

# Lab 6 Web ATM

EOS Lab 6 Web ATM



## I. Introduction

請在 VM 上撰寫一隻 socket 程式。多個客戶端同時連線進來對同一個帳戶進行存款與提款的動作，而伺服器端要能正確處理同時存取所導致的 race condition 問題，才不至於造成客戶金錢上的損失。

## II. Specification

- Server (自行撰寫)
  - 建立 socket server，等待 client 來建立連線。
  - server 端要能夠支援同時多人連線的情境。
  - 連線建立後，要根據客戶的要求做**存款**或是**提款**的動作。
  - 伺服器端只會有一個帳戶，由多人共用。
  - 輸出每次動作完成後，輸出帳戶的餘額。

After deposit: 2

After withdraw: 1

- Client (自行撰寫)
  - 撰寫 socket client，指定要求(存款或提款、多少錢)，然後將該資訊傳給 server 處裡。

### III. Illustration

---

- Server

```
./server <port>
```

- <port> 為 server listen 的端口

- Client

```
./client <ip> <port> <deposit/withdraw> <amount> <times>
```

- <ip> 為 server 所在的 ip
- <port> 為 server listen 的端口
- <deposit/withdraw> 存款或提款
- <amount> 多少錢
- <times> 該動作將重複執行多少次

### IV. Note

---

- semaphore 的撰寫請參考 lab 6 的範例程式碼。
- 請透過 semaphore 來處理同時存取造成的 race condition 問題。
- 請在 server 程式結束時 (Ctrl + C)，需在程式結束前將建立出來的 semaphore 給清除。
- 先ctrl-b，再用指令來讓視窗滾動

```
:set mouse on
```

- 程式結束後請使用 `ipcs -s` 指令檢查是否清除semaphore。

### V. Demo & Submission

---

- 助教有提供 demo.sh ([https://drive.google.com/file/d/1sfT2zIHINJMQafF5eirUteZ7PX62oK\\_P/view?usp=sharing](https://drive.google.com/file/d/1sfT2zIHINJMQafF5eirUteZ7PX62oK_P/view?usp=sharing)) 來協助同學測試與 Demo。
- 請將程式碼以下列的格式擺放與命名，以方便助教評分。

```
xxxxxxxxx_eos_lab6
|-- Makefile
|-- server.c
|-- client.c
|-- demo.sh
|-- lab6.mp4
```

- 請將上述之資料夾壓縮為單一 zip 檔案，並上傳到 E3 上。

- 繳交截止時間：05/24 (五)
- 30秒展示影片：請說明如何實作server.c及client.c、利用demo.sh展示成果(須看得出金額加減交錯)，並使用 `ipcs -s` 指令檢查是否正確清除semaphore。