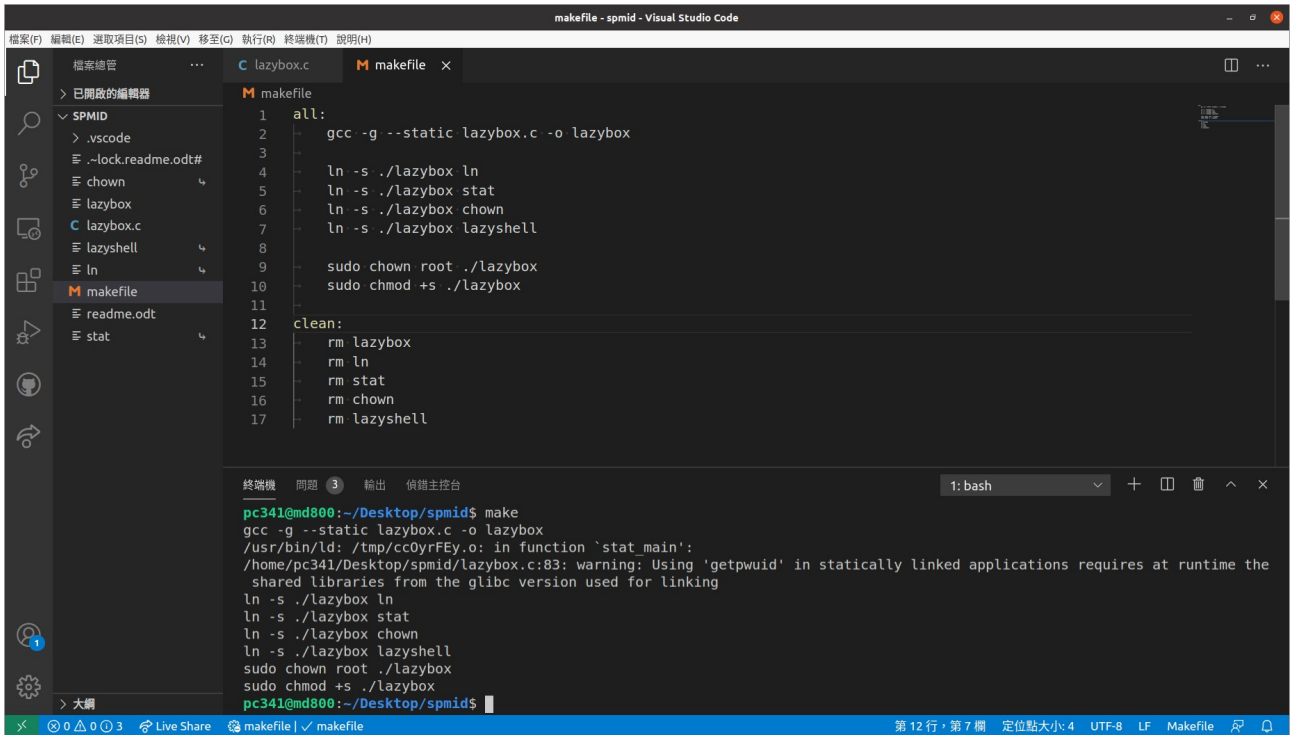


0. 執行 makefile 產生程式和 softlink，並同時更改檔案權限讓 chown 可以正常執行，要使用--static 靜態編譯，有報錯是因為過程中有使用把 uid 轉成 user name 所需的函式，而此函式屬於 glibc，因此會有錯誤。



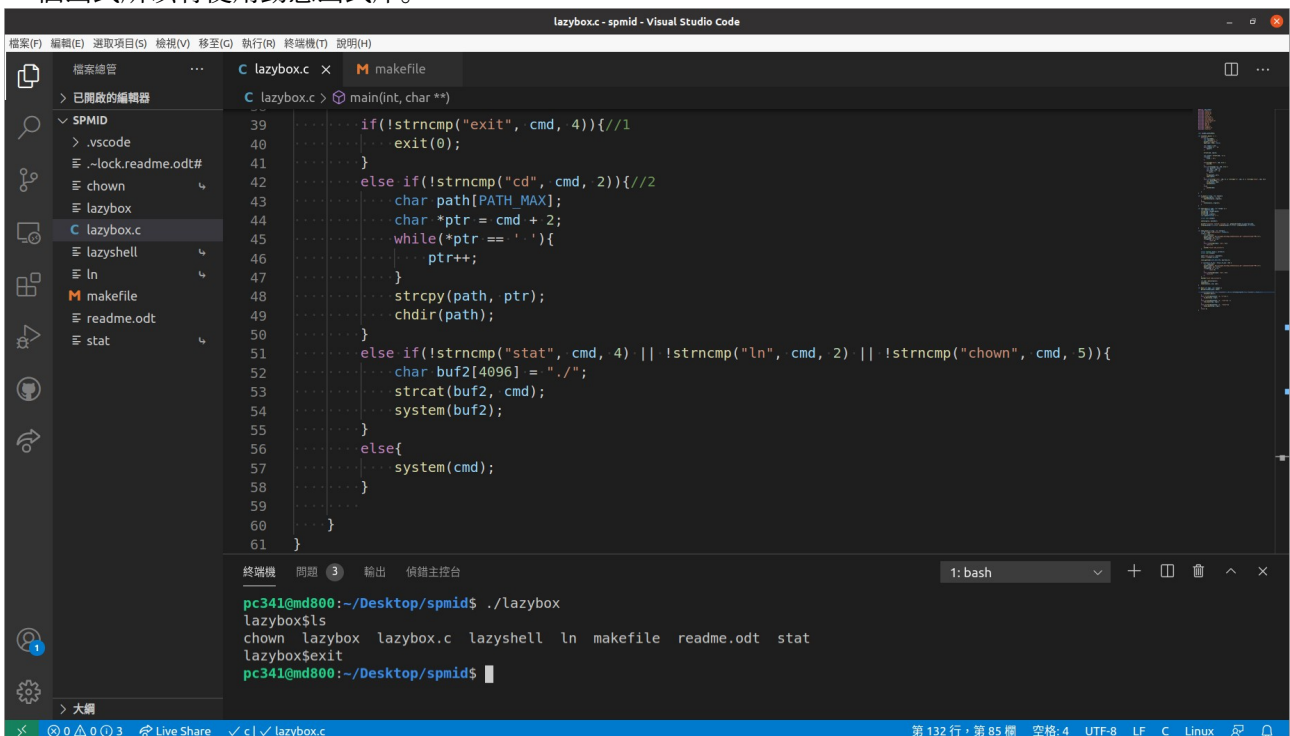
The screenshot shows the Visual Studio Code interface with a file explorer on the left and a terminal at the bottom. The file explorer shows a project named 'SPMID' with files like .vscode, .lock, readme.odt#, chown, lazybox, lazyshell, ln, makefile, readme.odt, and stat. The makefile content is as follows:

```
1 all:
2   gcc -g --static lazybox.c -o lazybox
3
4   ln -s ./lazybox ln
5   ln -s ./lazybox stat
6   ln -s ./lazybox chown
7   ln -s ./lazybox lazyshell
8
9   sudo chown root ./lazybox
10  sudo chmod +s ./lazybox
11
12 clean:
13   rm lazybox
14   rm ln
15   rm stat
16   rm chown
17   rm lazyshell
```

