Network Administration and System Administration

Midterm Examination

Time: 2025/04/07 09:10 - 12:10

Instructions and Announcements

- 考試時間共三小時，三人一組考試。分組連結及簽到結果。
- 考試期間禁止使用手機、電話、任何通訊軟體等與同組成員外任何人聯繫，也禁止組與組之間**一切討論與合作**，如被發現視為作弊行為，**期中考 0 分**，並依校規懲處。
- MacOS 的 VirtualBox、arm64 版本的各種作業系統及軟體，其可靠度、穩定度可能低於 x86_64 的版本。**請審慎評估是否合適利用 ARM64 來進行考試題目的解題。**
- 作答過程中請自行斟酌備份，避免電腦發生意外，損失過多進度，可考慮準備隨身碟或雲端空間備份進度。
- 為避免發生重大意外，請自行注意 VM 用量，同時開啟過多 VM 可能導致電腦當機，我們恕不負責。
- 線上考試的 Announcement 將會更新在 公告。
- 題目檔案壓縮檔可統一至 雲端硬碟下載，解壓縮密碼將會在考試開始後公佈。
- 完成題目時請至 Submission Form 上傳作答內容，每組每個 subtask **最多上傳 3 次**。
- 計分板連結 Scoreboard。
- 各題後面黑色星號數目代表我們估計的難度。請參考，可用來決定解題順序。
- 題目可能難免有疏誤之處。若發現有解不開題目、題敘不清的狀況，請盡快跟助教或老師反應，或斟酌時間先解別的題目。
- 滿分為 100 pts。

1 Shell script: Read The Manual! ★ ∼ ★★★ (19 points)

眾所周知，man page 是我們的好夥伴。但是你卻發現你多半時間都在用 google 搜尋 man page 的內容？讓我們套用一些神奇的演算法，來自制一個簡易的搜尋引擎吧！我們蒐集了一部分的 linux man page 的內容，請你寫一個 shell script 對這些文本做以下分析，這個 shell script 需支援`--task T`選項，並根據 T 的數值去做不同的分析，不同分析所需的參數均附在`--task T`後面。

本題所需要的檔案有：

- `library/`

1. (5 points) (★) 單字頻率分析 (`--task 1`) 首先，為了瞭解每個文本的內容，你想把文本拆成不同的單字，並計算他們出現的頻率。

   我們用**所有非英文字母的字元做分隔點來切分單字**，也就是說只要兩個英文字母之間有任何一個非英文字母的字元，那這兩個字母就不在同一個單字裡。同時避免大小寫的問題，**所有字母都統一轉為小寫**。

   例如，將以下文字：

   ```
   xxd xxd [OPTIONS] [FILE]
       Hex dump FILE (or stdin)
           -g N       Bytes per group
           -c N       Bytes per line
   ```

   解析出來的單字應為：
   xxd xxd options file hex dump file or stdin g n bytes per group c n bytes per line

   你的 shell script 需要接受兩個參數 (由左到右)：文本路徑、R。你要輸出文本內出現次數最高的前 R 個單字以及其出現次數，以冒號隔開。輸出的順序為先照出現次數由大到小，若有相同次數的，則再由字典序由小到大排序。

2. (6 points) (★★★) TF-IDF (`--task 2`) 接下來，我們希望找出每個文本裡面有哪些重要的單字，這有助於我們更加了解文本的內容。

   對出現在文本 T_j 的單字 w 以及一群文本 $\mathfrak{T}$，我們可以定義 TF-IDF 值如下：

   $$\text{TF-IDF}_{\mathfrak{T}}(T_j, w) = (1 + \ln(f(T_j, w))) \ln \frac{N}{n(\mathfrak{T}, w)} \quad , \tag{1}$$

   其中 $f(T_j, w)$ 為單字 w 在文本 T_j 的出現次數，$N(\mathfrak{T})$ 為 $\mathfrak{T}$ 的文本總數，$n(\mathfrak{T}, w)$ 為 $\mathfrak{T}$ 中出現單字 w 的文本總數。

   你的 shell script 需要接受 3 個參數：存有所有文本 $\mathfrak{T}$ 的目錄路徑、目標文本 T_j 的路徑、輸出的 TF-IDF 值數量 R。你必須輸出目標文本 T_j 中出現的所有單字中，TF-IDF 值前 R 大的單字以及其 TF-IDF 值到小數點後第 4 位，中間以冒號隔開，而輸出的順序為先照 TF-IDF 值排序由大到小。若有相同 TF-IDF 值者，則依照字典序由小到大排序。

