作業12：了解作業系統對備份的支援機制

學習目標（應用情境）：

* 許多重要的軟體都要有非常強健的容錯機制，例如：電腦忽然當機、停電等等。即使在這樣的情況下，系統可以暫停服務，但所有的服務不能發生錯誤。
* 交易系統就需要高強度容錯機制，例如：網路斷線、電腦當機、多個互斥交易（例如：同一張門票不能賣給多個人）
* 上述系統往往需要作業系統提供一些特別的功能，例如：強制資料寫入到硬碟（fsync）
* 以交易系統來說，必須底層作業系統保障「在任意時間，多個相關連的檔案」是一致的（consistency）
  + 做一次備份可能需要數十分鐘以上的時間。例如：第一個檔案『Ａ』於5:00備份成功，最後一次檔案『Ｚ』於7:00備份成功。那麼『A』與『Ｚ』在時間上就不具備一致性。
  + 如果『Z』是一個很大檔案共花了15分鐘才完成備份，那麼『Z』檔案中第一個block備份成功的時間與最後一個備份成功的block的時間相差了15分鐘。因此在『Z』檔案內部本身就不具備一致性。

題目：

1. 請在btrfs檔案系統上，建立subvolume
   * 對subvolume做snapshot，再對snapshot進行備份
2. 請設計一隻程式trans，這隻程式會先更新某檔案然後再更新另外一個檔案，請問你是否能透過這個程式驗證傳統的tar備份方法（不使用快照）會產生檔案不一致的問題。
   * 提示一：使用fsync
   * 提示二：注意檔名及inode編號，備份的順序可能與這二者有關
   * 提示三：可以觀察檔案時間，例如你的程式先更新檔案Ａ，再更新檔案Ｂ，如果在備份檔案中發現Ｂ比Ａ要來得新，那麼就是備份時出現時序上的錯誤。
3. 文件：
   * （40%）請附上截圖，說明你已經學會subvolume的快照功能，並且知道可以對快照進行備份。
   * （30%）請說明你的程式trans是怎樣設計的，是否可以用trans驗證使用tar備份可能發生「一致性問題」（無論結果是什麼。只要說明你的程式，和你所做的嘗試、邏輯即可） 。將關鍵程式碼寫在文件上。
   * （15%）請使用copy on write的想法說明可以很快的建立快照（提示：https://helmundwalter.de/en/blog/next-gen-backup-with-btrfs-snapshots-for-root-fs-and-databases/）
   * （15%）如果檔案系統是ext4，請問還可以做『一致性』備份嗎？（提示：https://linoxide.com/linux-how-to/lvm-snapshots-ext4-centos-7/）

作業繳交：

* 由於接近期末了，因此這次作業只需要繳交文件，文件必須是pdf格式。不需要繳交程式碼。
* 這次的作業可以非常的「難」，例如：說明btrfs是怎樣做快照的，快照有什麼樣的缺點，這些缺點又該如何解決。這次作業也可以很簡單，例如你的程式裡面分別執行system(“touch a”); system(“touch b”);』即可。

繳交：

1. 繳交期限：2020/12/31（星期四）早上8:00
2. 如果真的不會寫，記得去請教朋友。在你的報告上寫你請教了誰即可。

資源：

可使用下列主機，帳號為：s學號（如：s687410007），密碼為：ecourse上的email（如：shiwulo@gmail.com）

除了osdi1以外，其他的主機（osdi2~5）使用NFS連上osdi1，因此請在osdi1.cs.ccu.edu.tw上完成作業，或者使用你們自己的Linux完成作業。