The terminal shows the execution of the makefile:

```
pc341@md800:~/Desktop/spmid$ make
gcc -g --static lazybox.c -o lazybox
/usr/bin/ld: /tmp/cc0YrFEy.o: in function `stat_main':
/home/pc341/Desktop/spmid/lazybox.c:83: warning: Using 'getpwuid' in statically linked applications requires at runtime the
shared libraries from the glibc version used for linking
ln -s ./lazybox ln
ln -s ./lazybox stat
ln -s ./lazybox chown
ln -s ./lazybox lazyshell
sudo chown root ./lazybox
sudo chmod +s ./lazybox
pc341@md800:~/Desktop/spmid$
```

1. 當執行 lazybox 時會先印出 lazybox 的提示符號 “lazybox\$”然後再執行命令，輸入 exit 時結束程式。使用靜態編一吋，若程式中有不合法的函式，編譯器會跳出錯誤訊息，如上圖，但因為必須要用到那個函式所以有使用動態函式庫。



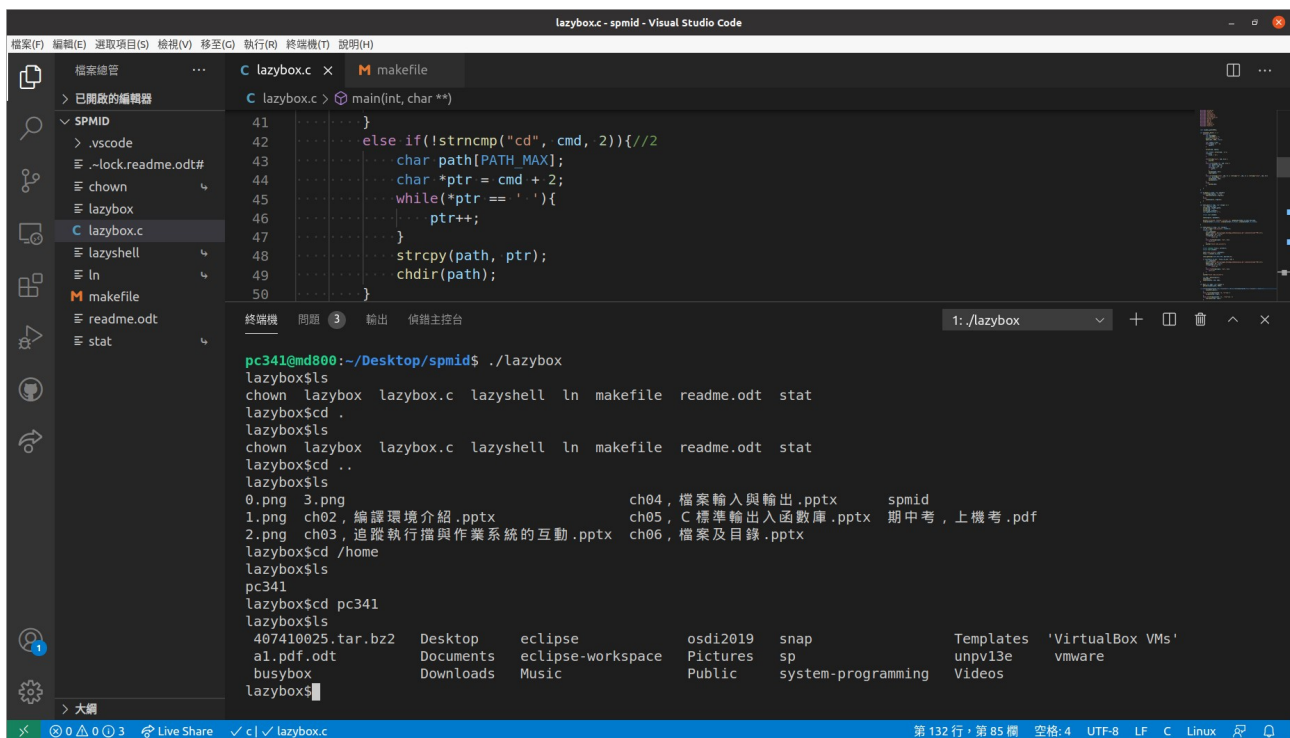
The screenshot shows the Visual Studio Code interface with a file explorer on the left and a terminal at the bottom. The file explorer shows a project named 'SPMID' with files like .vscode, .lock, readme.odt#, chown, lazybox, lazyshell, ln, makefile, readme.odt, and stat. The lazybox.c content is as follows:

```
39 if(!strcmp("exit", cmd, 4)){//1
40   exit(0);
41 }
42 else if(!strcmp("cd", cmd, 2)){//2
43   char path[PATH_MAX];
44   char *ptr = cmd + 2;
45   while(*ptr != ' '){
46     ptr++;
47   }
48   strcpy(path, ptr);
49   chdir(path);
50 }
51 else if(!strcmp("stat", cmd, 4) || !strcmp("ln", cmd, 2) || !strcmp("chown", cmd, 5)){
52   char buf2[4096] = ".";
53   strcat(buf2, cmd);
54   system(buf2);
55 }
56 else{
57   system(cmd);
58 }
59 }
60 }
61 }
```