3. (8 points) (★★★) BM25 (`--task 3`) 有了 TF-IDF，我們已經可以做出基本的搜尋引擎了。只要把詢問的字串一樣轉成 TF-IDF 的向量，就可以用內積的方法計算問題與文件的相似度了。但或許你已經發現 TF-IDF 似乎不太準確。基於 TF-IDF 的概念加以延伸，我們來實作一個經典的搜尋演算法：BM25！

對一群文本 $\mathfrak{T}$、$\mathfrak{T}$ 其中的一個文本 T_j、以及詢問句 q，BM25 $\mathcal{B}_{\mathfrak{T}}(q, T_j)$ 可計算如下：

$$\mathcal{B}_{\mathfrak{T}}(q, T_j) = \sum_{\{w \in q \text{ and } w \in T_j\}} \frac{(K_1+1)f(T_j, w)}{K_1(1-b+b\frac{d(T_j)}{\bar{D}}) + f(T_j, w)} \times \ln(\frac{N - n(\mathfrak{T}, w) + 0.5}{n(\mathfrak{T}, w) + 0.5} + 1) \quad , \quad (2)$$

其中 q 是詢問的句子、而 $\{w \in q \text{ and } w \in T_j\}$ 的條件則代表每個同時出現在 q 以及 T_j 中的 w 都列入加總、$d(T_j)$ 是文本 T_j 的單字總數、$\bar{D} = \frac{\sum_{T \in \mathfrak{T}} d(T)}{N(\mathfrak{T})}$ 是所有文本單字總數的平均值，$K_1 = 1.5$ 和 $b = 0.75$ 是兩個預先設定的常數。這樣算出來的 BM25 值就是文本 T_j 與 q 的相似度了!

你的 shell script 需支援 2 個參數：文本目錄路徑、R，並且從標準輸入讀取要詢問的句子。你的程式要輸出前 R 個最相似的文本之檔案名稱以及其 BM25 相似度、以冒號隔開。BM25 相似度輸出到小數點下四位。每個輸出項目的第二行，則輸出該文本的的第四行的內容 (在本題的檔案中，通常為說明指令功能的那一行，請見範例)。輸出的順序為先照相似度排序由大到小、若相似度相同則由檔名字典序由小到大排序。

備註：

你可以假設本題的選項/輸入都是正確的。

浮點數的輸出統一**四捨五入**到小數點後第四位，但在計算時可以盡量精準。輸出誤差不能超過 0.002

本提限用 shell script，不可以在 script 裡呼叫/執行其他程式語言 (內建指令如 awk、bc、sed 可以用)，如果不確定哪個指令可以用可以詢問助教。

我們會用工作站進行測試，並且使用相同的文本做測試。建立與修改任何檔案都不被允許

Task 1 的時間限制是 10 秒。Task 2 的時間限制是 30 秒。Task 3 的時間限制是 120 秒。

Task 3 中詢問的單字數量至多 100。

範例輸入/輸出

- Task 1

  命令列：`./ans.sh --task 1 library/2.txt 5`

  輸出：

  ```
  the:894
  to:487
  is:286
  a:276
  of:267
  ```

  命令列：`./ans.sh --task 1 library/10.txt 10`

  輸出：

  ```
  the:41
  apport:29
  report:22
  a:20
  to:20
  package:17
  ```

```
is:14
and:13
file:13
or:13
```

- Task 2
  命令列：`./ans.sh --task 2 library library/50.txt 5`

  輸出：

```
byobu:11.7417
tmux:11.6184
layout:10.0951
layouts:8.5523
restore:6.2896
```

  命令列：`./ans.sh --task 2 library library/934.txt 10`

  輸出：

```
xxd:29.9884
hexdump:19.6508
octets:14.5136
autoskip:13.9780
nugent:13.9780
seek:13.1267
tony:12.5860
infile:12.3026
lseek:11.2773
postscript:10.9319
```

- Task 3
  命令列：`./ans.sh --task 3 library 5`
  輸入：`how to convert hex string to binary?`

  輸出：

```
934.txt:14.5761
       xxd - make a hexdump or do the reverse.
381.txt:12.6142
       git-replace - Create, list, delete refs to replace objects
1.txt:12.5829
       BusyBox - The Swiss Army Knife of Embedded Linux
434.txt:10.9666
       gpg-connect-agent - Communicate with a running agent
527.txt:10.7072
       loadkeys - load keyboard translation tables
```

