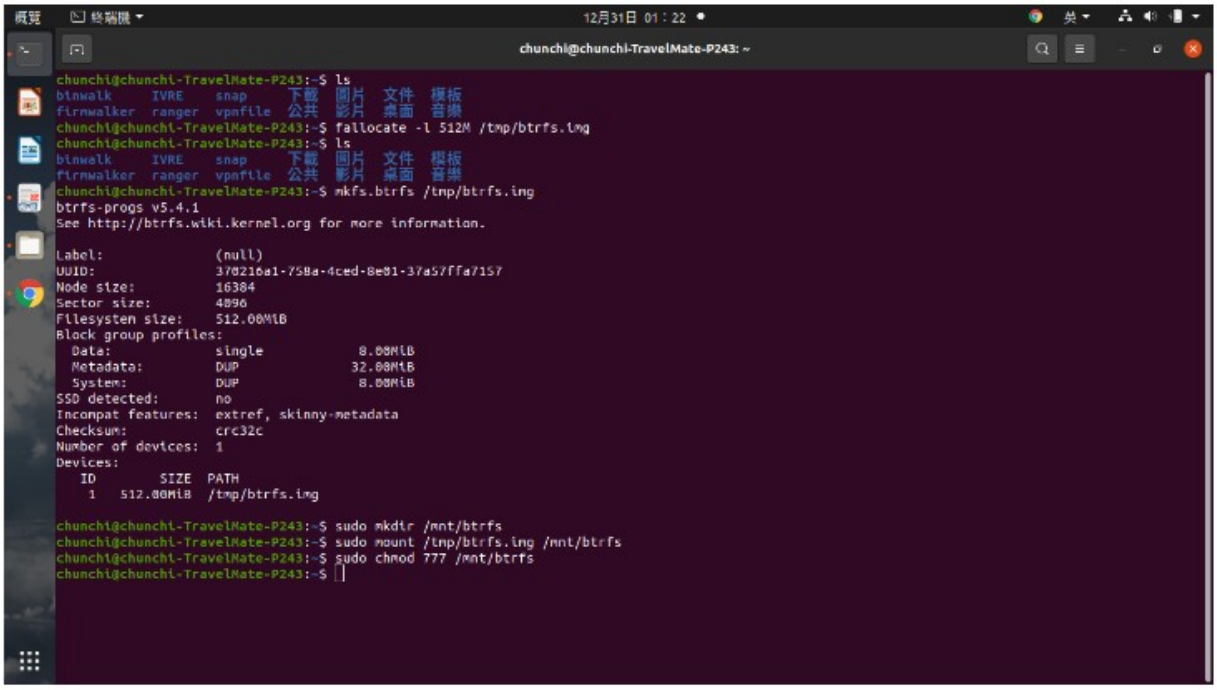
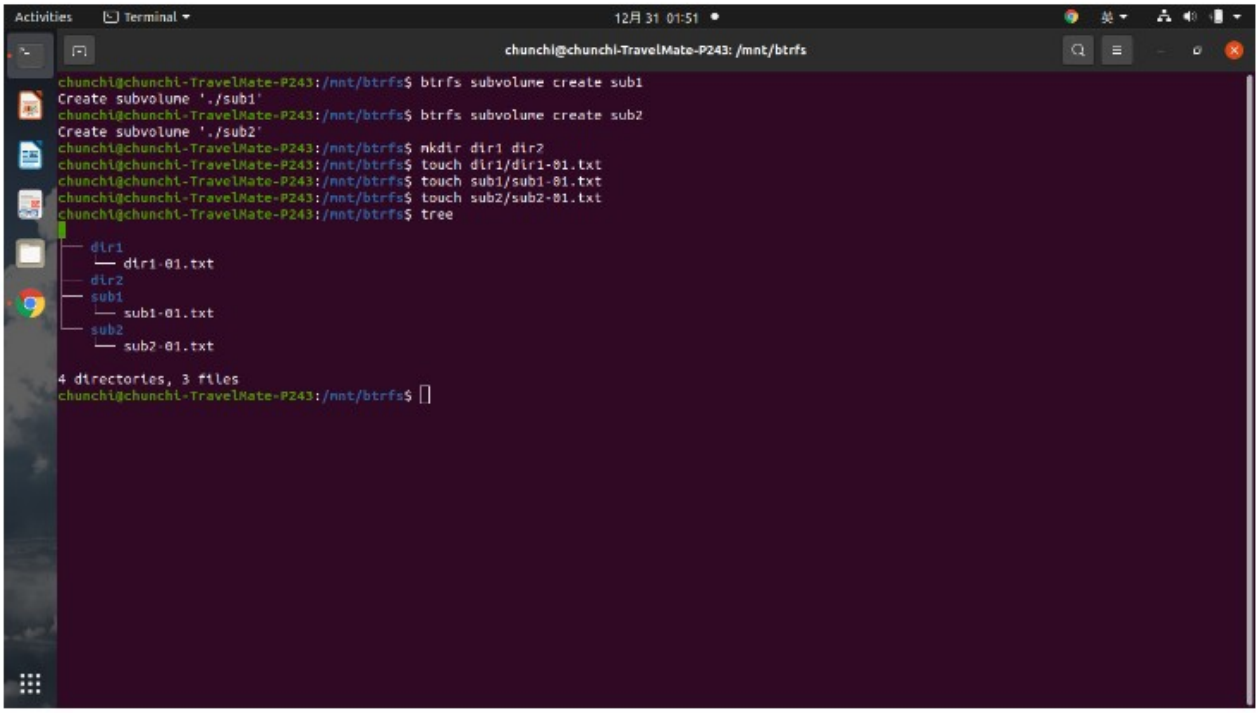
1.先建立一個 subvolume，當我們建立快照時可以發現存在兩個 subvolume。在建立的 snapshot 文件裡，可以發現我們看得到 dir1 裡的東西，卻無法看到 sub1 裡的東西，因為 sub1 為一個 subvolume。而我們對 subvolume 做快照時，並不會將 subvolume 裡的東西也做快照。而當我們對 sub1 進行快照時，可以看到 sub1 的東西出現在快照內。最終 snap-sub1 以及 sub1 底下個建立一個文件，都不會互相影響。