The terminal shows the execution of the lazybox program:

```
pc341@md800:~/Desktop/spmid$ ./lazybox
lazybox$ls
chown lazybox lazybox.c lazyshell ln makefile readme.odt stat
lazybox$exit
pc341@md800:~/Desktop/spmid$
```

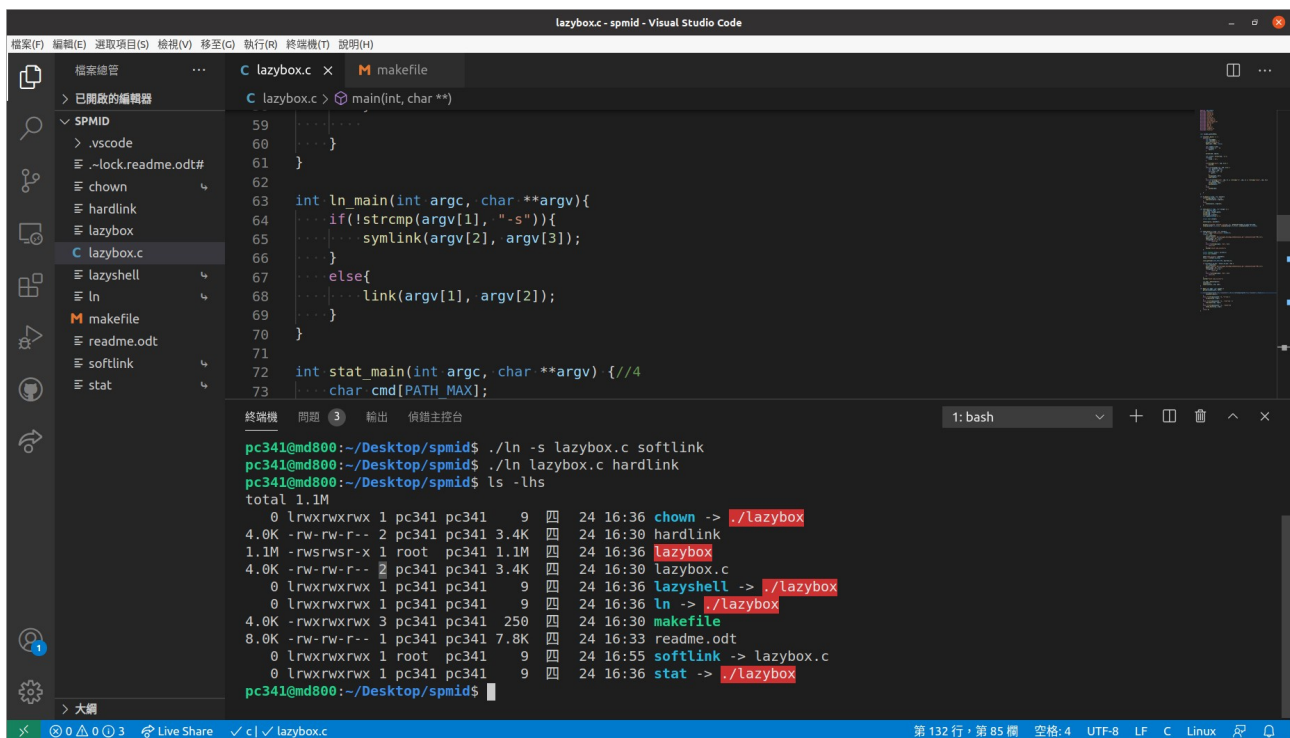
2. 內建 cd 在 lazyshell 裡面，使用 chdir 實做，同時使用 ls 驗證確實有更換路徑。  
分別測試 cd . (目前路徑)，cd .. (上一層)，cd /home (絕對路徑)，cd pc341 (相對路徑)



```
41     ...  
42     ...else if(!strcmp("cd", cmd, 2)){//2  
43     ...char path[PATH_MAX];  
44     ...char *ptr = cmd + 2;  
45     ...while(*ptr == ' '){  
46     ...ptr++;  
47     ...}  
48     ...strcpy(path, ptr);  
49     ...chdir(path);  
50     ...}
```

```
pc341@md800:~/Desktop/spmid$ ./lazybox  
lazybox$ls  
chown lazybox lazybox.c lazyshell ln makefile readme.odt stat  
lazybox$cd .  
lazybox$ls  
chown lazybox lazybox.c lazyshell ln makefile readme.odt stat  
lazybox$cd ..  
lazybox$ls  
0.png 3.png ch04, 檔案輸入與輸出.pptx spmid  
1.png ch02, 編譯環境介紹.pptx ch05, C 標準輸出函數庫.pptx 期中考, 上機考.pdf  
2.png ch03, 追蹤執行檔與作業系統的互動.pptx ch06, 檔案及目錄.pptx  
lazybox$cd /home  
lazybox$ls  
pc341  
lazybox$cd pc341  
lazybox$ls  
407410025.tar.bz2 Desktop eclipse osdi2019 snap Templates 'VirtualBox VMs'  
a1.pdf.odt Documents eclipse-workspace Pictures sp unpv13e vmware  
busybox Downloads Music Public system-programming Videos  
lazybox$
```