Submission

在 Google Form 上傳檔案。

---

2 Partition ★ ∼ ★★★ (20 points)

Resource

本題所需要的檔案有：

- `sda.qcow2`
- `sdb.qcow2`
- `sdc.qcow2`
- `run.sh`

注意以下幾點：

1. 以下是範例指令 run.sh，請自行依根據需要修改
2. 範例指令將會在背景執行一個掛載了三個硬碟映像檔案的虛擬機。
3. 此虛擬的 console 已經設定可用 VNC 連接，但請自行調整連接埠。
4. 網路的部分已經設定了 22 連接埠的 Port forwarding。請自行調整連接埠號碼。
5. 我們已經在 nasaws4.csie.ntu.edu.tw 上測試過，可以成功執行。
6. 如果想要使用其他虛擬機軟體，例如 VirtualBox，請自行解決碰到的問題。注意 sda.qcow2 中安裝的檔案是 x86_64 的 Linux 作業系統，因此**可能無法直接在非 x86_64 架構的機器上直接執行**。

以下是範例指令：

```
qemu-system-x86_64 \
  -name vm-demo \
  -m 2048 \
  -smp 2 \
  -hda sda.qcow2 \
  -hdb sdb.qcow2 \
  -hdc sdc.qcow2 \
  -netdev user,id=net0,hostfwd=tcp::2222-:22 \
  -device e1000,netdev=net0 \
  -vnc :1 \
  -monitor unix:/tmp/qemu-monitor-socket,server,nowait \
  -pidfile /tmp/vm.pid \
  -daemonize
```

username: `nasa`, password: `nasa2025`

Tasks

只有 Tasks 3,4 有相依性，需要依照順序完成，其餘問題可以獨立、以任意順序完成。

1. (6 points) (★) 請寫一份 shell script 做到：
   (a) 每次執行會產生一份 lv-data1 的 snapshot。
   (b) snapshot 各佔 0.5 GiB。
   (c) 如果執行前有超過三個從 lv-data1 產生的 snapshot，則僅留下兩個最新的 snapshot，而把其他的刪除。在這個狀況下，執行後會留下最新的三個 snapshot。
   (d) 使用`snapshot-<%Y%m%d_%H%M%S>` 作為名稱。舉例來說，2025 年 3 月 22 日 3:48:23 建立的 snapshot 叫做：`snapshot-20250322_034823`（時區隨意）。

   注意：Demo 前請先將 snapshot 全部刪除。

2. (8 points) (★★★) 將原本在 /dev/md127 上的 RAID5 的陣列拆掉改組成 RAID0，命名為/dev/mdTeam-id，並將 /dev/sdc3 釋放出來。這邊/dev/sdc3 指的是在 sdc.qcow2 硬碟檔案中的第 3 個 partition。在操作過程中，請注意以下幾點：
   (a) 禁止複製檔案。不可使用如`mv`, `rsync`等指令來拷貝或者轉移檔案。
   (b) 新的 RAID0 volume 需要跟原本的 RAID5 volume 大小相同，操作過程中任何時刻不可毀損、刪除 volume 中的檔案。
   (c) 在資料從 RAID5 轉移到 RAID0 的過程中的任何時間點（不包含重新掛載裝置的短暫時間），必須仍能夠存取 RAID volume 上的檔案，而不能在轉移過程中將檔案服務下線。
   (d) 允許增加硬碟，但是改組後的 RAID 不可使用新增加的硬碟。
   (e) 請確認重開機後，可以自動掛載新組成的 RAID0 volume 到/mnt/raid。
   (f) Hint:
   - 可使用新增加的硬碟及 RAID1 來進行不下線的檔案轉移。
   - RAID1 使用過後可以想辦法丟棄。
   - 可使用 loop device (參見 losetup 指令) 來跳過一個 partition 上面最前面某個 offset 長度後進行掛載。而 md device 的 data offset 可以用`mdadm --examine /dev/<device>`來觀察。

   注意：請確實記錄每個執行步驟的指令，並且在向助教 demo 時展示完整的操作指令歷程。

3. (3 points) (★) 請將/dev/nasa_vg/lv2 在不影響內部檔案的情況下，縮小成 500MiB 。並掛載到/mnt/lv1。最後可使用的檔案系統空間也要隨著 logical volume 的大小而調整。

   注意：請確實記錄每個執行步驟的指令，並且在向助教 demo 時展示完整的操作指令歷程

4. (3 points) (★) 承上題，將多出的空間全部分給/dev/nasa_vg/lv1。並掛載到/mnt/lv2。最後可使用的檔案系統空間也要隨著 logical volume 的大小而調整。