搭建環境



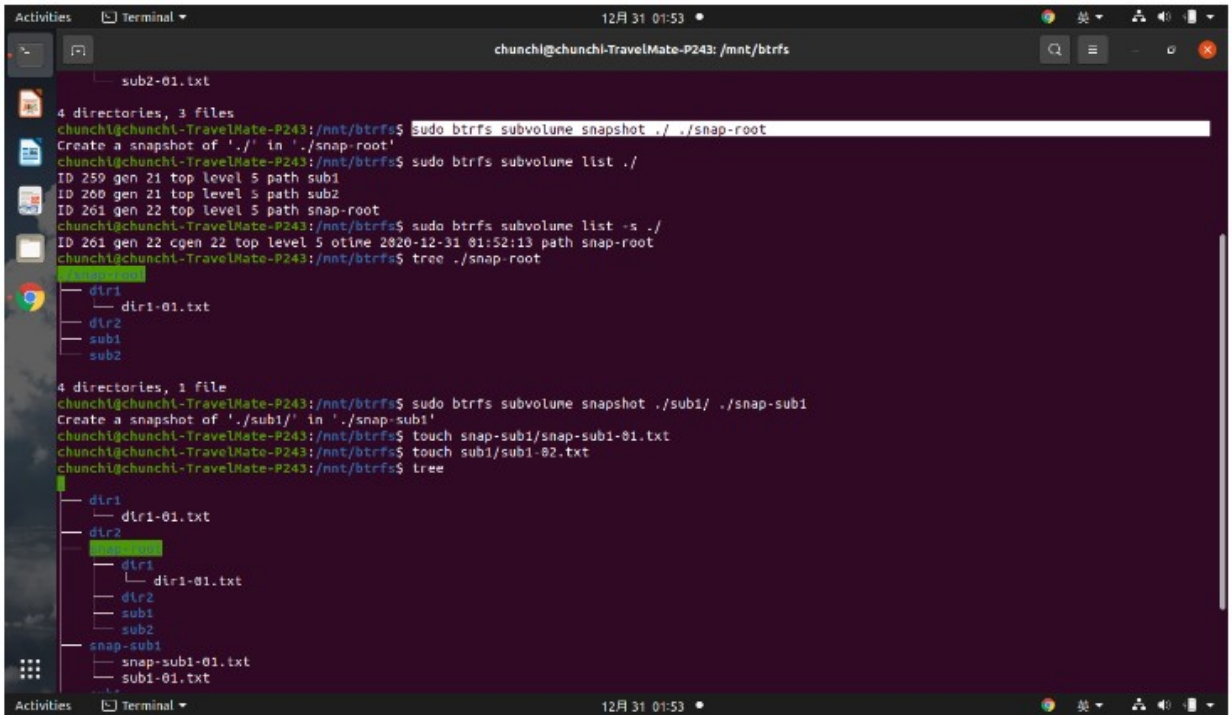

```
fallocate -l 512M /tmp/btrfs.img

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



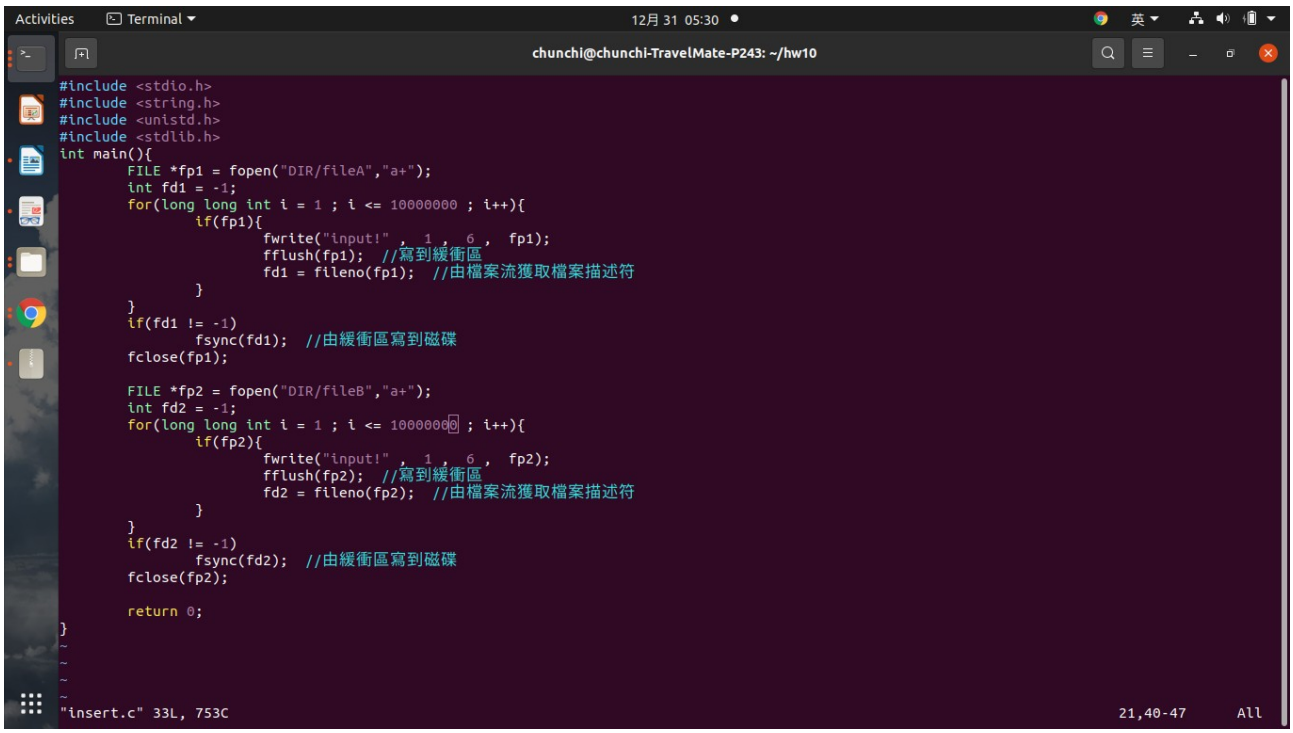