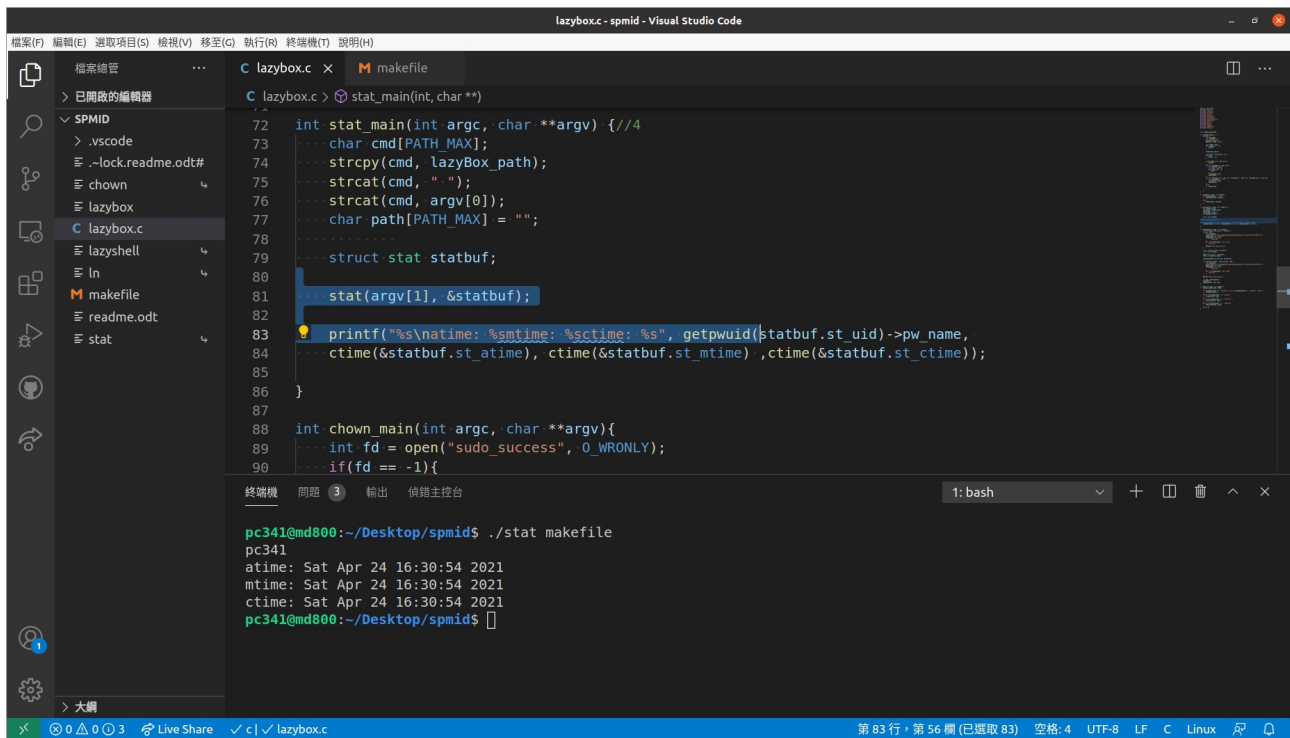
3. 自製 ln 指令，對 lazybox.c 建立 softlink 和 hardlink，測試是成功的，可以看到 softlink 連結到 lazybox.c，同時 hardlink 和 lazybox.c 的連結檔案為 2，所以也是成功的。



```
59     ...  
60     ...}  
61     ...}  
62  
63     int ln_main(int argc, char **argv){  
64     ...if(!strcmp(argv[1], "-s")){  
65     ...symlink(argv[2], argv[3]);  
66     ...}  
67     ...else{  
68     ...link(argv[1], argv[2]);  
69     ...}  
70     ...}  
71  
72     int stat_main(int argc, char **argv) {///  
73     ...char cmd[PATH_MAX];
```

```
pc341@md800:~/Desktop/spmid$ ./ln -s lazybox.c softlink  
pc341@md800:~/Desktop/spmid$ ./ln lazybox.c hardlink  
pc341@md800:~/Desktop/spmid$ ls -lhs  
total 1.1M  
0 lrwxrwxrwx 1 pc341 pc341 9 四 24 16:36 chown -> ./lazybox  
4.0K -rw-rw-r-- 2 pc341 pc341 3.4K 四 24 16:30 hardlink  
1.1M -rwsrwsr-x 1 root pc341 1.1M 四 24 16:36 lazybox  
4.0K -rw-rw-r-- 2 pc341 pc341 3.4K 四 24 16:30 lazybox.c  
0 lrwxrwxrwx 1 pc341 pc341 9 四 24 16:36 lazyshell -> ./lazybox  
0 lrwxrwxrwx 1 pc341 pc341 9 四 24 16:36 ln -> ./lazybox  
4.0K -rw-rw-rwx 3 pc341 pc341 250 四 24 16:30 makefile  
8.0K -rw-rw-r-- 1 pc341 pc341 7.8K 四 24 16:33 readme.odt  
0 lrwxrwxrwx 1 root pc341 9 四 24 16:55 softlink -> lazybox.c  
0 lrwxrwxrwx 1 pc341 pc341 9 四 24 16:36 stat -> ./lazybox  
pc341@md800:~/Desktop/spmid$
```

