**系統程式期中考**

**（上機考部分）**

|  |  |  |
| --- | --- | --- |
| **姓名：** | **學號：** | **成績：** |

Ron博士的實驗室新購一台檔案伺服器，這台檔案伺服器使用mount指令掛載在/home/labBook目錄底下。

Ron博士希望每個研究生，研究進行到一個進度的時候，可以利用labBook程式將實驗相關的數據複製到/home/labBook底下，為了保障資料的安全性，因此只有Ron可以修改、刪除/home/labBook目錄下的任何東西（包含目錄及檔案），labBook的使用方式如下：

使用者必須在自己的家目錄執行labBook，也就是~/

1. 紀錄檔案

假設研究生的名稱是shiwulo，那麼shiwulo只能在/home/shiwulo底下執行

labBook filename

效果：labBook會將filename複製到在/home/labBook/shiwulo/yyyy-mm-dd-hh:mm:ss/filename，其中yyyymmdd-hh:mm:ss是目錄名稱

1. 紀錄目錄

假設研究生的名稱是shiwulo，那麼shiwulo只能在/home/shiwulo底下執行

labBook Dir-name

效果：labBook會將/home/shiwulo/Dir-name中所有的目錄及檔案複製到/home/backup/shiwulo/yyyy-mm-dd-hh:mm:ss/Dir-name底下

1. 研究生可以檢視他所上傳的所有檔案及目錄

假設該名研究生是shiwulo

那麼shiwulo可以觀看/home/labBook/shiwulo下所有的目錄及檔案的名稱，但shiwulo對/home/labBook/shiwulo下所有檔案、資料夾都沒有修改的權利。shiwulo不可以進入其他人的目錄，例如：不能進入/home/labBook/xxx

相關函數：

time, gmtime, sprint, geteuid, getuid, getpwnam, chown, readdir, opendir, closedir

評分方式：

1. （30 pt）可以複製一個檔案到/home/labBook/username/中（沒有自動建立以日期為目錄的功能）

舉例：/home/shiwulo/labBook myfile，會把myfile複製到/home/labBook/myfile

（hint：open, read, write, close）

1. （30 pt）可以複製一個檔案到/home/labBook/username/中，如果該檔案有hole，那麼複製出的檔案也有hole（沒有自動建立以日期為目錄的功能）

（hint: lseek）

1. （10 pt）複製的時候自動以日期時間替目錄建立

舉例：/home/shiwulo/labBook myfile，會把myfile複製到/home/labBook/2018-04-22-13:01:23/myfile

（hint：man 2 time, man 3 localtime, sprint, mkdir, chdir）

1. （20 pt）可以複製目錄及其子目錄...（整個目錄樹）
2. （20 pt）研究生可以查閱自己的研究日誌，但不能修改已上傳的研究日誌，不能觀看別人的研究日誌，指導教授可以瀏覽、修改所有研究生的檔案

（hint：man 2 chown, getpwnam，這一題請假設教授的id是ron，研究生的id是std0, std01...std9）

繳交內容：

1. 所有的程式碼
2. makefile：該makefile要能夠編譯你的程式碼，產生執行檔labBook，並且建立labBook資料夾，設定該資料夾權限
3. readme，請依照評分方式，依照1~5寫清楚你做出了哪些功能

例如：

1. 功能完成
2. 功能完成
3. 功能完成
4. 功能部分完成：只能複製第一層的目錄
5. 功能未完成

測試環境：

助教會建立教授的帳號ron，研究生的帳號std0, std1 ... std9，然後進入你所繳交的目錄，執行sudo makefile，助教會根據你的readme測試你的程式碼的正確性，請注意：助教只會輸入「正確的輸入」，因此你不需要做錯誤檢查。此外你的makefile應該要自行建立labbook目錄，例如：mkdir ../labBook

===可執行的程式片段===

/\*

Usage:

sudo chowngrp2root someFile

假設root的id是0

\*/

#include <sys/types.h>

#include <unistd.h>

int main(int argc, char\*\* argv)

{

chown(argv[1], 0, 0);

}

/\*

gettime.c

print time

\*/

#include <stdio.h>

#include <time.h>

int main(void)

{

time\_t t = time(NULL);

struct tm \*pTM = localtime(&t);

int yyyy = pTM->tm\_year + 1900;

int mm = pTM->tm\_mon + 1;

int dd = pTM->tm\_mday;

int hh = pTM->tm\_hour;

int min = pTM->tm\_min;

int ss = pTM->tm\_sec;

printf("%d-%d-%d-%d:%d:%d\n", yyyy, mm, dd, hh, min, ss);

}

/\*

name2id.c

usage: name2id username

\*/

#include <pwd.h>

#include <stdio.h>

#include <sys/types.h>

#include <assert.h>

int main(int argc, char\*\* argv)

{

struct passwd \*user;

user= getpwnam(argv[1]);

assert(user!=NULL);

printf("name:%s\n", user->pw\_name);

printf("uid:%d\n", user->pw\_uid);

printf("gid:%d\n", user->pw\_gid);

}