OS HW4 108820018 蔡翔宇

Code 安裝&執行方式:

Q.11.13

以下皆在 ubuntu vm 中執行，需先安裝 vim

(1)

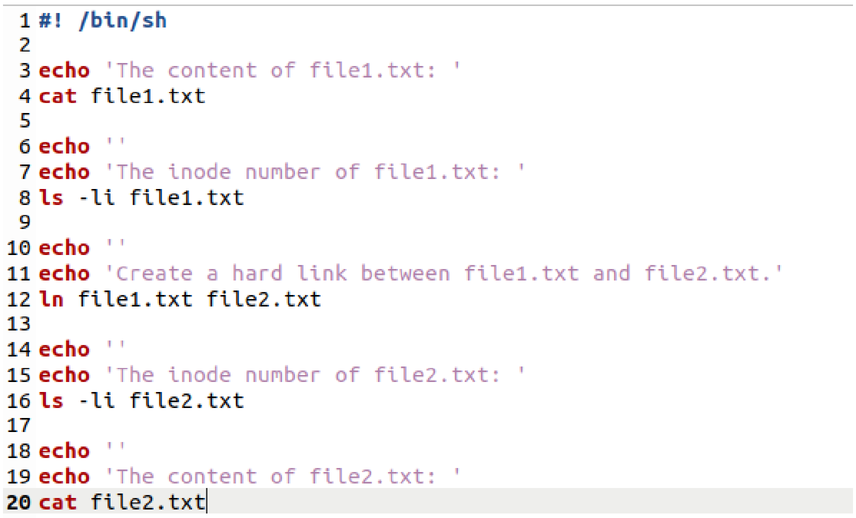
- What are the inode values of file1.txt and file2.txt? Are they the same or different?
  - Both 656679, they are the same.

- Do the two files have the same or different contents?
  - Yes, they have the same contents.

- Code 執行步驟

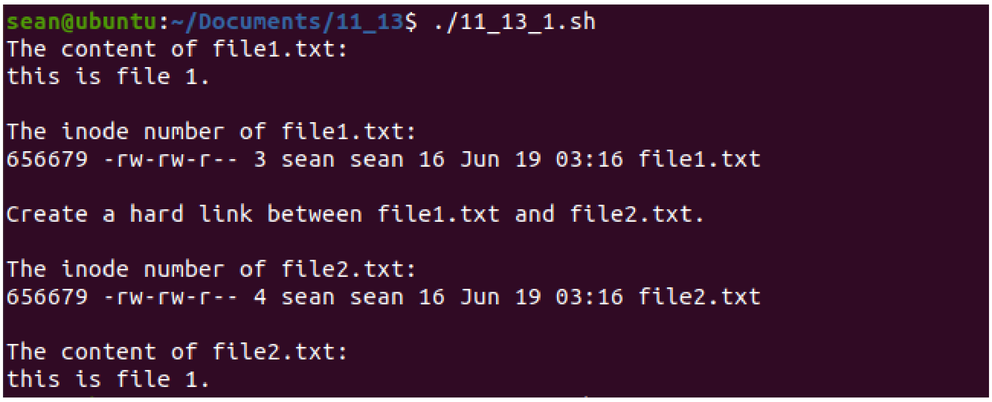
在 treminal 進入 question_11_13 資料夾後輸入./11_13_1.sh

11_13_1.sh 內容:



```
#! /bin/sh

echo 'The content of file1.txt: '
cat file1.txt

echo ''
echo 'The inode number of file1.txt: '
ls -li file1.txt

echo ''
echo 'Create a hard link between file1.txt and file2.txt.'
ln file1.txt file2.txt

echo ''
echo 'The inode number of file2.txt: '
ls -li file2.txt

echo ''
echo 'The content of file2.txt: '
cat file2.txt
```