#### 4. 自製 stat，分別印出目標檔案的 username atime mtime ctime。

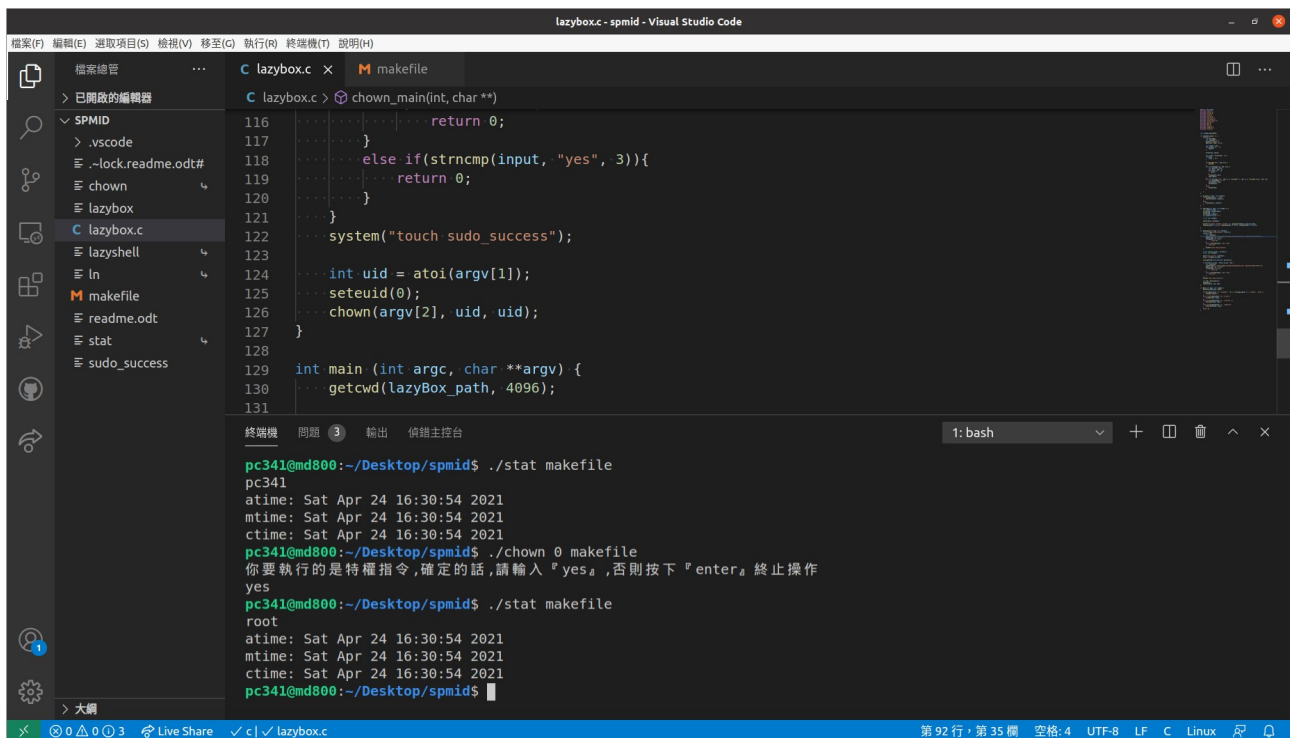


```
lazybox.c - spmid - Visual Studio Code
檔案(F) 編輯(E) 選取項目(S) 檢視(V) 移至(C) 執行(R) 終端機(T) 說明(H)

C lazybox.c x M makefile
C lazybox.c > stat_main(int, char **)
72 int stat_main(int argc, char **argv) { //4
73     char cmd[PATH_MAX];
74     strcpy(cmd, lazyBox_path);
75     strcat(cmd, ".");
76     strcat(cmd, argv[0]);
77     char path[PATH_MAX] = "";
78     struct stat statbuf;
79     stat(argv[1], &statbuf);
80
81     printf("%s\natime: %smtime: %sctime: %s", getpwuid(statbuf.st_uid)->pw_name,
82           ctime(&statbuf.st_atime), ctime(&statbuf.st_mtime), ctime(&statbuf.st_ctime));
83 }
84
85 int chown_main(int argc, char **argv){
86     int fd = open("sudo_success", O_WRONLY);
87     if(fd == -1){
88
89     }
90 }

終端機 問題 3 輸出 偵錯主控台 1: bash
pc341@md800:~/Desktop/spmid$ ./stat makefile
pc341
atime: Sat Apr 24 16:30:54 2021
mtime: Sat Apr 24 16:30:54 2021
ctime: Sat Apr 24 16:30:54 2021
pc341@md800:~/Desktop/spmid$
```

#### 5. 自製 chown，更改 makefile 的 owner，並使用 stat 驗證是否更改成功。



```
lazybox.c - spmid - Visual Studio Code
檔案(F) 編輯(E) 選取項目(S) 檢視(V) 移至(C) 執行(R) 終端機(T) 說明(H)

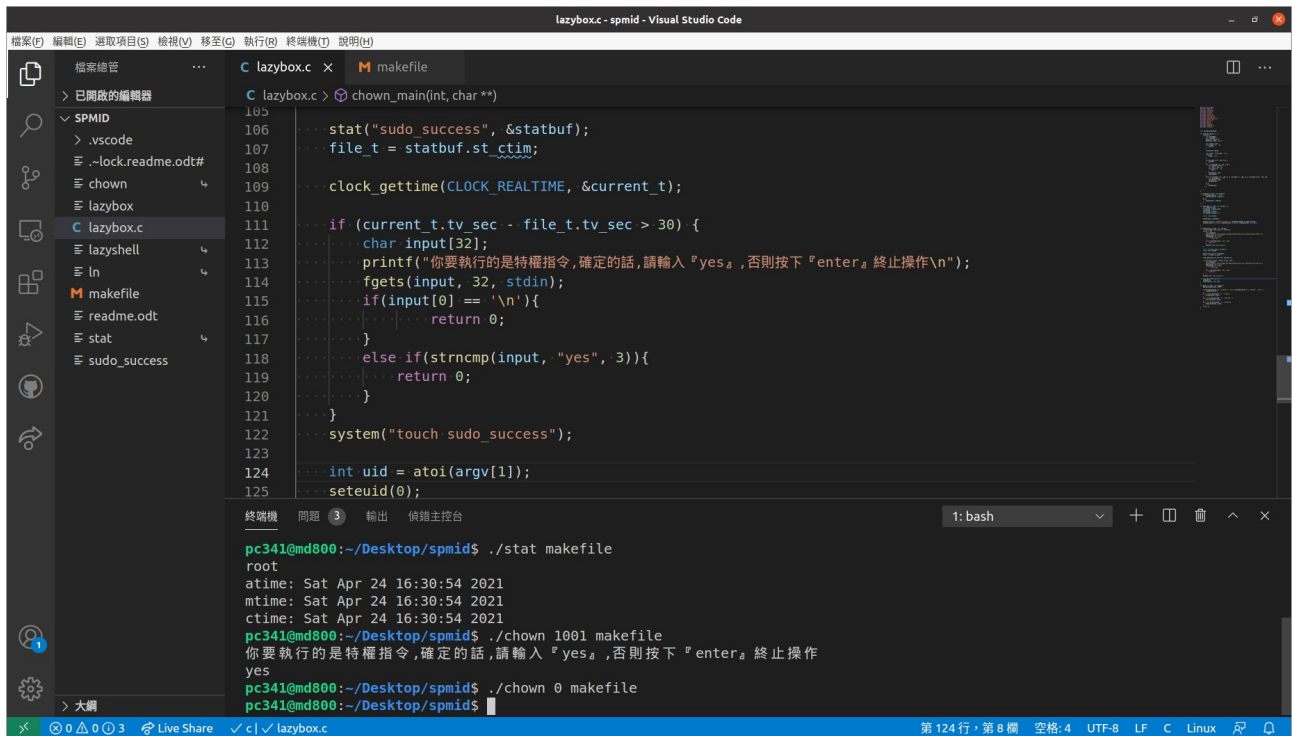
C lazybox.c x M makefile
C lazybox.c > chown_main(int, char **)
116     return 0;
117 }
118 else if(strcmp(input, "yes", 3)){
119     return 0;
120 }
121 }
122 system("touch sudo_success");
123
124 int uid = atoi(argv[1]);
125 seteuid(0);
126 chown(argv[2], uid, uid);
127 }
128
129 int main (int argc, char **argv) {
130     getcwd(lazyBox_path, 4096);
131 }

終端機 問題 3 輸出 偵錯主控台 1: bash
pc341@md800:~/Desktop/spmid$ ./stat makefile
pc341
atime: Sat Apr 24 16:30:54 2021
mtime: Sat Apr 24 16:30:54 2021
ctime: Sat Apr 24 16:30:54 2021
pc341@md800:~/Desktop/spmid$ ./chown 0 makefile
你要執行的是特權指令,確定的話,請輸入 'yes',否則按下 'enter' 終止操作
yes
pc341@md800:~/Desktop/spmid$ ./stat makefile
root
atime: Sat Apr 24 16:30:54 2021
mtime: Sat Apr 24 16:30:54 2021
ctime: Sat Apr 24 16:30:54 2021
pc341@md800:~/Desktop/spmid$
```

