

## 作業六：mybusybox

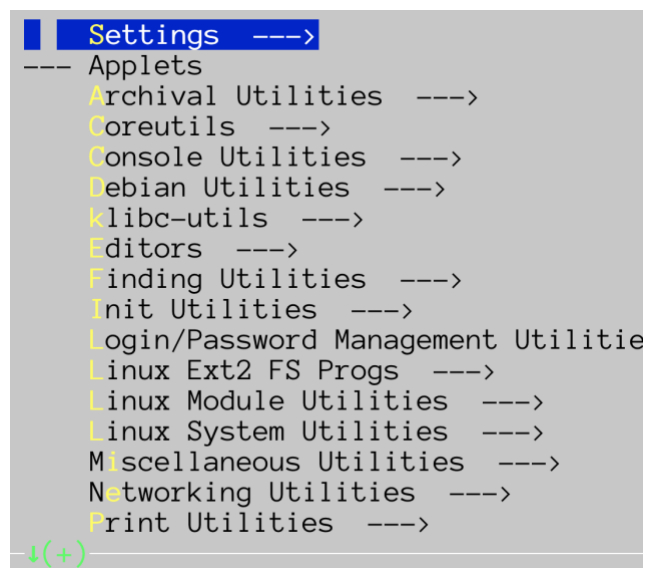
### 學習目標：

- 了解嵌入式系統的 busybox 的目的
- 了解如何透過 softlink 讓同一個程式有不一樣的行為
- 能夠透過 man 了解 system 這個函數怎樣使用

### 題目：

#### A. 安裝 busybox

- `git clone https://git.busybox.net/busybox/`
- 詳細的安裝步驟請參考 INSTALL
- `sudo apt-get install libncurses5-dev` (這是 make menuconfig 時需要的函數庫)
- `make defconfig`
- `make menuconfig`, 最起碼選擇下列的二個選項



```
Settings ---->
--- Applets
  Archival Utilities ---->
  Coreutils ---->
  Console Utilities ---->
  Debian Utilities ---->
  klibc-utils ---->
  Editors ---->
  Finding Utilities ---->
  Init Utilities ---->
  Login/Password Management Utilities
  Linux Ext2 FS Progs ---->
  Linux Module Utilities ---->
  Linux System Utilities ---->
  Miscellaneous Utilities ---->
  Networking Utilities ---->
  Print Utilities ---->
↓(+)
```

```

t(-)
(/var/run) Directory for pidfiles
[*] Include busybox applet
[*] Support --show SCRIPT
[*] Support --install [-s] to install applet links
Don't use /usr
Drop SUID state for most applets
[*] Enable SUID configuration via /etc/busybox.conf
[*] Suppress warning message if /etc/busybox.conf
[*] exec prefers applets
(/proc/self/exe) Path to busybox executable
[*] Support NSA Security Enhanced Linux
Clean up all memory before exiting (usually not ne
[*] Support LOG_INFO level syslog messages
--- Build Options
[*] Build static binary (no shared libs)
[*] Force NOMMU build
[*] Cross compiler prefix
t(+)
<Select> < Exit > < Help >

```

```

t(-)
[*] Support NSA Security Enhanced Linux
Clean up all memory before exiting (usually not
[*] Support LOG_INFO level syslog messages
--- Build Options
[*] Build static binary (no shared libs)
[*] Force NOMMU build
[*] Cross compiler prefix
[*] Path to sysroot
(-g -z lazy) Additional CFLAGS
[*] Additional LDFLAGS
[*] Additional LDLIBS
[*] Avoid using GCC-specific code constructs
[*] Use -mpreferred-stack-boundary=2 on i386 arch
[*] Use -static-libgcc
--- Installation Options ("make install" behavior)
What kind of applet links to install (as soft-li
(./_install) Destination path for 'make install'
t(+)

```

- make
- make install
- \$ sudo chroot ~/busybox/\_install/ ash (這是在 chroot 的情況下，執行 busybox) (要離開 busybox，輸入 exit)

B. 寫一隻程式叫做 myBusybox，並且建立如下的 softlink，這些 softlink 都連接到 myBusybox，myBusybox 會依照 softlink 的名稱，使用 system() 執行相對應的功能

- cp
- ls
- cat
- chown

請注意，當助教執行 make 以後，請產生 myBusybox 和上述四個 softlink，並將這些檔案放在跟 makefile 相同的資料夾中

報告：

- 執行 readelf -d /usr/bin/ls, readelf -d ~/busybox/\_install/bin/busybox。請在報告上列出這二個程式所需要的『動態鏈結函數庫』
- 請給一張螢幕快照，顯示『\$ sudo chroot ~/busybox/\_install/ ash』的執行結果

- 請參考 <https://zh.wikipedia.org/wiki/BusyBox>, 以 50 個左右的字說明 busybox 的主要用途 (你可以直接從 wikipedia 上把相關文字複製貼上到你的報告)

繳交：

1. 程式碼和 makefile, 助教執行『make』指令後, 必須自動產生 myBusybox 及四個 soft link (即: cp、ls、cat、chown)
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