   注意：請確實記錄每個執行步驟的指令，並且在向助教 demo 時展示完整的操作指令歷程

Submission

找助教 Demo。

3 DNS ★ ∼ ★★ (17 points)

Resources

- 請利用 debian 12.9.0 netinst ISO 安裝虛擬機完成本題，雲端上提供的檔案有：
  - p3-arm64
    - debian-12.9.0-arm64-netinst.iso
    - run_vm.sh
  - p3-x86_64
    - debian-12.9.0-amd64-netinst.iso
    - run_vm.sh
- run_vm.sh
  - 執行方法：將 debian iso 檔和此檔案放在你的工作目錄後執行./run_vm.sh
  - 若在工作站上作答此題目，為避免 MAC address, port 的衝突，請使用此腳本。
  - 這個腳本能根據 Group ID 與 VM ID 自動分配名稱、MAC address、工作站上的 ssh, VNC 和 webserver 的連接埠，並自動架設/執行 VM。
  - 自動設定兩張網卡：第一張為內部通訊使用 (VDE)，第二張為能連外網的 NAT 網卡。
  - 開機後仍需要到 /etc/network/interfaces 設定 static IP 位址。
  - 若你想自己執行 QEMU 指令，這裡也附上指令。請根據你們的 TeamID 和 VM-ID(第幾台 VM) 修改 Port 和 MAC address

```
#Offset = 10*(TeamID)+(VM-ID)
$ qemu-system-x86_64 \
-cpu host -m 2048 \
-enable-kvm \
-drive file=group[TeamID]_vm[VM-ID].qcow2,format=qcow2 \
-cdrom debian-12.9.0-amd64-netinst.iso \
-boot d \
-monitor stdio \
-device e1000,netdev=vdenet,mac=52:54:00:[GroupID(16)]:00:[VMID(16)] \
-netdev vde,id=vdenet \
-netdev user,id=netuser0,hostfwd=tcp::[2200+Offset]-:22, \
hostfwd=tcp::[8000+Offset]-:80 \
-device e1000,netdev=netuser0 \
-vnc :[Offset]
```

Story

在 Kivotos 世界有個巨大的圖書館，在這裡每本書都擁有獨特的位置標籤，而這些標籤是以 **IP 位址** 的形式表示。Ui 是這座圖書館的管理員，每當有人來詢問書名（Domain Name）時，她就必須在記憶中搜尋對應的書架位置（IP），然後告訴來訪者。起初，來訪者不多，Ui 能輕鬆應對。然而，隨著 Kivotos 世界的人對知識的渴望越來越強烈，前來查詢的讀者數量激增，讓 Ui 感到不堪重負。

Tasks

1. **(5 points) (★) Ui 的困擾 (Build a PowerDNS server)**

   為了解決這個問題，她決定建立一台 DNS server，讓機器來幫助她自動回應查詢。請幫她建立一台 DNS server。

   (a) DNS server 設定
   - 使用 Debian 12.9 建立一台名為 `nasa-dns-primary` 的 PowerDNS 伺服器
   - 這台伺服器的 Server IP 設定為 `10.0.[TeamID].1`，而 `10.0.[TeamID].0/24` 為其所在的 subnet。
   - 安裝 PowerDNS Admin 作為 Web 管理介面。

   (b) Zone 與 Record 設定
   - 在 `nasa-dns-primary` 上新增一個 zone，名稱為 `nasa.local`，並在該 zone 中加入以下記錄：
     - `exam[TeamID].nasa.local` 指向 `10.0.[TeamID].100`
     - `www.exam[TeamID].nasa.local` 指向 `exam[TeamID].nasa.local`
   - 在 `nasa-dns-primary` 上再新增一個 zone 處理你的 subnet 的反解，並在該 zone 中加入以下記錄：
     - `10.0.[TeamID].100` 指向 exam[TeamID].nasa.local

   (c) Demo (請先呼叫助教)
   - 請透過 PowerDNS Admin 介面展示已建立的 zone 與記錄。
   - 使用 `dig` 指令在本機查詢各項記錄，證明配置成功。

2. **(6 points) (★★) Ui 的危機 (Master/Slave DNS)**

   有一天，Ui 在圖書館內喝著咖啡，卻不小心將整杯咖啡灑在了 nasa-dns-primary 伺服器上，導致機器短路，整個 DNS 服務停擺。來訪者們無法查詢書籍的位置，圖書館陷入混亂，Ui 才意識到高可用性 (High Availability, HA) 的重要性。她決定建立一個備份伺服器，確保即使主機故障，服務仍能繼續運作。

   (a) Slave DNS Server 設定
   - 使用 Debian 12.9 建立另一台名為 `nasa-dns-secondary` 的 VM
   - IP 設定為 `10.0.[TeamID].2`)，subnet 設定與 `nasa-dns-primary` 的 VM 相同
   - 安裝 PowerDNS 作為 Slave DNS server
   - Hint: 可以直接 clone 上一題的機器，再手動更改 IP

   (b) Master/Slave 同步設定
   - 充分設定 Master 與 Slave server 使得 `nasa-dns-primary` (Master) 上 zone `nasa.local` 的記錄，能夠自動同步至 `nasa-dns-secondary` (Slave)。

