本次目標：

1. 練習寫 Client APP (CA) 跟 Trusted APP (TA)，並用 QEMU 的環境測試

2. 簡單使用 TEE Client API 跟 TEE Internal Core API

3. 練習 bitwise operations (Driver 一定會用到)

每個 APP 都有兩個部分

CA：

1. 下 TA Command 跟 TA 要系統時間，TA 回傳過來後在視窗中輸出秒數跟毫秒數。

系統時間要取得兩種：

(1) REE 的時間 (格式為, 從 1970 年 1 月 1 號開始，到我們當下所經過的秒數跟毫秒數)

(2) TEE 的時間 (格式為, QEMU 開機後，系統經過的秒數跟毫秒數)

2. 傳送 Instruction (這裡是指計組的 machine code) 給 TA，TA 解析並計算後會回傳結果，輸出考慮兩種 instruction，每個都是 16 bit 長度 (包含 opcode, destination, source, value, flag, etc)

(請先參考 https://justinmeiners.github.io/lc3-vm/，這是一個實作虛擬機的 project，會教我們如何解析 instruction / machine code)

我們要傳送下列兩種 instruction 給 TA，TA 的部分會再去解析

(1) ADD：0001000011000011 → 3 + 3 = 6，回傳 6 給 CA (TA 會照著上面網頁的格式去解析)

(2) Bitwise AND： 0101000111000010 → 7 & 2 = 2，回傳 2 給 CA

(上面的 0001… 是二進位表示法，是 16 bits 的 integer，例如 0001000011000011 = 4291，所以 CA 會傳送 4291 給 TA，TA 必須用 bitwise operation 解析這串數字的內容，格式在上面的網頁中有，但是這邊我們先不考慮 register，直接當成數字操作就好，並且只需要考慮 source)

TA：

1. 接收到 CA 請求系統時間，呼叫 TEE Internal Core API 取得 TEE 跟 REE 的系統時間，並回傳結果 (會有一個資料結構 TEE_Time，可以參考 API 的文件或去翻 optee_os 的 source code)

2. 用 bitwise operations 解析 CA 傳過來的 instruction code，計算並回傳結果

參考 https://justinmeiners.github.io/lc3-vm/ 寫出解析 instruction 的 function

範例寫法：

```
switch( parse_instruction(instruction) ) {
    case ADD: …
    case AND: …
}
```

所以總共有三個 TA command：

(1) TA_CMD_GET_SYSTEM_TIME

(2) TA_CMD_GET_REE_TIME

(3) TA_CMD_CALC_INSTRUCTION

將回傳結果截圖上傳到 Skype 群組。

期限 ：2021/5/7 23:59 前

第一部分的參數 ：

CA Send	CA Receive	TA Receive	TA Return
TA_CMD_GET_SYSTEM_TIME	TEE 的系統時間印出秒數/毫秒數	TA_CMD_GET_TEE_TIME	TEE 的系統時間(秒數、毫秒數)
TA_CMD_GET_REE_TIME	REE 的系統時間印出秒數/毫秒數	TA_CMD_GET_REE_TIME	REE 的系統時間(秒數、毫秒數)

第二部分參數 ：

CA Send	CA Receive	TA Receive	TA Return
TA_CMD_CALC_INSTRUCTION 還有 ADD Instruction (例如上面的 4291)	TA 解析 Instruction 後算出來的結果	TA_CMD_CALC_INSTRUCTION 跟 ADD Instruction	計算的結果
TA_CMD_CALC_INSTRUCTION 還有 AND Instruction	TA 解析 Instruction 後算出來的結果	TA_CMD_GET_REE_TIME 跟 AND Instruction	計算的結果