```
cd /mnt/btrfs

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
```

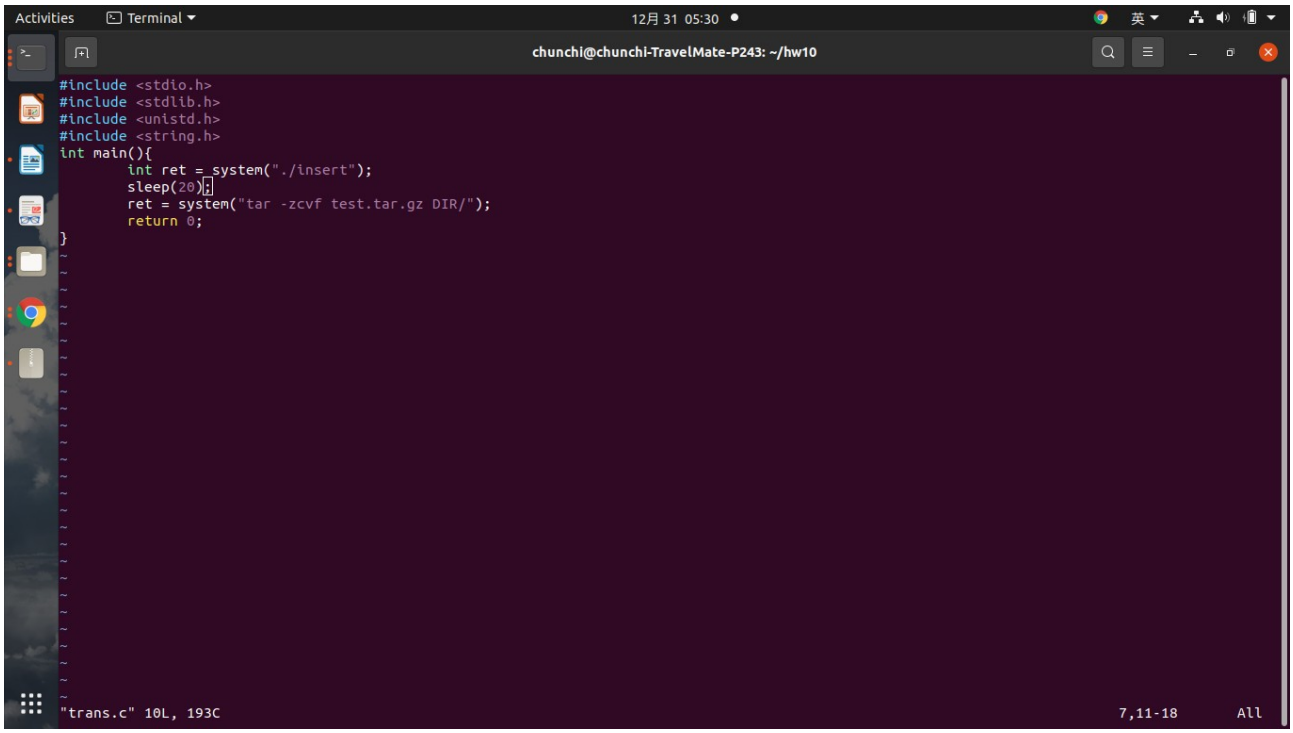
建立snapshot

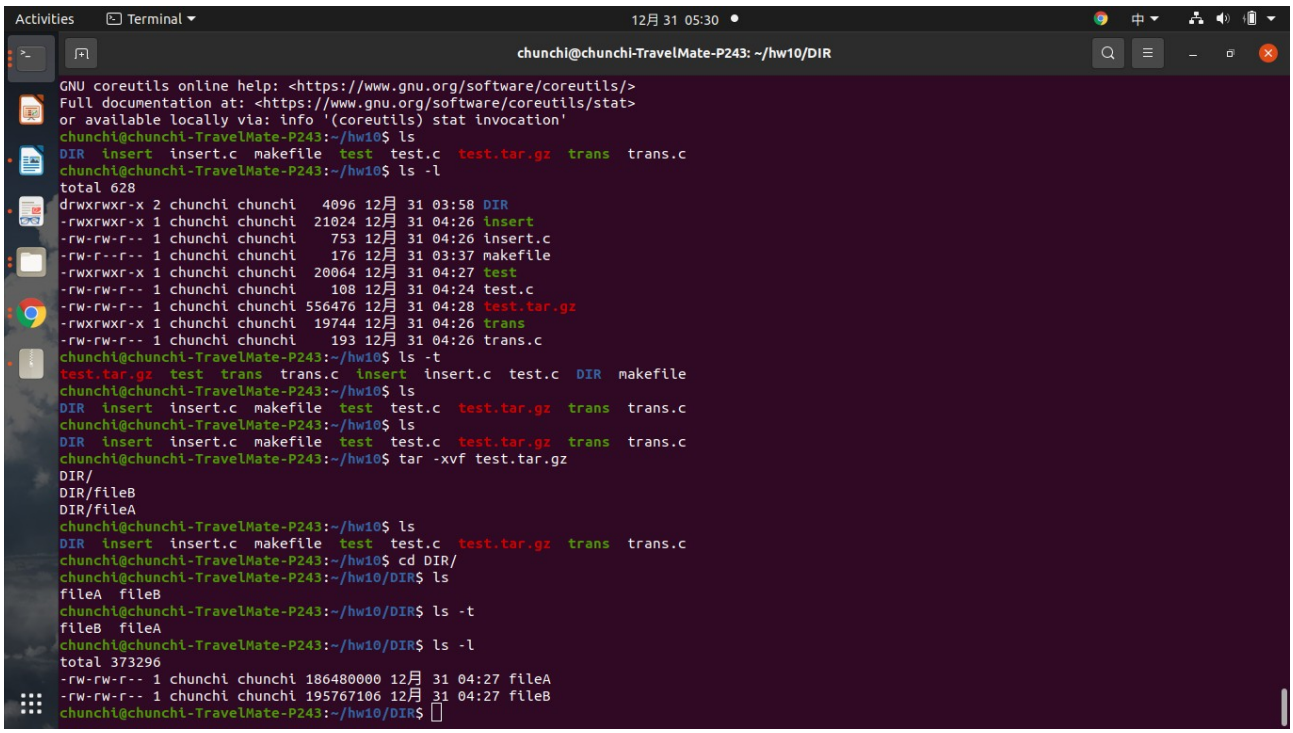



```
sudo btrfs subvolume snapshot ./ ./snap-root

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。


Activities
Terminal
12月 31 05:30
英
chunchi@chunchi-TravelMate-P243: ~/hw10
#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 <= 1000000 ; 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;
}
"insert.c" 33L, 753C
21,40-47
All


Activities
Terminal
12月 31 05:30
英
chunchi@chunchi-TravelMate-P243: ~/hw10
#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;
}
"trans.c" 10L, 193C
7,11-18
All



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,請問還可以做『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。