");
    sleep(20);
    ret = system("tar -zcvf test.tar.gz DIR/");
    return 0;
}
```

The terminal window has a search icon, a menu icon, and window control buttons (minimize, maximize, close) on the right. On the left side of the terminal window, there is a vertical dock with icons for various applications, including a file manager, a web browser, and a terminal. The bottom status bar shows the text '"trans.c" 10L, 193C' on the left, '7,11-18' in the center, and 'All' on the right.

```
Activities Terminal 12月 31 05:30
chunchi@chunchi-TravelMate-P243: ~/hw10/DIR
GNU coreutils online help: <https://www.gnu.org/software/coreutils/>
Full documentation at: <https://www.gnu.org/software/coreutils/stat>
or available locally via: info '(coreutils) stat invocation'
chunchi@chunchi-TravelMate-P243:~/hw10$ ls
DIR insert insert.c makefile test test.c test.tar.gz trans trans.c
chunchi@chunchi-TravelMate-P243:~/hw10$ ls -l
total 628
drwxrwxr-x 2 chunchi chunchi 4096 12月 31 03:58 DIR
-rwxrwxr-x 1 chunchi chunchi 21024 12月 31 04:26 insert
-rw-rw-r-- 1 chunchi chunchi 753 12月 31 04:26 insert.c
-rw-rw-r-- 1 chunchi chunchi 176 12月 31 03:37 makefile
-rwxrwxr-x 1 chunchi chunchi 20064 12月 31 04:27 test
-rw-rw-r-- 1 chunchi chunchi 108 12月 31 04:24 test.c
-rw-rw-r-- 1 chunchi chunchi 556476 12月 31 04:28 test.tar.gz
-rwxrwxr-x 1 chunchi chunchi 19744 12月 31 04:26 trans
-rw-rw-r-- 1 chunchi chunchi 193 12月 31 04:26 trans.c
chunchi@chunchi-TravelMate-P243:~/hw10$ ls -t
test.tar.gz test trans trans.c insert insert.c test.c DIR makefile
chunchi@chunchi-TravelMate-P243:~/hw10$ ls
DIR insert insert.c makefile test test.c test.tar.gz trans trans.c
chunchi@chunchi-TravelMate-P243:~/hw10$ ls
DIR insert insert.c makefile test test.c test.tar.gz trans trans.c
chunchi@chunchi-TravelMate-P243:~/hw10$ tar -xvf test.tar.gz
DIR/
DIR/fileB
DIR/fileA
chunchi@chunchi-TravelMate-P243:~/hw10$ ls
DIR insert insert.c makefile test test.c test.tar.gz trans trans.c
chunchi@chunchi-TravelMate-P243:~/hw10$ cd DIR/
chunchi@chunchi-TravelMate-P243:~/hw10/DIR$ ls
fileA fileB
chunchi@chunchi-TravelMate-P243:~/hw10/DIR$ ls -t
fileB fileA
chunchi@chunchi-TravelMate-P243:~/hw10/DIR$ ls -l
total 373296
-rw-rw-r-- 1 chunchi chunchi 186480000 12月 31 04:27 fileA
-rw-rw-r-- 1 chunchi chunchi 195767106 12月 31 04:27 fileB
chunchi@chunchi-TravelMate-P243:~/hw10/DIR$
```

3.請使用 copy on write 的想法說明可以很快的建立快照

假設今天同時有很多個使用者讀取一個檔案，他們會獲得相同的檔案指標。如果假設有其中一位使用者修改內容時，系統會複製一份專用副本給其使用者，但其他使用者看到的資源仍不會改變。同時資料庫也會採取在它寫入時複製，給予使用者提供一份快照，因此可以很快的建立快照。

4.如果檔案系統是 ext4,請問還可以『一致性』備份嗎？

可以，但需要搭建在 LVM 之上，因為要建立一個新的快照，需要建立快照的 VG 跟 LV，而 LVM 的 snapshot 有 COW 特性，因此可以做一致性備份。而他的用途相對來說廣泛，也是唯一可以擴展新功能且具備多種功能的分區方法。