系統程式設計 System Programming 2021

Hw02 使用mmap 進行檔案複製

電機四 406415089 宋 O 天

<前置作業>

先看一下 ./mmap_cp myHole myHole2 會發生甚麼事，

```
nash@SleepyCat:~/Desktop/system-programming/ch04_copy$ ./mmap_cp myHole myHole2
file size = 30000006
mmap  input: Success
inputPtr = 0x7f110f1a3000
mmap output: Success
outputPtr = 0x7f110d506000
memory copy
time(memcpy) = 13042869 nanosec
```

就和老師說的一樣，mmap 把檔案內的空洞都複製下來了。

```
nash@SleepyCat:~/Desktop/system-programming/ch04_copy$ ls -lhs myH*
12K -rw------- 1 nash nash 29M  三   21 16:35 myHole
29M -rw------- 1 nash nash 29M  三   21 16:27 myHole2
```

<Code>

這邊我刻意改了一下老師的 code，利用 clock_gettime()印出更精確的時間。因為用 time 函數發現好像只會印出固定的數字，查了一下 Stackoverflow 發現好像跟硬體有點關係，轉成更大的資料型態也沒用，所以我只好用 clock_gettime 來做。

```
time_begin:1616343796  time_end:1616343796
time_consume :0
nash@SleepyCat:~/Desktop/system-programming/ch04_copy$
```

輸入 ./mmap_cp2 ./myHole myHole3 產生 myHole3，我把 data_off 和 hole_off 印出來看

```
nash@SleepyCat:~/Desktop/system-programming/ch04_copy$ ./mmap_cp2 ./myHole myHole3
data off : 9998336
hole off : 10002432
data off : 19996672
hole off : 20000768
data off : 29999104
hole off : 30000005
data off : -1
hole off : -1
time_consume :141864179
```

我在 hole.c 裡每次都是都是利用 lseek 移動 10M 的大小，並寫入一個數字，如”1”(string)

大小一定是小於 4K，而 Unix 存取硬碟最小單位

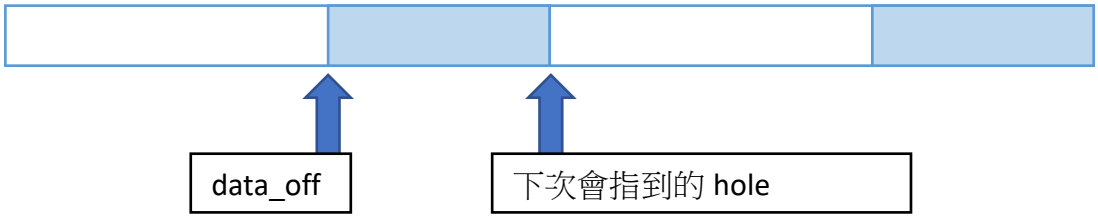
邏輯區塊是在 partition 進行 filesystem 的格式化時，所指定的『最小儲存單位』，這個最小儲存單位當然是架構在 sector 的大小上面(因為 sector 為硬碟的最小物理儲存單位啊！)，Block 的大小為 sector 2 的次方倍數。磁頭一次可以讀取一個 block，我們在格式化的時候，指定 Block 為 4 KBytes (亦即由連續的八個 sector 所構成一個 block)

10,000,000/4096 = 2441(用了 2441 個 blocks)

4096*2441 = 999836，999836+4096 = 10002432(明顯 Data 寫在第 2442 個 block)

然後發現即使一開始 hole 在檔案前方，data_off 和 hole_off 的順序仍然不需要討論，
因為我們先寫了 data_off = cur_off，之後才 hole_off = cur_off。data_off 在指到 data 時，
檔案指標就移動過了，自然會找到下一個 hole 的位置，開頭的 hole 就自動忽略了。

```
while(1){
    //case1 : data_off > hole_off
    cur_off = lseek(inputFd,cur_off,SEEK_DATA);
    data_off = cur_off;
    //printf("data off : %-ld\n",data_off);
    cur_off = lseek(inputFd,cur_off,SEEK_HOLE);
    hole_off = cur_off;
    //printf("hole off : %-ld\n",hole_off);
```



最後看一下 mmap_cp2 執行過後的 hole 檔案大小，確實在硬碟上只佔 12K

```
nash@SleepyCat:~/Desktop/system-programming/ch04_copy$ ./mmap_cp2 ./myHole myHole3
time_consume :80684
nash@SleepyCat:~/Desktop/system-programming/ch04_copy$ ls -lhs myHol*
12K -rw------- 1 nash nash 29M  三   21 21:53 myHole
29M -rw------- 1 nash nash 29M  三   21 16:27 myHole2
12K -rw------- 1 nash nash 29M  三   22 17:17 myHole3
```

另外，關於執行時間的部分，我是把它夾在，while 迴圈進入前和盡如後來測量，但我一直覺得這個方法會因為 programing style(例如 while 裡面有沒有 printf()、if 判斷是執行多久等等影響)，不是一個很好衡量用 mmap 的方法是否比一般 mycp 來的快的方法，想問老師怎麼看這問題，但通常直接用記憶體映射的方式會省去 read、write 系統呼叫的 overhead，會比較快一點。

參考資料

Linux 磁碟與硬體管理
http://linux.vbird.org/linux_basic/0230filesystem/0230filesystem.php
記憶體映射函數 mmap 的使用方法
https://welkinchen.pixnet.net/blog/post/41312211-%E8%A8%98%E6%86%B6%E9%AB%94%E6%98%A0%E5%B0%84%E5%87%BD%E6%95%B8-mmap-%E7%9A%84%E4%BD%BF%E7%94%A8%E6%96%B9%E6%B3%95

致謝

好友 楊 O 禕