   (c) Demo(請先呼叫助教)
   - 請於 `nasa-dns-primary` 上透過 PowerDNS Admin 新增或修改任何一筆記錄。
   - 在 `nasa-dns-secondary` 上使用 `dig` 測試，若能及時查詢到更新後的結果，即表示同步成功。

3. **(3 points) (★) Ui 的發現 (Load Balancing)**

   即使 Ui 成功建立了 Master-Slave 架構，她很快發現了一個新的問題——來訪者的查詢請求仍然過於集中在 Master，使得 Master 的反應速度變慢而 Slave 則大部分時間閒置。

   因此 Ui 決定透過 DNSdist 來自動將查詢請求分配至不同的伺服器，使 `nasa-dns-primary` 和 `nasa-dns-secondary` 之間的負擔更為均衡。

(a) 建立 DNSdist Server

- 使用 Debian 12.9 建立第三台 VM，名為 `nasa-dns-dist`。
- IP 設定為：`10.0.[TeamID].3`。subnet 設定和之前的兩台 VM 相同。
- 安裝並啟用 DNSdist，可接受 DNS request。
- 將 `nasa-dns-dist` 設為可處理的 DNS request 的伺服器，會將得到的 request 以 Round-robin 的方式平均傳給 `nasa-dns-primary` 和 `nasa-dns-secondary` 進行解析。

(b) Demo (請先呼叫助教)

- 請先在 `nasa-dns-primary` 與 `nasa-dns-secondary` 上，各自建立一個名為 `test.local` 的 zone，並在該 zone 裡對 `dist[TeamID].test.local` 加入 TXT 記錄，內容分別為 `"I'm primary."` 與 `"I'm secondary."`
- 在 `nasa-dns-dist` 上以 `dig @10.0.[TeamID].3` 和 `dig @localhost` 對該 record 查詢 TXT 紀錄，檢查回應是否都會有 `"I'm primary."` 與 `"I'm secondary."` 交替出現，以證實負載分配成功。

4. **(3 points) (★) Koharu 的請求 (DNS over https)**

DNS protocol 是使用明文的 UDP 傳輸，這意味著 DNS 的查詢是可以被監聽的。這讓圖書館的常客，Koharu，感到不安，因為她不希望自己的閱讀紀錄被大家知道。

為了解決這個問題，Koharu 找到了 Ui，請求她在圖書館的 DNS 系統中開放 DNS-over-HTTPS (DoH)，讓 DNS 查詢變成加密傳輸，保護用戶的隱私與安全。

(a) DoH 設定

- 在 `nasa-dns-dist` 上啟用 DNS-over-HTTPS (DoH)，允許用戶端透過 HTTPS 進行 DNS 查詢。
- 設定 DNSdist 接受 DoH request，並將其轉發至 `nasa-dns-primary` 與 `nasa-dns-secondary` 進行解析。

(b) Demo

- 使用 `dig +https` 測試 DoH 服務是否能正常運作，並確保查詢結果與標準 DNS 查詢一致。

Submission

- 請呼叫 TA 並進行 demo。

---

4 OPNsense ★ ∼ ★★★ (18 points)

註：建議使用 x86_64 架構的設備（如 204 教室的電腦）的 VirtualBox 作答本題。如果一定需要用 ARM 架構的 Mac 作答本題，建議將本題所有 VM 內網路介面的 MTU 設定為 < 1400 bytes 的大小。

Resources

雲端上提供的檔案有：

- `p4-arm64`

  - `alpine-standard-3.21.3-aarch64.iso`
  - `OPNsense-25.1-ufs-serial-vm-aarch64.qcow2.bz2`
- `p4-x86_64`
  - `OPNsense-25.1-ufs-serial-vm-amd64.qcow2.bz2`
  - `OPNsense-25.1-vga-amd64.img.bz2`
  - `alpine-standard-3.21.3-x86_64.iso`
  - 這裡 OPNsense 的相關檔案擇一使用即可
- 檔案和 lab、作業用的一模一樣，已經有的同學不必再次下載。

