

考試目標：

- 基於作業中的busybox的概念實現自己的版本的『lazybox』
- 瞭解有些指令一定要在shell中實作

題目描述：

1. (20pt) (目的：了解靜態編譯、exit必須在lazybox內實現) 撰寫一支程式，名為lazybox，可以不斷的讀取使用者的命令，然後使用system()的方式呼叫外部的應用程式，請注意，要先印出lazybox的提示符號“lazybox\$”然後再執行命令，exit必須另外處理。Lazybox必須使用靜態編譯（--static），在報告中說明你如何驗證你的程式碼沒有使用動態函數庫

```
#include <stdio.h>
int main() {
    while(1) {
        char command[4096];
        printf("lazybox$");
        fgets(command, 4096, stdin);
        if (command == "exit")
            exit(0);
        else {
            system(command);
        }
    }
}
```

2. (20pt) (目的：了解cd無法使用「外部的執行檔實作」，一定要內建於shell) 讓lazybox支援cd，也就是更換目前的工作目錄。提示：chdir()，必須處理

- .
- ..
- 相對路徑
- 絕對路徑

3. (10pt) (目的：將一個外部指令實現於lazybox中) 讓lazybox支援soft link、hard link（提示：第六章投影片的第54頁），支援的指令如下

- ln -s /home/shiwulo/cat sym_ln //第一個路徑一定是絕對路徑，softlink建立在目前的工作目錄，sym_ln是softlink的名稱
- ln -s /home/shiwulo/cat hard_ln //第一個路徑一定是絕對路徑，hardlink建立在目前的工作目錄，hard_ln是hard link的名稱

4. (10pt) (目的：了解時間格式的轉換、了解如何取得使用者的名稱) 讓lazybox支援stat指令（提示：第六章投影片的第82頁），支援的指令如下

- stat file，依序顯示fileowner、atime、mtime、ctime。時間的顯示方式在第六章投影片的第86頁，拿到fileowner的ascii的方式在第六章的88頁。file一定是在目前的工作目錄底下

5. (10pt) (目的：了解sudo的用法) 支援chown，可以改變目前工作目錄的某一個檔案的owner，這個指令是特權指令，因此必須使用setuid設定lazybox的權限。

輸入的形式如下

chown 1000 file //檔案一定為在目前的工作目錄下，使用者名稱必然是一個合法的userid

chown 0 file

6. (10pt) (目的：了解sudo後一段時間內不需要再次輸入密碼的實作方法) 延續上一題，執行chowner的時候必須『提醒訊息』如下：

你要執行的是特權指令，確定的話，請輸入『確定』二個字，否則按下『enter』終止操作

如果使用者輸入確定那麼就讓使用者改變檔案的擁有者，否則取消。當使用者輸入確定時，接下來的30秒內使用者有相同的操作的話，系統會直接允許chown這個特權指令。提示：建立一個檔案（例如：sudo_success，可以使用system(touch sudo_success))，檔案的owner是root，透過ctime得知上次使用者執行這個指令的時間。如果時間間隔小於一分鐘，就直接執行特權指令

7. (10pt) (目的：了解busybox的實作方式) 完成lazybox。建立softlink連到lazybox，包含：ln、chown、stat及lazysHELL，如果執行lazysHELL，那麼其行為如同第一小題，如果執行ln、chown、stat等，就不會有命令列，而是直接執行第3、4、5、6小題的功能。

8. (10pt) (目的：了解busybox的實作方式) 讓lazybox更像是busybox，當使用者輸入ln、chown、stat等，那麼優先執行與lazybox擺在同一個工作目錄的softlink

- lazybox一執行時，立即拿到目前的工作目錄（第六章，第66頁）
- 將命令接上工作目錄成為完整路徑X，然後system(x)

繳交：

- makefile及編譯所需要的source code
 - 你可以在makefile內加入sudo，以設定正確的檔案權限
 - 一定要確認你的程式都是用靜態編譯完成
 - 原則上只需要產生lazybox和相對應的softlink。如果你將每個功能實現成一個執行檔，請在pdf檔案內說明
- 一個readme.pdf檔案，說明你完成了哪些，或者哪些是部分完成，讓助教可以依照你的方式測試執行結果。特別注意，完成的項目及完成的程度一定要具體寫好，助教必須依照大家寫出來的功能測試、給分
- 在readme.pdf內不需要撰寫特別的報告。
- 所有的檔案壓縮成.tar

tar jcvf 你的學號.tar.bz2 DirName

總體提示：

```
#include <stdio.h>

char lazyBox_path[4096];
int lazysHELL_main() {
    while(1) {
        char command[4096];
        printf("lazybox$");
        fgets(command, 4096, stdin);
        if (command == "exit")
            exit(0);
        else if (command == "cd") {

        }
        else {
            system(command);
        }
    }
}

int ln_main() {

}

int stat_main() {

}

int chown_main() {
    struct timespec file_t, current_t;
    struct stat;
    stat("sudo_success", &stat);
    file_t = stat.st_ctimespec;
    clock_gettime(CLOCK_REALTIME, &current_t);
    if (current_t - file_t > 30sec) {
        if (prompt()==false) return;
    }
    system("touch sudo_success");
    /*...*/
}

int main(int arg, char **argv)
{
    getcwd(&lazyBox_path, 4096);
    if (arg[0] == "lazybox") {
        lazysHELL_main();
    }
    else if (argv[0] == "something")
    {
        something();
    }
}
```