6. 第一張圖接續上面，因為已經過了 30 秒，所以必須再次輸入 yes，而 30 秒內再次將 owner 改成 root，因為還沒 30 秒所以不會詢問。

第二張圖，因為又過了 30 秒所以再次詢問，這時按下 enter 可以取消。

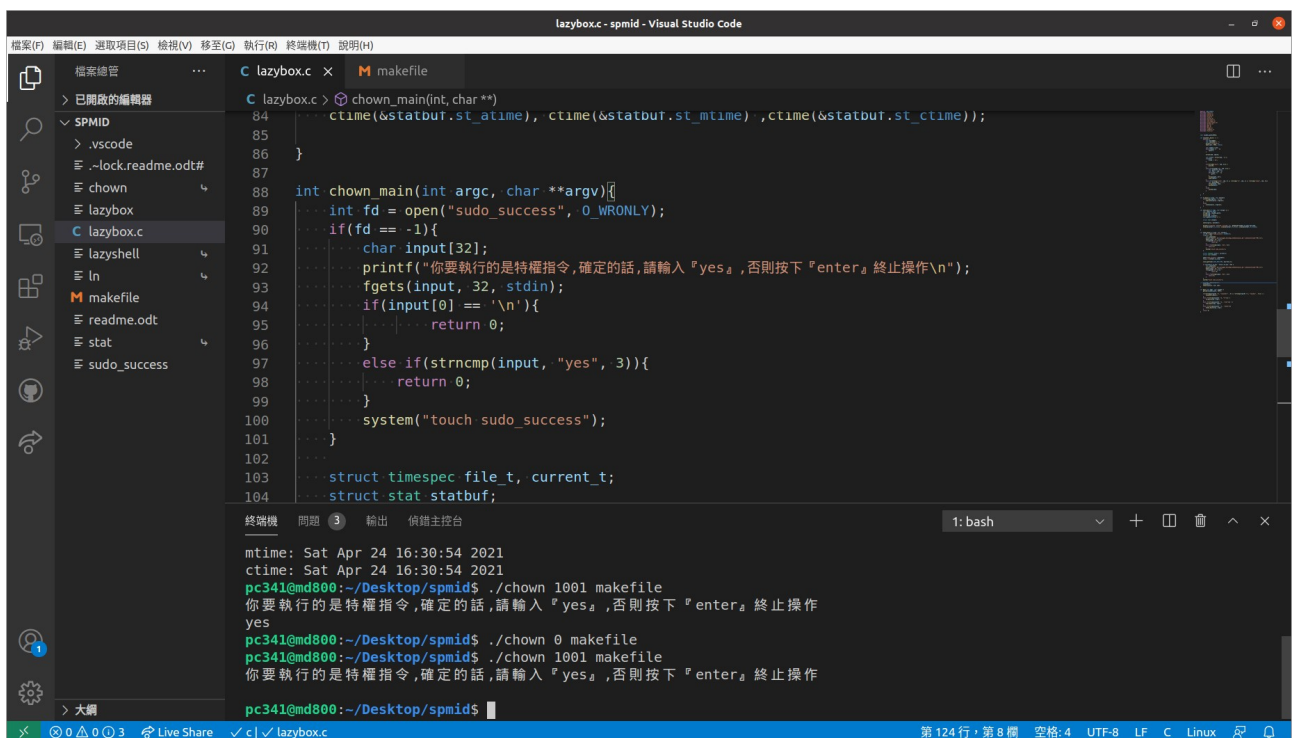
這邊有一點要注意的是因為一開始沒有 sudo\_success 這個檔案，直接建立的話會在 30 秒內，造成不用詢問就能進入的 bug，所以程式碼面有先判斷有沒有這個檔案，讓成是在各種情況下都能順利執行。



```
lazybox.c - spmid - Visual Studio Code
檔案總管 編輯(E) 選取項目(S) 檢視(V) 移至(G) 執行(R) 終端機(T) 說明(H)
> 已開啟的編輯器
SPMID
> .vscode
lock.readme.odt#
chown
lazybox
C lazybox.c
lazyspell
ln
M makefile
readme.odt
stat
sudo_success

C lazybox.c > chown_main(int, char **)
105
106 stat("sudo_success", &statbuf);
107 file_t = statbuf.st_ctim;
108
109 clock_gettime(CLOCK_REALTIME, &current_t);
110
111 if (current_t.tv_sec - file_t.tv_sec > 30) {
112     char input[32];
113     printf("你要執行的是特權指令，確定的話，請輸入 'yes'，否則按下 'enter' 終止操作\n");
114     fgets(input, 32, stdin);
115     if (input[0] == '\n') {
116         return 0;
117     }
118     else if (strcmp(input, "yes", 3)) {
119         return 0;
120     }
121 }
122 system("touch sudo_success");
123
124 int uid = atoi(argv[1]);
125 seteuid(0);

終端機 問題 3 輸出 偵錯主控台 1: bash
pc341@md800:~/Desktop/spmid$ ./stat makefile
root
atime: Sat Apr 24 16:30:54 2021
mtime: Sat Apr 24 16:30:54 2021
ctime: Sat Apr 24 16:30:54 2021
pc341@md800:~/Desktop/spmid$ ./chown 1001 makefile
你要執行的是特權指令，確定的話，請輸入 'yes'，否則按下 'enter' 終止操作
yes
pc341@md800:~/Desktop/spmid$ ./chown 0 makefile