Description

- 請在你的電腦開一台 OPNsense 虛擬機。這台虛擬機應該至少要有一個對外網路介面，和一個對內網路介面。對內的網路介面要建立 VLAN 11。
  - 對外的網路介面可以自由設定，但是必須可以存取 Internet。建議使用 VirtualBox 的 NAT Network（因為第二部份會需要將兩臺 VM 放在同一個可以存取 Internet 的網路中）或 NAT。
  - 對內的網路介面的 IP 設定如下：
    * LAN (untagged)：建議仿照 Lab 使用 VirtualBox 的 Host-only Network，IP 可以任意設定，讓你的電腦可以直接連線到 VM 即可。
    * VLAN 11 (tagged) IP: 10.30.11.1/24，需開啟 DHCP server 服務在這個介面上，分配 IP 給客戶端。所分配的 IP 需要在上述的 10.30.11.1/24 的 subnet 內。
- 在你的電腦上安裝一台 Alpine 虛擬機（以下稱為 client）。這台虛擬機只有一個網路介面。
  - 唯一的網路介面與 OPNsense 虛擬機的**對內網路介面直接相接**。
  - 請在 VM 端做好 VLAN tag 的設定（不可使用 hypervisor 端的 VLAN tag 功能），使得它可以存取到 OPNsense 上設定好的 VLAN 11 網路介面。
  - Client 透過 OPNsense 的 DHCP server 自動取得相應 10.30.11.0/24 subnet 中的 IP 位址。
  - 請先安裝好 sshd 等服務（可以參考 Lab 作法）。

Tasks

1. (★) **Firewall Lab 好困難**
   (a) (2 points) 請在 VLAN 11 下那台 Alpine VM 上執行 `ip a; ip r` 指令驗證設定是否正確。此 VM 在 VLAN 11 的介面應從 OPNsense 的 DHCP server 取得 IPv4 位置，並從此 IPv4 位置對外存取 Internet。
   (b) (2 points) VLAN 11 下的機器只能連線到防火牆本身、ws[1-6].csie.org 和任意的 DNS 服務 (TCP/UDP port 53)。
   (c) (2 points) 在 2025/4/8 這天，因為 FHVirus 要使用 client 連到一些色色的網站，因此希望那天防火牆允許 VLAN 11 到 這個網址 裡列出的網址對應的 IP 位址。除了 2025/4/8 之外的時間都不可以連到這些網站。請使用 alias 中 URL Table (IPs) 和 schedule 功能完成本題，這使得 OPNsense 會定期自行下載這個列表，而不是手動一個一個網址輸入。助教檢查時可能會請你啟用、解除這條規則，並展示其對於過濾網站造成的差異。

2. (★★★) **OPNsense 也想脫單**

   Note: 本題和第三部份（我不會 VPN 怎麼辦）是獨立的，沒有順序關係，可以分開 demo。

   FHVirus 加入了臺大卷卷義大利麵社（NTU Curly Pasta Club，簡稱 NTUCPC），並擔任社團唯一的網管。隻身負責社團網管事務的 FHVirus，只能孤單的解決各式各樣的問題……（故事詳見 HW0）

   FHVirus：「雖然我只有一個人，但我的防火牆不可以承受這種孤獨！」

   於是 FHVirus 想設定備援防火牆，但他最近在讀期中考有點忙[1]，所以他從上古的 NASA 三階挖出了文件，請你照著文件完成備援防火牆的設定吧！文件連結：https://hackmd.io/@FHVirus/SJIalZdp1g

   (a) (5 points) 兩臺防火牆在 WAN、VLAN 11 兩個子網路中，都使用 CARP 共享一個 Virtual IP (VIP) (每個子網路一個，共兩個)。設定 DHCP server 發給 client IP 時，以兩臺防火牆在 VLAN 11 中共享的 VIP 做為 default gateway，使得任意一臺防火牆停機時，client 都可以存取 Internet。

   (b) (2 points) 可以在 Master 設定 rules 並同步到 Backup 防火牆。

3. (5 points) (★★) **我不會 VPN 怎麼辦**

   Note: 本題和第二部份（OPNsense 也想脫單）是獨立的，沒有順序關係，可以分開 demo。

   架好防火牆後，FHVirus 現在需要一個能隨時隨地連線進內網的方法。但是架 VPN server 實在太麻煩[2]了！現在請你：

   (a) 從 Alpine 虛擬機建立一個 SSH tunnel，使得連線到這台機器 VLAN 11 介面的 Port 4567 時，會連線到 OPNsense 的 WebGUI 管理介面。

   (b) 請在 OPNsense 上面設定一個 port forwarding 機制，使得從外部可以連線到 OPNsense 的 WAN IP address 的 Port 4567，此時會連線到 alpine 虛擬機的 Port 4567，因此可以存取到 OPNsense 的 WebGUI 管理介面。[3]

   這樣就可以從 WAN 連到 OPNsense 而不需要設定 allow all 這種爛 rule 了！

   請 demo 從任何主機透過 WAN 的介面連線到 OPNsense 的 WebGUI，並展示相對應的 NAT 設定。

---

5 Cisco Switch ★ ∼ ★★ (13 points)

Resources

- midterm.pka
  (SHA256 checksum: 94d46b2cbebf7bbdf198fde64f8e0c19d41745f9ab70fb5a8b0eae6558124028)
- Packet Tracer 8.2.0 安裝檔
- Packet Tracer account: cisco.packet.tracer@yopmail.com
- Packet Tracer password: Cisco.packet.tracer0

