

OS2022 MP4 Report

1. Briefly explain how you solve each problem. If your solutions are reasonable and correct, you can get the points. Note that you can get points even if you fail to solve the problems. (10 points)

(a) Problem 1: Large Files

這題需要實作的部分有 `bmap` 以及 `itrunc`，此外因為要增大檔案容量，我將 `inode` 裡面改放成10個Direct，1個Singly Indirect，2個Doubly Indirect，以下是 `bmap` 以及 `itrunc` 的實作細節：

- `bmap`: 我們需要先去檢查 `bn` 是對應到 `ip->addrs` 的哪一個entry，如果 `bn < 10`，那該block就是用Direct去指到，如果 `10 <= bn < 266` 就是對應到Singly Indirect，如果 `266 <= bn < 65802` 就是對應到第一個Doubly Indirect，如果 `65802 <= bn < 131338` 就是對應到第二個Doubly Indirect，否則 `bn` 就是invalid的。在把對應到的block讀進memory的時候，我們要先去計算為了得到該 `bn` 所對到的block，會需要哪些block去紀錄，而那些block如果還沒有被allocate也需要去allocate，最後找到實際對應到該 `bn` 的block時，也要去檢查他是否有被allocate，如果沒有也要去allocate，最後再將該block之 `addr` 回傳。
- `itrunc`: 我們要去go through這個inode裏所記錄的每一個block，也要去go through為了紀錄每一個block所用到的其他block，並且把他們都free掉。

這題另外需要注意的是，如果把某個block讀進memory，當不再使用時要以 `brelease` 去release它。

此外 `NDIRECT` 要設成10，另外define一個 `NDOUBLYINDIRECT` 設成 `(NINDIRECT * NINDIRECT)`，且 `struct dinode` 與 `struct inode` 的 `addrs` 大小都要設成 `NDIRECT+3`。

(b) Problem 2: Symbolic Links to Files

這題要實作的有 `sys_symlink` 和 `sys_open` 兩個function：

- `sys_symlink`: 我們首先先去用 `create` 建一個檔，在建立的過程中，`create` 會去檢查是否這個 `path` 已經有代表的其他檔案，或是有沒有其他不合法的操作；如果有，會回傳0，此時 `sys_symlink` 就會回傳-1代表有error；若沒有，則我們將 `target` 這個string的長度寫入inode，也將 `target` 這個string寫入inode，寫入長度的目的主要是為了讓之後open的時候，要去讀 `target` 這個string才知道要讀多少bytes。
- `sys_open`: 主要要將讀 `symlink` 的部分實作出來，在 `sys_open` 的時候會先去檢查一個file是不是symlink而且有沒有開 `O_NOFOLLOW`，如果是symlink且沒開 `O_NOFOLLOW`，那就會用一個while迴圈去跑，如果走到的檔案是symlink，那就會先把inode內存的 `target` 讀出來，並且將那個 `target` 所對到的inode讀進來並且更新 `ip`，只要 `ip->type == T_SYMLINK` 就會繼續跑while迴圈，此外也會記錄一個counter，只要迴圈走超過20層就會終止並回傳error，因為助教有說最多就那麼深。

(c) Problem 3: Symbolic Link to Directories

這題要實作的有 `sys_chdir` 和 `namex` 兩個function：

- `sys_chdir`: 這裡要補的是走到symlink如何處理。和上題 `sys_open` 概念有點像，一樣用一個while迴圈去檢查當前走到的檔案，如果走到的是symlink，就把其紀錄的 `target` 讀出來並且將 `target` 對應的inode讀入、更新 `ip`，只要 `ip->type == T_SYMLINK` 就會繼續往下更新 `ip`，同時一樣會有一個counter，如果迴圈走超過20層就會終止並回傳error。
- `namex`: 主要也是補處理symlink的部分。在 `while((path = skipelem(path, name)) != 0)` 這個迴圈裡，我會去檢查當前讀取到的dir會不會其實是symlink，如果是的話，我會更新 `path` 以及 `ip`，`path` 就替換成該symlink指到的target加上後面的子路徑，例如 `/a/b -> /b`，當我在讀 `/a/b/c` 的時候，走到 `/a/b` 時發現它是symlink，我就更新 `path` 成 `/b/c` 並且將 `ip` 重新指到根目錄，然後繼續跑while loop。我一樣會設一個counter來避免無窮迴圈，只要迴圈走超過20層就會終止並回傳error。

