作業八：

學習目標：

在傳統 UNIIX 系統中，權限只有二種「全給」或「受限」，在比較新的 Linux 支援「capabilities」，可以將 super user 的權限「部分」給予某個應用程式，藉由 capabilities，可以讓 Linux 的權限設定更為細緻，進而增加系統的安全性。

例如：chown_super 的「能力」只有「允許任何可以執行這個檔案的人，改變任意檔案的 owner」，而不是「允許任何可以執行這個檔案的人，擁有所有權限，包含改變檔案的 owner」。

這可以避免 chown_super 被駭客攻擊時，駭客從 chown_super 得到控制系統的所有權限。

先練習：

執行下面命令，黃色底的部分是註解

$ cp /usr/bin/chown ./chown_super 將 chown 複製到當前目錄下

$ sudo setcap CAP_CHOWN+ep ./chown_super 讓 chown_super 擁有更改任意檔案的 owner 的權利。權利的選項可以 man capabilities 查看

http://man7.org/linux/man-pages/man7/capabilities.7.html

這句話的意思是：因為 setcap 需要用超級使用者的權限設定，因此用

sudo 執行，然後賦予 chown_super change owner 的權限（即：CAP_CHOWN），而這個權限的賦予方式是：

Permitted（強制賦予）：無論這個 process 的老爸是誰，都立即賦予他這項權限

Inheritable（允許）：如果這個 process 的老爸有這個權限，那麼他「才會」有這個權限（這個例子不會用到）

Effective：執行這個檔案的時候，所設定的權限有效（我知道這個很怪，但如果設定檔案的「能力」，「e」一定要有）（這個屬性在設定 task 時（我們還沒教怎樣建立 task）的意義才會明確）

$ ./chown_super YOUR_USER_NAME /bin/ls 將/bin/ls 的檔案的 owner 變更為「你自己」

$ ./chown_super root /bin/ls 再將/bin/ls 的檔案的 owner 變回 root

題目：

將 nice 複製到自己的目錄下，名為 nice_pro，必且讓 nice_pro 擁有提高優先權的能力

舉例：

還未設定前：

shiwulo@vm:~$ ./nice_pro -n -10 ls　執行「ls」時將優先等級提高 10

./nice_pro: cannot set niceness: Permission denied

a　chown　chown_super　downloads　files　git　kill_super

nice_pro　snap　strace_pro　workspace

設定後（不會出現權限不足的問題）

shiwulo@vm:~$ ./nice_pro -n -10 ls

a　chown　chown_super　downloads　files　git　kill_super

nice_pro　snap　strace_pro　workspace

報告：

1. 報告上面寫上姓名（可隱匿一個字）和學號
2. 從 man capabilities 裡面隨便挑三個權限，並說明那三個權限是什麼樣的用途（大致上就是英文翻譯成中文再加上一點點自己的理解）

繳交：

1. 程式碼和 makefile，助教執行『sudo make』指令後，必須自動產生 nice_pro。
2. 撰寫報告，格式並須為 pdf。測試報告前請附上姓名（可隱匿一個字）及學號
3. 請將所有檔案壓縮成.tar.bz2。繳交到 ecourse2 上
4. 不能遲交
5. 再次提醒，助教會將所有人的作業於 dropbox 上公開
6. 繳交期限：2021/5/11 早上八點
7. 如果真的不會寫，記得去請教朋友。在你的報告上寫你請教了誰即可。

關於程式碼：

作業八不需要寫程式碼，只要設定 capabilities 就可以了，底下節錄自 capabilities(7)

```
CAP_SYS_MODULE
      * Load and unload kernel modules (see init_module(2) and
        delete_module(2));
      * in kernels before 2.6.25: drop capabilities from the system-
        wide capability bounding set.
```

CAP_SYS_NICE

* Raise process nice value (nice(2), setpriority(2)) and change the nice value for arbitrary processes;
* set real-time scheduling policies for calling process, and set scheduling policies and priorities for arbitrary processes (sched_setscheduler(2), sched_setparam(2), sched_setattr(2));
* set CPU affinity for arbitrary processes (sched_setaffinity(2));
* set I/O scheduling class and priority for arbitrary processes (ioprio_set(2));
* apply migrate_pages(2) to arbitrary processes and allow processes to be migrated to arbitrary nodes;
* apply move_pages(2) to arbitrary processes;
* use the MPOL_MF_MOVE_ALL flag with mbind(2) and move_pages(2).