

Lab 4 簡易名字跑馬燈

I. Introduction

請在 Raspberry Pi 上撰寫一支名字跑馬燈。在 Raspberry Pi 上透過 writer 程式將英文字母寫到 driver 當中，透過 reader 程式將該字從 driver 中讀出來，最後透過 socket 傳遞給 VM 上頭的 [seg.py](#) 程式，其會把該字用十六段顯示器 (GUI) 呈現出來。

II. Specification

- **driver (自行撰寫)**

- 撰寫自行定義的 write function，將 writer 傳送過來的計數值給存起來。
- 撰寫自行定義的 read function，將經過處理之後的計數值回傳給使用者。
- 所謂的處理，就是將字轉換為16段顯示器可以接受的資料格式 - **1個長度為 16 的陣列，每一格儲存的不是 0 就是 1。**
- 16段顯示器顯示資訊表

```
bits_for_seg = (
    1st bit    # top left
    2nd bit    # top right
    3rd bit    # upper right
    4th bit    # lower right
    5th bit    # bottom right
    6th bit    # bottom left
    7th bit    # lower left
    8th bit    # upper left
    9th bit    # upper left slash
    10th bit   # upper middle
    11th bit   # upper right slash
    12th bit   # middle right
    13th bit   # lower right slash
    14th bit   # upper middle
    15th bit   # lower left slash
    16th bit   # middle left
)

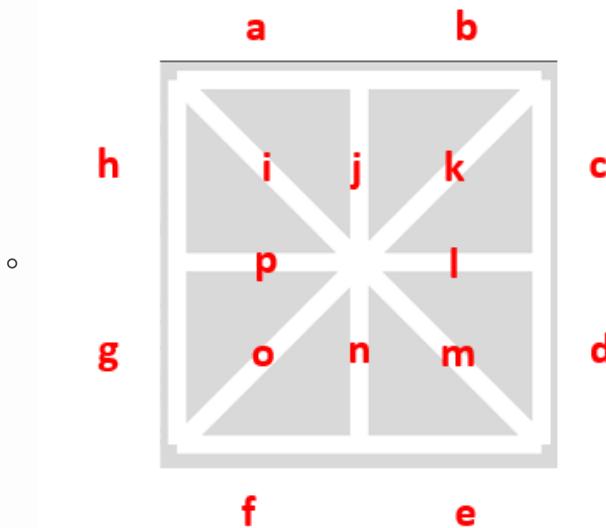
seg_for_c[27] = {
    0b1111001100010001, // A
    0b0000011100000101, // b
    0b1100111100000000, // C
    0b0000011001000101, // d
    0b1000011100000001, // E
    0b1000001100000001, // F
    0b1001111100010000, // G
    0b0011001100010001, // H
    0b1100110001000100, // I
```

```

0b1100010001000100, // J
0b0000000001101100, // K
0b0000111100000000, // L
0b0011001110100000, // M
0b0011001110001000, // N
0b1111111100000000, // O
0b1000001101000001, // P
0b0111000001010000, // Q
0b1110001100011001, // R
0b1101110100010001, // S
0b1100000001000100, // T
0b0011111100000000, // U
0b0000001100100010, // V
0b0011001100001010, // W
0b0000000001010100, // X
0b00000000010100100, // Y
0b1100110000100010, // Z
0b0000000000000000
};

}

```



- **writer** (自行撰寫)

- 每隔一秒，就將新的字母寫到 driver 當中。

- **reader** (助教提供)

- 每隔一秒，就去讀取 driver，透過 socket 將資料傳遞給 [seg.py](#) 程式。

- [seg.py](#) (助教提供)

- 當作16段顯示器，將收到的資料，透過 Tkinter library 給顯示出來。

- 檔案連結:

https://drive.google.com/drive/folders/16X-oVUCSW33w5DffnL2dhy0D_kalKIWM?usp=sharing

III. Illustration

RPi

- driver

```
sudo insmod mydev.ko
```

- writer

- <name> 為英文名字。

```
./writer <name>
```

- reader

- <ip>, <port> 填入 VM 之 ip address。

- <dev> 填入 mknod 建立的 character device 名稱，如 /dev/mydev

```
./reader <ip> <port> <dev>
```

VM

- seg.py

- <port> 填入 socket 聆聽的端口。

```
python3 seg.py <port>
```

IV. Note

- driver 的撰寫請參考 lab pdf 的範例程式碼。
- VM 上須安裝 Tkinter library，以利程式正確執行。

```
sudo apt install python3-tk
```

- 由於 driver 與一般程式 (reader, writer) 所使用的定址空間不同，在傳遞資料的時候需要透過 **copy_from_user()**, **copy_to_user()** 這兩個 function 來協助完成。

V. Demo & Submission

- 請透過 make 進行 **cross compile** (請勿直接在 RPi 上編譯)
- 助教會提供 [demo.sh](#) 來協助同學 Demo。
- 執行 .sh 前務必先 "mknod"，並修改 .sh 裡的 IP 等資訊
- 請將程式碼以下列的格式擺放與命名，以方便助教評分。

```
lab4_<學號_姓名>.zip
|-- Makefile
|-- mydev.c
|-- writer.c
|-- reader.c
|-- seg.py
|-- demo.sh
|-- lab4.mp4
```

- 請將上述之資料夾壓縮為單一 zip 檔案，並上傳到 E3 上。
- 展示影片(90秒內，超過會扣分)：
 - 展示你的名字跑馬燈(一定要拍到跑馬燈本人)
 - 說明如何實作 mydev.c 及 writer.c
 - 請在畫面展示程式碼重點並口頭講解
 - 助教主觀認定有手抖就扣10分，所以建議**用螢幕錄影**