執行結果:



```
sean@ubuntu:~/Documents/11_13$ ./11_13_1.sh
The content of file1.txt:
this is file 1.

The inode number of file1.txt:
656679 -rw-rw-r-- 3 sean sean 16 Jun 19 03:16 file1.txt

Create a hard link between file1.txt and file2.txt.

The inode number of file2.txt:
656679 -rw-rw-r-- 4 sean sean 16 Jun 19 03:16 file2.txt

The content of file2.txt:
this is file 1.
```

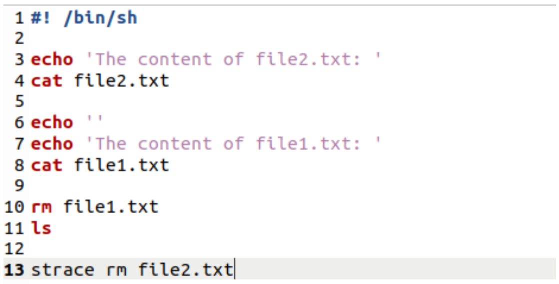
(2)

- Are the contents of file1.txt and file2.txt the same or different?
  - They are the same.
- Next, remove file1.txt. Does file2.txt still exist as well?
  - Yes.
- Check what system call is used for removing file2.txt by the following command: *strace rm file2.txt*.

- Code 執行步驟

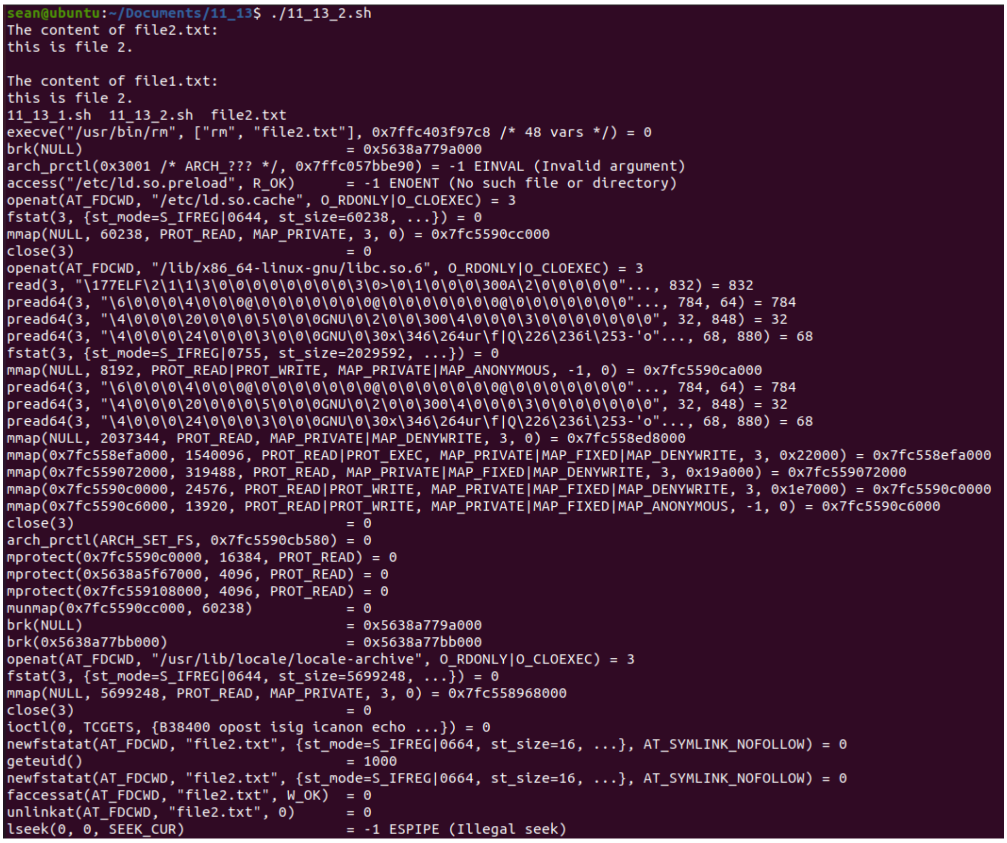
修改 file2.txt 內容後在 terminal 輸入./11_13_2.sh

