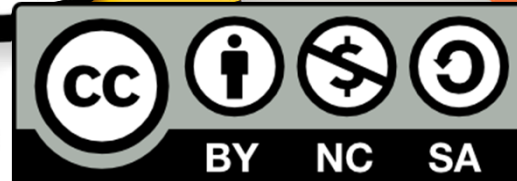


中正資工, 2021, 系統程式設計基於Linux

授課老師：羅習五 副教授



為什麼需要學習「系統程式設計」

- 🍏 作業系統將「硬體」隱藏起來，提供程式設計介面（API）供我們間接操作硬體
 - ♣️ 硬碟 => 檔案 + 目錄
 - ♣️ CPU => 行程 + 執行緒（或統稱為task）
- 🍏 作業系統提供「虛擬」的功能
 - ♣️ 行程間的通訊，網路、copy-paste
 - ♣️ 每個行程完整的「256TB」虛擬記憶體
- 🍏 提供了更方便的程式設計介面
 - ♣️ 多核心程式，如何存取共用資料
 - ♣️ 具有I/O優化的函數庫

關於「抽象化」

- 🍏 SSD或HDD中的資料以4KB為單位 (block) , 從0開始編號 (LBA)
 - ♣️ 將一連串的block構成「有序列」, 形成檔案
 - ♣️ 檔案只是一個編號
 - ♣️ 某些檔案由作業系統給予特別的解釋, 例如: 目錄, 其實是一個檔案, 裡面記載「這個目錄」具有哪些檔案, 檔案的名稱 (檔名)
 - ♣️ 賦予「權限」。例如: 我無法讀取別人的home。RWX RWX RWX
- 🍏 某些系統資訊以「檔案的形式」呈現 (pseudo file system)
 - ♣️ 例如: /proc

關於「抽象化」

- 🍏 CPU一次只能執行「一連串指令」
- 🍏 將「一連串指令」抽象化成「執行檔」，使用者可以自由的選擇要執行的檔案
- 🍏 作業系統可以將「執行檔」放入「記憶體」，成為「應用程式」
- 🍏 作業系統再將CPU抽象化成多個，因此可以同時執行多個「應用程式」
- 🍏 我們可能需要在「應用程式」間，做複製、貼上
- 🍏 可能需要設定優先權並且可以「刪除」發生錯誤的程式

關於「抽象化」

- 🍏 以printf為例
- 🍏 透過libc，將「字串格式化」
- 🍏 如果遇到「\n」（換行符號），將字串輸出到「標準輸出」
 - ♣️ 命列作業系統：write, 1, "我是羅習五", 11（總共「4」個參數）
- 🍏 標準輸出可能是（第二個參數「1」的「實體」是啥？）
 - ♣️ 螢幕
 - ♣️ 網路（例如：透過telnet、ssh等）
 - ♣️ 特殊的傳輸線材（例如：RS232、UART、Serial port），這種接頭主機板上一定會有，但對大部分的人而言使用USB更方便

關於系統呼叫 (system call) 的編號

🍏 <https://filippo.io/linux-syscall-table/>

🍏 「write」的編號是「1」

🍏 因此呼叫Linux kernel的參數是

♣️ 1 (代表write) , 1 (代表標準輸出) , 0x?????? (字串的位址) , 11 (字串的長度)

關於「抽象化」

- 🍏 因此「作業系統」不單單只是讓我們更容易使用「硬體」
- 🍏 作業系統也在硬體上開創出新的「使用方式」、「使用限制」
- 🍏 透過這門課，可以了解如何「控制」作業系統，也可以了解作業系統的功能

「系統程式設計」的介面標準

- 🍏 IEEE POSIX
 - ♣️ Portable Operating System Interface
 - ♣️ 目標：讓所有的作業系統有一致的介面設計
 - ♣️ 不只程式設計介面，也包含了基本的軟體：如：壓縮軟體、命令列（shell）
- 🍏 SUS
 - ♣️ Single UNIX Specification（SUS），擴充了POSIX的規範，更加的UNIX
 - ♣️ 最新版為SUSV4-2018
- 🍏 GNU/OS
 - ♣️ “GNU 是個自由軟體作業系統——也就是說，它尊重使用者的自由。”
 - ♣️ GNU作業系統的核心並沒有很成功，大部分都使用了Linux做為作業系統的核心（kernel）。因此Linux的明確名稱是：GNU/Linux