pc341@md800:~/Desktop/spmid$
```



```
lazybox.c - spmid - Visual Studio Code
檔案總管 編輯(E) 選取項目(S) 檢視(V) 移至(G) 執行(R) 終端機(T) 說明(H)
> 已開啟的編輯器
SPMID
> .vscode
lock.readme.odt#
chown
lazybox
C lazybox.c
lazyspell
ln
M makefile
readme.odt
stat
sudo_success

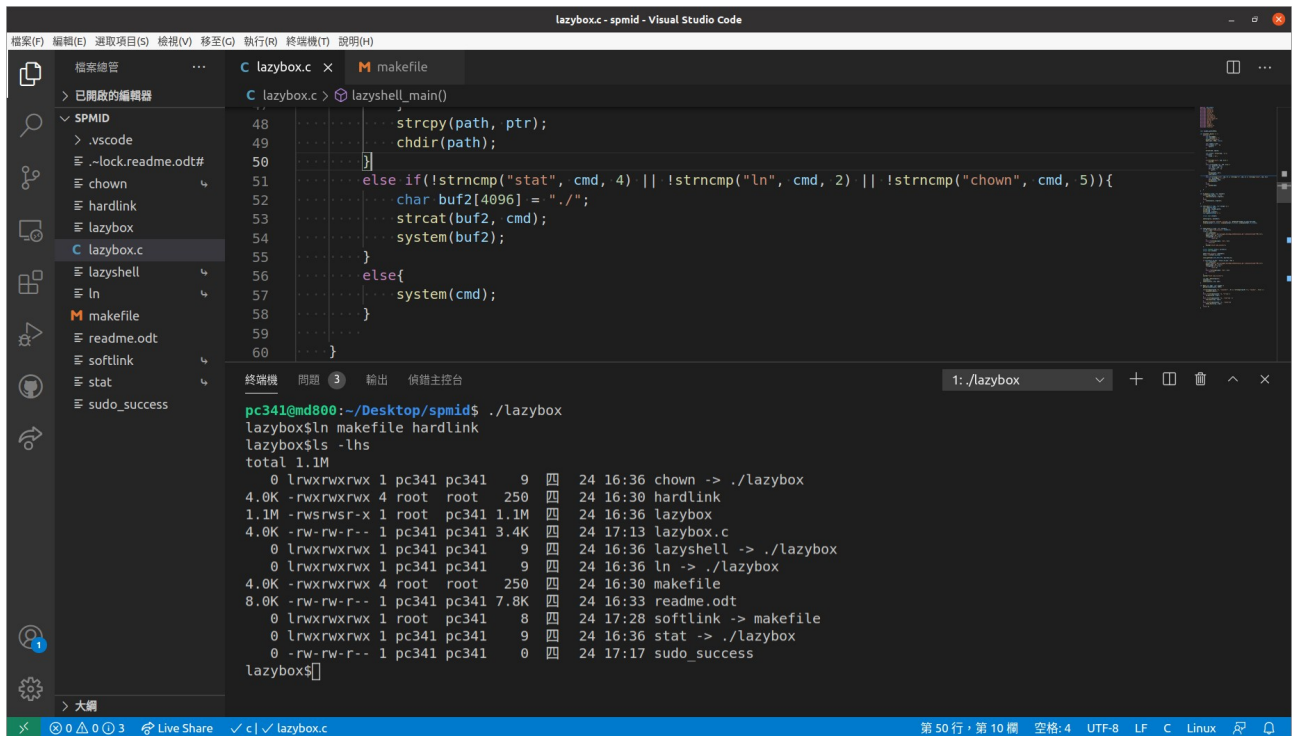
C lazybox.c > chown_main(int, char **)
84 ctime(&statbuf.st_atime), ctime(&statbuf.st_mtime), ctime(&statbuf.st_ctime));
85
86 }
87
88 int chown_main(int argc, char **argv) {
89     int fd = open("sudo_success", O_WRONLY);
90     if (fd == -1) {
91         char input[32];
92         printf("你要執行的是特權指令，確定的話，請輸入 'yes'，否則按下 'enter' 終止操作\n");
93         fgets(input, 32, stdin);
94         if (input[0] == '\n') {
95             return 0;
96         }
97         else if (strcmp(input, "yes", 3)) {
98             return 0;
99         }
100         system("touch sudo_success");
101     }
102
103     struct timespec file_t, current_t;
104     struct stat statbuf;

終端機 問題 3 輸出 偵錯主控台 1: bash
mtime: Sat Apr 24 16:30:54 2021
ctime: Sat Apr 24 16:30:54 2021
pc341@md800:~/Desktop/spmid$ ./chown 1001 makefile
你要執行的是特權指令，確定的話，請輸入 'yes'，否則按下 'enter' 終止操作
yes
pc341@md800:~/Desktop/spmid$ ./chown 0 makefile
pc341@md800:~/Desktop/spmid$ ./chown 1001 makefile
你要執行的是特權指令，確定的話，請輸入 'yes'，否則按下 'enter' 終止操作
pc341@md800:~/Desktop/spmid$
```