11_13_2.sh 內容:

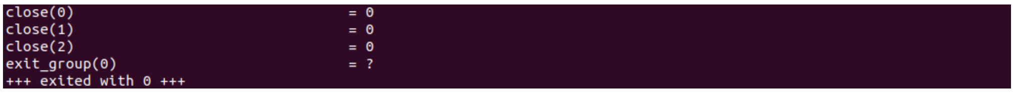


```
#! /bin/sh

echo 'The content of file2.txt: '
cat file2.txt

echo ''
echo 'The content of file1.txt: '
cat file1.txt

rm file1.txt
ls

strace rm file2.txt
```

11_13_2.sh 執行結果:



```
sean@ubuntu:~/Documents/11_13$ ./11_13_2.sh
The content of file2.txt:
this is file 2.

The content of file1.txt:
this is file 2.
11_13_1.sh  11_13_2.sh  file2.txt
execve("/usr/bin/rm", ["rm", "file2.txt"], 0x7ffc403f97c8 /* 48 vars */) = 0
brk(NULL)                               = 0x5638a779a000
arch_prctl(0x3001 /* ARCH_??? */, 0x7ffc057bbe90) = -1 EINVAL (Invalid argument)
access("/etc/ld.so.preload", R_OK)      = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
fstat(3, {st_mode=S_IFREG|0644, st_size=60238, ...}) = 0
mmap(NULL, 60238, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7fc5590cc000
close(3)                                = 0
openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\300A\2\0\0\0\0\0"..., 832) = 832
pread64(3, "\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0"..., 784, 64) = 784
pread64(3, "\4\0\0\0\20\0\0\0\5\0\0\0GNU\0\2\0\0\300\4\0\0\0\3\0\0\0\0\0\0\0", 32, 848) = 32
pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\30x\346\264ur\f|Q\226\236i\253-'o"..., 68, 880) = 68
fstat(3, {st_mode=S_IFREG|0755, st_size=2029592, ...}) = 0
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7fc5590ca000
pread64(3, "\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0"..., 784, 64) = 784
pread64(3, "\4\0\0\0\20\0\0\0\5\0\0\0GNU\0\2\0\0\300\4\0\0\0\3\0\0\0\0\0\0\0", 32, 848) = 32
pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\30x\346\264ur\f|Q\226\236i\253-'o"..., 68, 880) = 68
mmap(NULL, 2037344, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7fc558ed8000
mmap(0x7fc558efa000, 1540096, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x22000) = 0x7fc558efa000
mmap(0x7fc559072000, 319488, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x19a000) = 0x7fc559072000
mmap(0x7fc5590c0000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1e7000) = 0x7fc5590c0000
mmap(0x7fc5590c6000, 13920, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7fc5590c6000
close(3)                                = 0
arch_prctl(ARCH_SET_FS, 0x7fc5590cb580) = 0
mprotect(0x7fc5590c0000, 16384, PROT_READ) = 0
mprotect(0x5638a5f67000, 4096, PROT_READ) = 0
mprotect(0x7fc559108000, 4096, PROT_READ) = 0
munmap(0x7fc5590cc000, 60238)           = 0
brk(NULL)                               = 0x5638a779a000
brk(0x5638a77bb000)                     = 0x5638a77bb000
openat(AT_FDCWD, "/usr/lib/locale/locale-archive", O_RDONLY|O_CLOEXEC) = 3
fstat(3, {st_mode=S_IFREG|0644, st_size=5699248, ...}) = 0
mmap(NULL, 5699248, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7fc558968000
close(3)                                = 0
ioctl(0, TCGETS, {B38400 opost isig icanon echo ...}) = 0
newfstatat(AT_FDCWD, "file2.txt", {st_mode=S_IFREG|0664, st_size=16, ...}, AT_SYMLINK_NOFOLLOW) = 0
geteuid()                               = 1000
newfstatat(AT_FDCWD, "file2.txt", {st_mode=S_IFREG|0664, st_size=16, ...}, AT_SYMLINK_NOFOLLOW) = 0
faccessat(AT_FDCWD, "file2.txt", W_OK)  = 0
unlinkat(AT_FDCWD, "file2.txt", 0)      = 0
lseek(0, 0, SEEK_CUR)                   = -1 ESPIPE (Illegal seek)
```



```
close(0)                                = 0
close(1)                                = 0
close(2)                                = 0
exit_group(0)                           = ?
+++ exited with 0 +++
```

