

[作業系統概論 HW12]

409410025 邱 O 恩

[1] 在 btrfs 檔案系統上，建立 subvolume

(a) use **fallocate** 來虛擬化硬碟，接著在 **tmp** 創建 **btrfs** 檔案系統

```
shiwulo@vm:~$ fallocate -l 512M /tmp/btrf.img
shiwulo@vm:~$ mkfs.btrfs /tmp/btrf.img
btrfs-progs v5.4.1
See http://btrfs.wiki.kernel.org for more information.

Label:                      (null)
UUID:                       c5aca00f-8f70-4dff-bdd9-0b3061885f52
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/btrf.img
```

[1]

(b) 把檔案掛到 `mnt/btrfs` 後，建立 `subvolume`，利用 `tree` 指令觀察

```
shiwulo@vm:~$ sudo mount /tmp/btrfs.img /mnt/btrfs/  
shiwulo@vm:~$ mount|grep btrfs
```

```
shiwulo@vm:/mnt/btrfs$ ls  
shiwulo@vm:/mnt/btrfs$ btrfs subvolume create sub0  
Create subvolume './sub0'  
ERROR: cannot create subvolume: Permission denied  
shiwulo@vm:/mnt/btrfs$ sudo btrfs subvolume create sub0  
Create subvolume './sub0'
```

```
shiwulo@vm:/mnt/btrfs$ tree  
.  
└── sub0
```

(c) 對 `subvolume` 做 `snapshot`

```
shiwulo@vm:/mnt/btrfs$ sudo btrfs subvolume snapshot ./ ./snapr  
Create a snapshot of './' in './snapr'  
shiwulo@vm:/mnt/btrfs$ sudo btrfs subvolume list ./  
ID 256 gen 9 top level 5 path sub0  
ID 257 gen 10 top level 5 path snapr  
shiwulo@vm:/mnt/btrfs$ ls  
snapr sub0
```

(d) 用 `tar` 對 `snapshot` 備份成 `backup.tar.bz2`

```
shiwulo@vm:/mnt/btrfs$ tar -jcvf ~/hw/oshw12/backup.tar.bz2 snapr  
snapr/  
snapr/sub0/  
shiwulo@vm:/mnt/btrfs$ cd ~  
shiwulo@vm:~$ cd ~/hw/oshw12  
shiwulo@vm:~/hw/oshw12$ ls  
backup.tar.bz2 makefile trans.c
```

[2] 透過 trans.c 驗證傳統的 tar 備份方法，產生檔案不一致的問題。

(a) 在 oshw12/ 中執行我的 trans.c，先對 testb.txt 寫入完在寫入 testa.txt

```
open( "~/hw/oshw12/sn/testb.txt" , O_RDWR | O_CREAT | O_APPEND , S_IRWXU );
open( "~/hw/oshw12/sn/testa.txt" , O_RDWR | O_CREAT | O_APPEND , S_IRWXU );

write( fdb , "test\0" , 4 ) ;
write( fda , "test\0" , 4 ) ;
```

(b) 執行 trans.c 去寫入 testb，接著寫入 testa，並且用 tar 指令打包備份。

```
shiwulo@vm:~/hw/oshw12$ ./transs
shiwulo@vm:~/hw/oshw12$ ./transs
shiwulo@vm:~/hw/oshw12$ ./transs
```

```
shiwulo@vm:~/hw$ tar -cvpzf result.tgz -c oshw12
```

(c) 發現檔案壓縮順序是 testa 後才 testb，和我對檔案寫入的順序不同。

```
oshw12/testa.txt
oshw12/testb.txt
oshw12/result.tgz
```

(d)

由上述方法驗證用傳統的 tar 方法進行壓縮，和檔案寫入的順序不一致。

先壓縮的檔案會比後壓縮的檔案更舊，此時 b 檔案比 a 檔案還新。

但是用 trans.c 跑得結果應該是 testa 要比較新，

因此得證了用 tar 備份會產生不一致的結果。

[3] 用 copy on write 的想法說明可以很快的建立快照

Copy on write 的概念在於，製造新檔案的時候，
會共享原本檔案的 data block。

只有在 write，也就是修改 data block 的時候才會進行 copy，也就是並不用
把原先所有的資料都複製下來，write 的時候才做真正的 copy，
這樣的做法可以節省很多時間。
因此可以很快地建立快照。

[4] 檔案系統是 ext4，請問還可以做『一致性』備份嗎？

可以，ext4 的檔案系統可以透過 LVM Snapshots 來完成『一致性』備份。