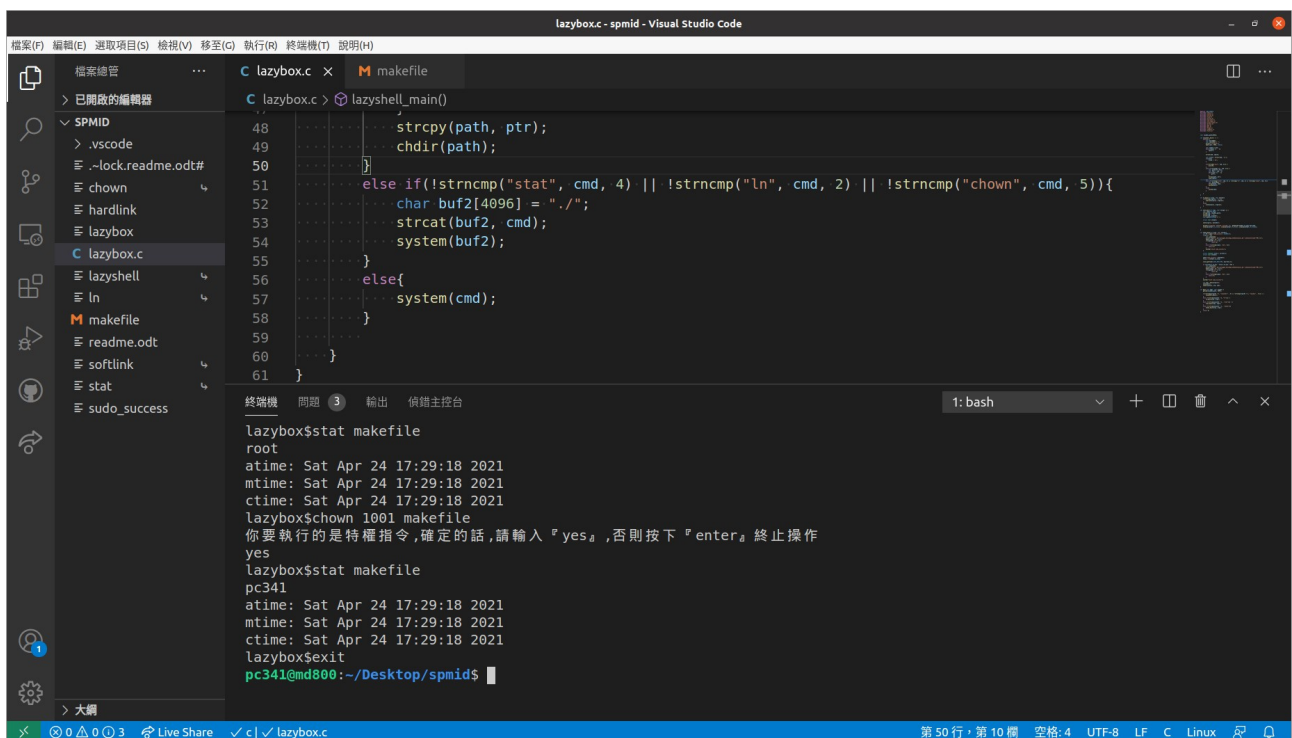
7. 在一開始 make 的時候就已建立 softlink，測試時也是使用 softlink 來進行測時，上面皆正常運作，所以也成功。

8. 在 lazyshell 模式下，優先執行同個資料夾下的 softlink 指令，ln 的部份在測試時可以加一行 printf 來測試是否使用 softlink ln，而 stat 和 chown 在第二張圖，格式很明顯是我們自製的 function，因此也成功。



```
lazybox.c - spmid - Visual Studio Code
檔案(F) 編輯(E) 選取項目(S) 檢視(V) 移至(G) 執行(R) 終端機(T) 說明(H)
C lazybox.c x M makefile
C lazybox.c > lazyshell_main()
48 strcpy(path, ptr);
49 chdir(path);
50 }
51 else if(!strcmp("stat", cmd, 4) || !strcmp("ln", cmd, 2) || !strcmp("chown", cmd, 5)){
52 char buf2[4096] = ".";
53 strcat(buf2, cmd);
54 system(buf2);
55 }
56 else{
57 system(cmd);
58 }
59 }
60 }
61 }

終端機 問題(3) 輸出 偵錯主控台
1: ./lazybox
pc341@md800:~/Desktop/spmid$ ./lazybox
lazybox$ln makefile hardlink
lazybox$ls -lhs
total 1.1M
0 lrwxrwxrwx 1 pc341 pc341 9 四 24 16:36 chown -> ./lazybox
4.0K -rwxrwxrwx 4 root root 250 四 24 16:30 hardlink
1.1M -rwsrwsr-x 1 root pc341 1.1M 四 24 16:36 lazybox
4.0K -rw-rw-r-- 1 pc341 pc341 3.4K 四 24 17:13 lazybox.c
0 lrwxrwxrwx 1 pc341 pc341 9 四 24 16:36 lazyshell -> ./lazybox
0 lrwxrwxrwx 1 pc341 pc341 9 四 24 16:36 ln -> ./lazybox
4.0K -rwxrwxrwx 4 root root 250 四 24 16:30 makefile
8.0K -rw-rw-r-- 1 pc341 pc341 7.8K 四 24 16:33 readme.odt
0 lrwxrwxrwx 1 root pc341 8 四 24 17:28 softlink -> makefile
0 lrwxrwxrwx 1 pc341 pc341 9 四 24 16:36 stat -> ./lazybox
0 -rw-rw-r-- 1 pc341 pc341 0 四 24 17:17 sudo_success
lazybox$
```



```
lazybox.c - spmid - Visual Studio Code
檔案(F) 編輯(E) 選取項目(S) 檢視(V) 移至(G) 執行(R) 終端機(T) 說明(H)
C lazybox.c x M makefile
C lazybox.c > lazyshell_main()
48 strcpy(path, ptr);
49 chdir(path);
50 }
51 else if(!strcmp("stat", cmd, 4) || !strcmp("ln", cmd, 2) || !strcmp("chown", cmd, 5)){
52 char buf2[4096] = ".";
53 strcat(buf2, cmd);
54 system(buf2);
55 }
56 else{
57 system(cmd);
58 }
59 }
60 }
61 }

終端機 問題(3) 輸出 偵錯主控台
1: bash
lazybox$stat makefile
root
atime: Sat Apr 24 17:29:18 2021
mtime: Sat Apr 24 17:29:18 2021
ctime: Sat Apr 24 17:29:18 2021
lazybox$chown 1001 makefile
你要執行的是特權指令,確定的話,請輸入 'yes',否則按下 'enter' 終止操作
yes
lazybox$stat makefile
pc341
atime: Sat Apr 24 17:29:18 2021
mtime: Sat Apr 24 17:29:18 2021
ctime: Sat Apr 24 17:29:18 2021
lazybox$exit
pc341@md800:~/Desktop/spmid$
```

1~8 項皆完成！