[1] 而且他才不管你忙不忙
[2] 其實很簡單，讓我掰一下故事嘛。
[3] 注意，一般而言，這樣設定是會創造出安全漏洞的！因此 FHVirus 是個不合格的網路管理者！

Tasks

1. (5 points) (★) **基本設定**

   請依照以下敘述完成 Switch 基本設定。

   - PC0 和 PC2 屬於 VLAN5
   - PC1 和 PC3 屬於 VLAN8
   - NTP Server 屬於 VLAN99
   - Switch0 和 Switch1 之間的兩條線要形成 Port Channel 1，且 PC0 和 PC2，PC1 和 PC3 要可以透過 Switch 間的 Port Channel 上的 VLAN 設置，使得他們可以互通。
   - 設定 Switch0 在 Interface VLAN99 上的 IP 為 10.99.0.1/8
   - 設定 Switch1 在 Interface VLAN99 上的 IP 為 10.99.0.2/8
   - Switch 之間以及 Switch 和 NTP Server 之間要可以透過 VLAN 99 來互通。這也表示 Port Channel 1 上面要有對應的 VLAN 設定。

2. (3 points) (★) **知道現在是幾點嗎？**

   NTP（Network Time Protocol）是一個用來同步網路設備之間的時間的協議，它對於維持系統的正確性和協調性非常重要。尤其是在進行 debug 時，準確的時間能幫助確定事件發生的順序，分析造成問題的原因。

   在本題中，請利用 NTP Server 設定 Switch 0 以及 Switch 1 的時間。由於 NTP 設定需要一段時間才會生效，你可以點擊左下角的快轉按鈕來快進時間。

3. (5 points) (★★) **不可以用自己的電腦哦！**

   在現實中，公司通常要求員工使用公司提供的設備進行工作，以保護公司重要資料不外流。然而，如果員工偷偷將公司的網路線接到自己的設備上，則可能造成資料洩漏的風險。

   在本題中，你需要完成以下設定：

   - 當 Cable 1 連接到 PC4 時，PC4 無法 ping 到任何一台設備
   - PC4 還沒有設置 IP 設定。請幫它設一個可以跟 PC1、PC3 互通的 IP 設定。
   - 但當 Cable 1 接回 PC3 時，PC3 能夠重新 ping 到 PC1
   - 評分過程中，只能點擊左下角的快轉按鈕來快進時間，不能中途修改其他設定
   - 點擊網路線上的綠色三角形，可以拖動該線並連接到另一個網路設備

註：本次考試不提供 Assessment Items 以及 Connectivity Tests，完成後請直接找助教評分，謝謝。

Submission

- 請呼叫助教協助 demo，可一次 demo 多題。

---

6 Nginx ★ ∼ ★★ (13 points)

Before you start

- 請利用 debian 12.9.0 netinst ISO 安裝虛擬機完成本題，雲端上提供的檔案有：
  - `p6-arm64`
    - `debian-12.9.0-arm64-netinst.iso`
  - `p6-x86_64`
    - `debian-12.9.0-amd64-netinst.iso`

Tasks

1. (1 points) (★) **Basic Setups**
   - 請確認虛擬化的網路設定確保你的本機能夠連得到 VM。
   - 請安裝 Nginx Server 所需的相關套件，並啟動 Nginx Service。
   - 開啟 ufw 防火牆:
     - 允許 port 22 及 port 80 的連入連線
     - 拒絕所有其他連入的連線
     - 允許所有對外的連線
   - 請附上:
     - 連線至 `http://{host}:{port}` 的瀏覽器畫面截圖，應該要會出現顯示 Welcome to nginx!
     - 執行 ufw status verbose 的結果畫面截圖。
     - 助教可能會在部分狀況下，請同學額外增加網頁上的內容，並加以檢驗。

2. (1 points) (★) **Access Log**
   - 請將 Nginx Server 的 access log 建立並保存在以下的路徑中：/var/log/nginx/nasabook.log。
   - 請將 log 格式改為: “{remote address} - {request} - {status} - {local time}”。下面是一個範例的格式：`{192.168.2.11} - {GET / HTTP/1.1} - {304} - {06/Apr/2025:15:35:29 +0800}`
   - 請附上執行 cat /var/log/nginx/nasabook.log 的畫面截圖。如果 log 是空白的請多連線你的網頁幾次。