為什麼選擇UNIX介紹系統程式設計

🍏 他是更開放的標準

🍏 更重要的是：他很成功

🍀 以手機、平板電腦來說：市佔率幾乎是100%。Apple使用BSD。Google使用Linux

🍀 伺服器：幾乎全部是UNIX（在某些場合，幾乎是Linux）

🍀 桌上型電腦：16.3%是MacOS（BSD），其餘幾乎為Windows

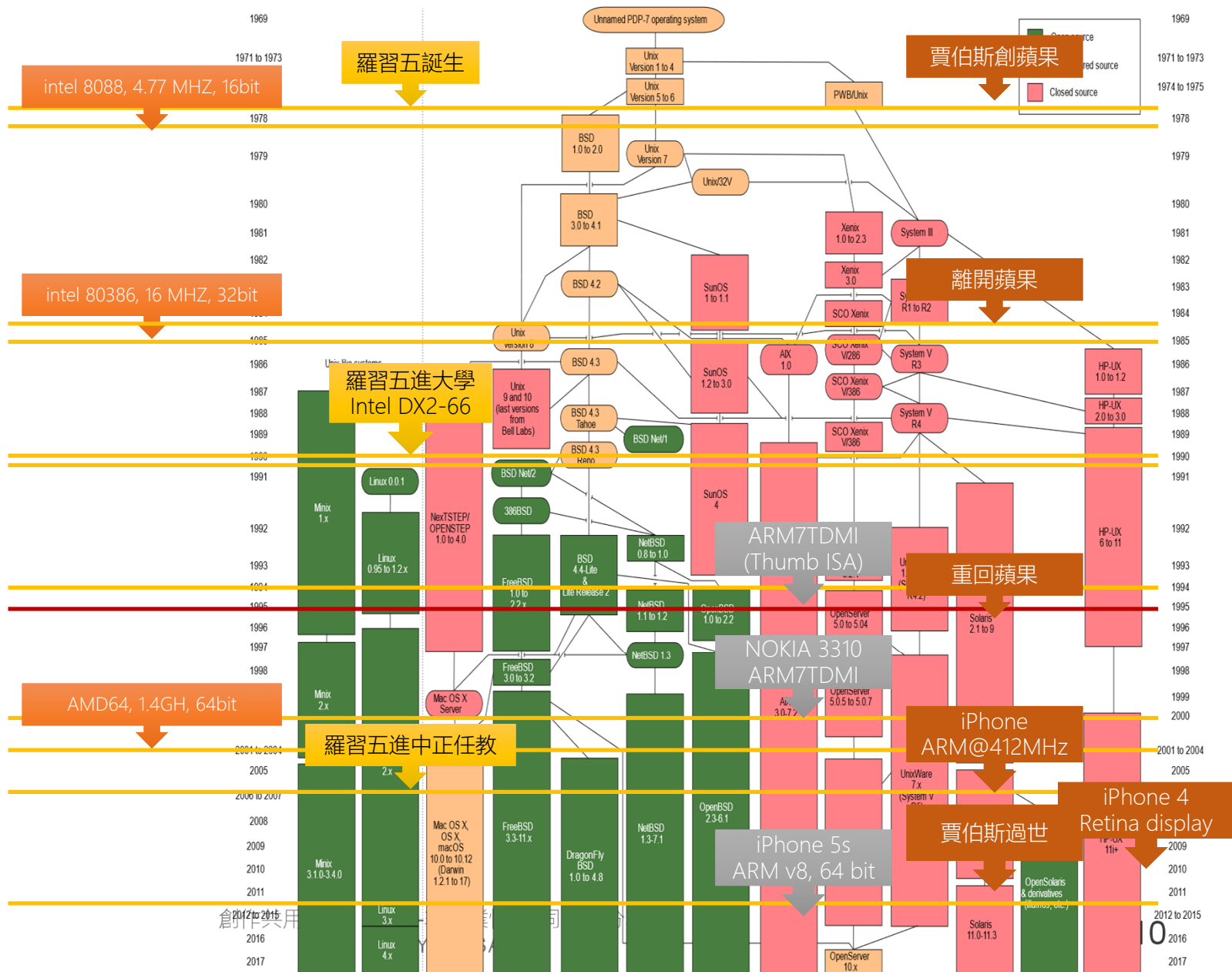
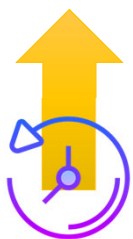
🍏 Windows的設計許多都借鏡UNIX