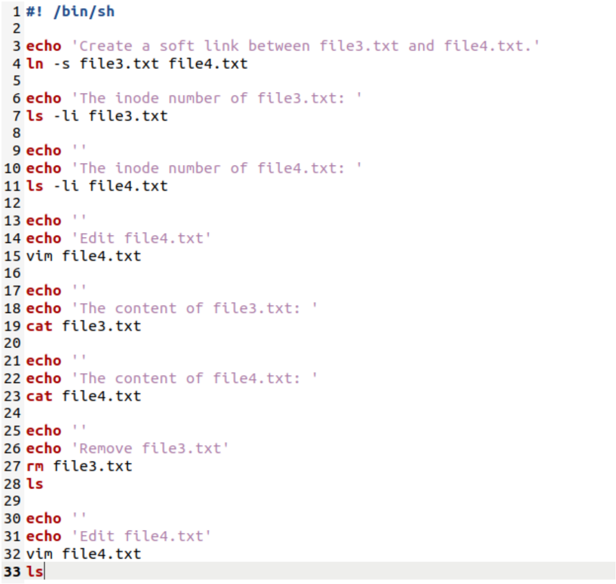
(3)

- Are the inode numbers of file3.txt and file4.txt the same, or is each unique?
  - They are each unique.
- Next, edit the contents of file4.txt. Have the contents of file3.txt been altered as well?
  - Yes.
- Last, delete file3.txt. Explain what happens when you attempt to edit file4.txt
  - Once file3.txt is deleted, the content of file4.txt become empty. Then, when attempting to edit file4.txt, the system creates a file3.txt which the content is the same as file4.txt.

- Code 執行步驟

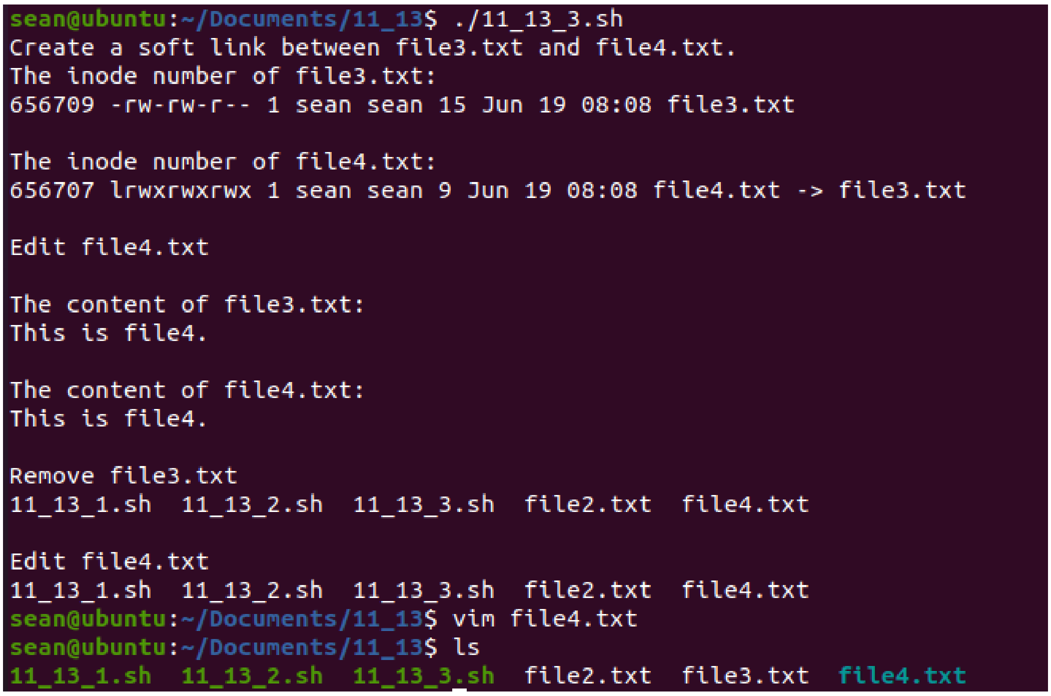
在 terminal 輸入./11_13_3.sh

11_13_3.sh 內容:



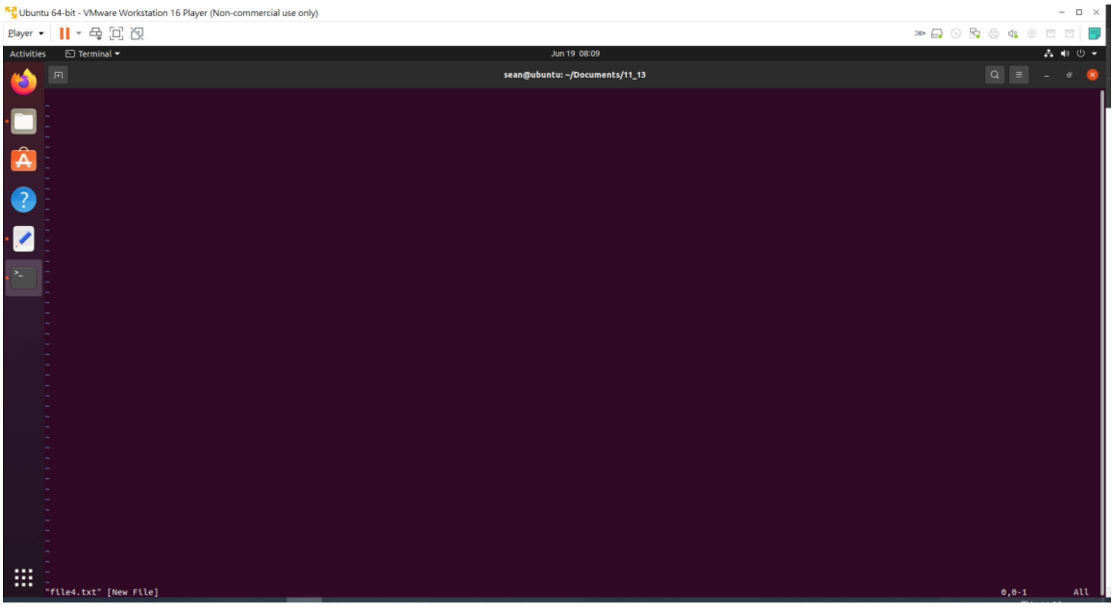
```
#! /bin/sh

echo 'Create a soft link between file3.txt and file4.txt.'
ln -s file3.txt file4.txt

echo 'The inode number of file3.txt: '
ls -li file3.txt

echo ''
echo 'The inode number of file4.txt: '
ls -li file4.txt

echo ''
echo 'Edit file4.txt'
vim file4.txt

echo ''
echo 'The content of file3.txt: '
cat file3.txt

echo ''
echo 'The content of file4.txt: '
cat file4.txt

echo ''
echo 'Remove file3.txt'
rm file3.txt
ls

echo ''
echo 'Edit file4.txt'
vim file4.txt
ls
```

11_13_3.sh 執行結果:



```
sean@ubuntu:~/Documents/11_13$ ./11_13_3.sh
Create a soft link between file3.txt and file4.txt.
The inode number of file3.txt:
656709 -rw-rw-r-- 1 sean sean 15 Jun 19 08:08 file3.txt

The inode number of file4.txt:
656707 lrwxrwxrwx 1 sean sean 9 Jun 19 08:08 file4.txt -> file3.txt

Edit file4.txt

The content of file3.txt:
This is file4.

The content of file4.txt:
This is file4.

Remove file3.txt
11_13_1.sh  11_13_2.sh  11_13_3.sh  file2.txt  file4.txt

Edit file4.txt
11_13_1.sh  11_13_2.sh  11_13_3.sh  file2.txt  file4.txt
sean@ubuntu:~/Documents/11_13$ vim file4.txt
sean@ubuntu:~/Documents/11_13$ ls
11_13_1.sh  11_13_2.sh  11_13_3.sh  file2.txt  file3.txt  file4.txt
```