3. (1 points) (★) **404 Not Found**
   - 請設定 Nginx，當伺服器無法找到請求的頁面時，使用下圖的貓貓來回覆 404 Error response (可參見 `https://http.cat/404`，可取得貓貓圖的檔案)，並回傳 404 status code。
   - 請附上:
     - 連線至 `http://{host}:{port}/404_not_found` 的瀏覽器畫面截圖，網頁無需排版，有出現貓貓即可。
     - 在本機執行 curl -I `http://{host}:{port}/404_not_found` 的畫面截圖。
     - 上述 `http://{host}:{port}/404_not_found` 並不是真實存在的網址，而是用來檢驗在你的 Nginx 伺服器出現不存在網址時應有的行為。
     - 助教可能會在部分狀況下，請同學額外增加網頁上的內容，並加以檢驗。

4. (3 points) (★★) **PHP on Server**

   - 請在本機（可以操作瀏覽器的一台機器，或許是你的筆電，或者是 204 的桌機電腦）設定 hostname 跟 IP 的配對檔案（Linux、MacOS 中的`/etc/hosts`、或 Windows 中的 `C:\Windows\System32\drivers\etc\hosts`），使得 my-website.ntu.edu.tw 這個 domain name 指到你的 VM 的 IP 位址。(記得考試結束要改回來喔！)
   - 請將這一個用 php 計算圓周率的小程式，改成可以在`http://my-website.ntu.edu.tw:{port}/do.php`看到執行結果。在 do.php 檔案的最上面有 shebang 可能要去除掉。
   - 請附上:
     - 連線至 `http://my-website.ntu.edu.tw:{port}/do.php` 的瀏覽器畫面截圖。
     - 助教可能會在部分狀況下，請同學額外增加網頁上的內容，並加以檢驗。

5. (3 points) (★★) **Reverse Proxy & User Directory**

   作業已經讓你有能力在同一個 Nginx Server 上面設定多個網站，用不同的網域名稱連上時會有不同的內容，我們再來複習一次吧 (前一小題尚未完成者，此題亦不計分)

   (a) 請用前一小題的做法，讓 csie-website.ntu.edu.tw 這個 domain name 也指到你的 VM，當從本機瀏覽 `http://csie-website.ntu.edu.tw:{port}` 可以看到「國立台灣大學資訊工程學系」的網站。(參見 `https://csie.ntu.edu.tw/`) 請附上:
   - 連線至 `http://csie-website.ntu.edu.tw:{port}` 的瀏覽器畫面截圖。

   (b) 請新增一位使用者 nasacat，並在 /home/nasacat/htdocs 下建立 midterm.php 和 midterm.html。

   (c) 請在前一小題的結果存在的情況下，當從本機連上 `http://csie-website.ntu.edu.tw:{port}/~nasacat/midterm.php`（注意是 php 檔案）即可顯示“Medium Midterm”字樣。當從本機連上 `http://csie-website.ntu.edu.tw:{port}/~nasacat/midterm.html`（注意是 html 檔案）則可顯示“Hard Midterm”字樣，html 和 php 內容能顯示字樣即可。請附上:
   - 連線至 `http://csie-website.ntu.edu.tw:{port}/~nasacat/midterm.php` 的瀏覽器畫面截圖。(請確認 PHP 有被成功執行，而不是被 Server 直接返回檔案)
   - 連線至 `http://csie-website.ntu.edu.tw:{port}/~nasacat/midterm.html` 的瀏覽器畫面截圖。

6. (4 points) (★★) **N Web Servers?!**

   你知道同一台機器上可以同時存在 N 個 web servers 嗎？在實務中，不同的 web server 開發上會有不同的考量面向，例如 Nginx 強大在速度、Apache 強大在功能的齊全，因此有些人會用

Nginx 接收 request，簡單的網站就直接回傳靜態網頁、複雜的則轉交給 Apache 處理；甚至不只功能上的考量，在「同一台」實體機上，因為網頁的開發語言不同，我們可能要讓 Nginx 把 request 轉傳給另一個 server 處理，比方說這個網站的後端是用 JavaScript 開發的，可能就要轉給 Nodejs server，因此這題要來讓你嘗試簡單地建置雙 server！

- 請你另外安裝 Apache，將它開在 8080 port，並且讓 Nginx 在看到 `http://{host}:{port}/apache` 時導向 Apache server，它應該要能顯示 Apache 預設的歡迎畫面。
- 請附上:
  - 連線至 `http://{host}:{port}/apache` 的瀏覽器畫面截圖。
- Hint: 如果你連上去出現的是 Apache 顯示的 404 畫面，代表你超級接近了，也許可以再多想想 Nginx redirect 時多加什麼資訊可以讓 Apache 知道你要的是 root index page (/)？

Submission

- 請呼叫 TA 並進行 demo。