移除 file3.txt 後嘗試編輯 file4.txt 時內容為空:



Ubuntu 64-bit - VMware Workstation 16 Player (Non-commercial use only)
Player
Activities
Terminal
Jun 19 08:09
sean@ubuntu: ~/Documents/11_13

```
"file4.txt" [New File]                                              0,0-1         All
```

Q.12.16

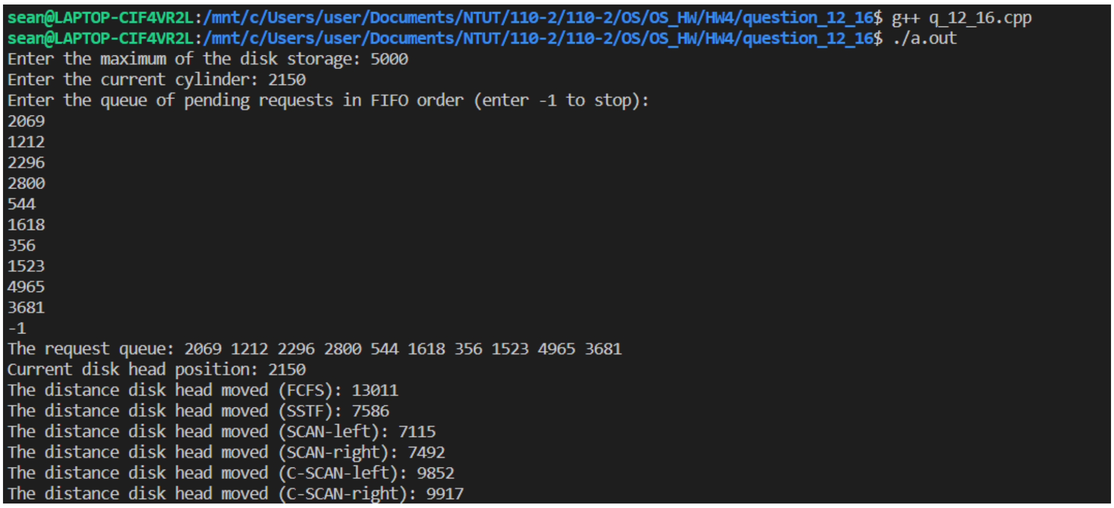
- 在 bash 進入 question_12_16 資料夾
- 輸入 g++ q_12_16.cpp
- 輸入./a.out

即可執行

注意

輸入每個 request 後皆須按一下 enter，欲結束輸入時輸入-1

程式會自行去除超出 max. disk storage 或 <0 之 request



```
sean@LAPTOP-CIF4VR2L:/mnt/c/Users/user/Documents/NTUT/110-2/110-2/OS/OS_HW/HW4/question_12_16$ g++ q_12_16.cpp
sean@LAPTOP-CIF4VR2L:/mnt/c/Users/user/Documents/NTUT/110-2/110-2/OS/OS_HW/HW4/question_12_16$ ./a.out
Enter the maximum of the disk storage: 5000
Enter the current cylinder: 2150
Enter the queue of pending requests in FIFO order (enter -1 to stop):
2069
1212
2296
2800
544
1618
356
1523
4965
3681
-1
The request queue: 2069 1212 2296 2800 544 1618 356 1523 4965 3681
Current disk head position: 2150
The distance disk head moved (FCFS): 13011
The distance disk head moved (SSTF): 7586
The distance disk head moved (SCAN-left): 7115
The distance disk head moved (SCAN-right): 7492
The distance disk head moved (C-SCAN-left): 9852
The distance disk head moved (C-SCAN-right): 9917
```