🍀 使用google查詢Windows的設計介面「“UNIX的keyword” + Windows」

🍀 Windows 10具有三個子系統：Win32、OS/2、Linux，即Windows 10上可以直接執行Linux的執行檔

<http://gs.statcounter.com/os-market-share>

1960之前誕生了
virtual memory、
i-node、hard
disk等技術

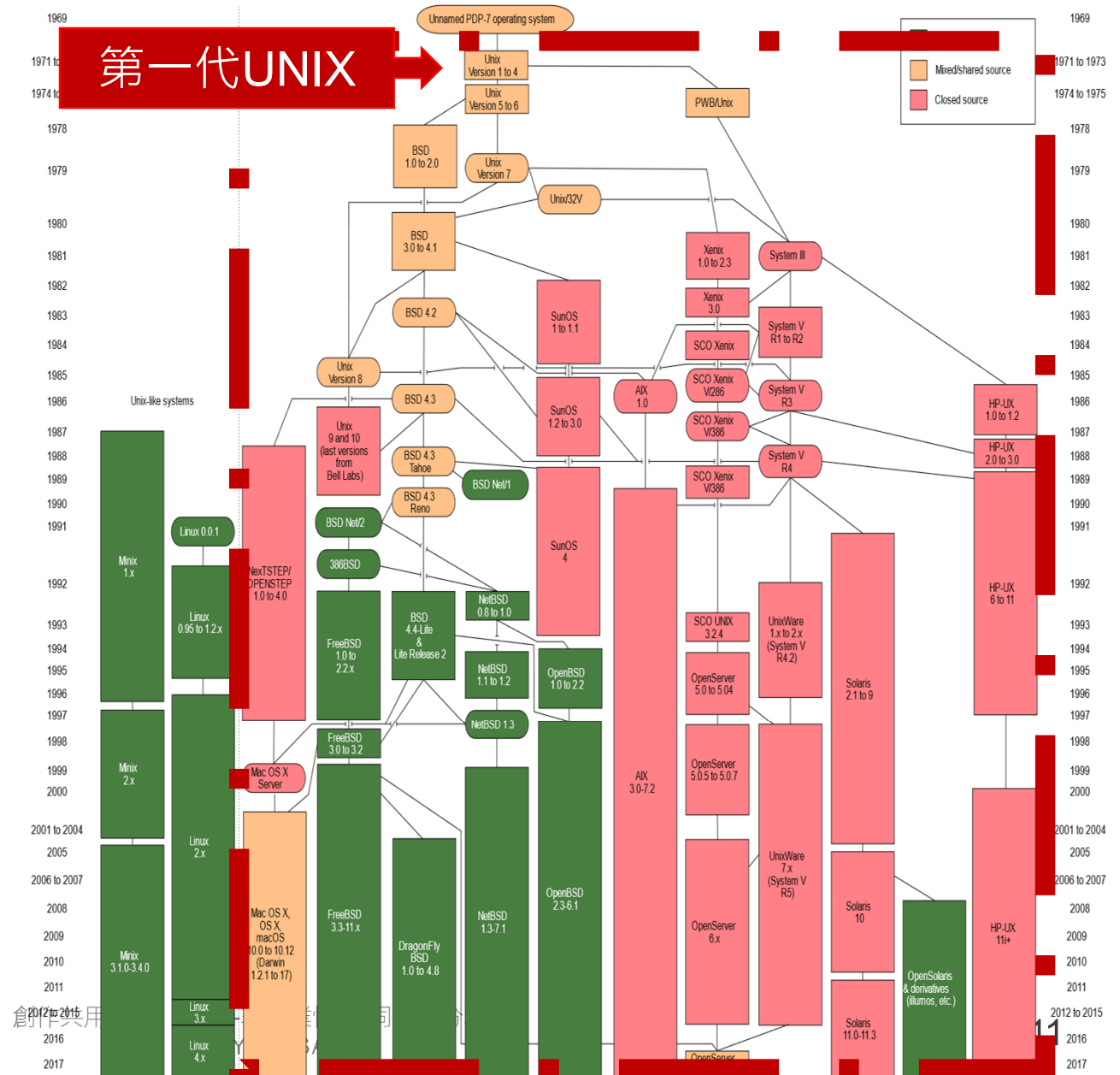


UNIX

🍏 繼承了第一代UNIX
的程式碼

🍏 BSD (Mac) 、
Solaris、 AIX
(IBM) 、 HPUX
(HP)

中正大學－羅習五

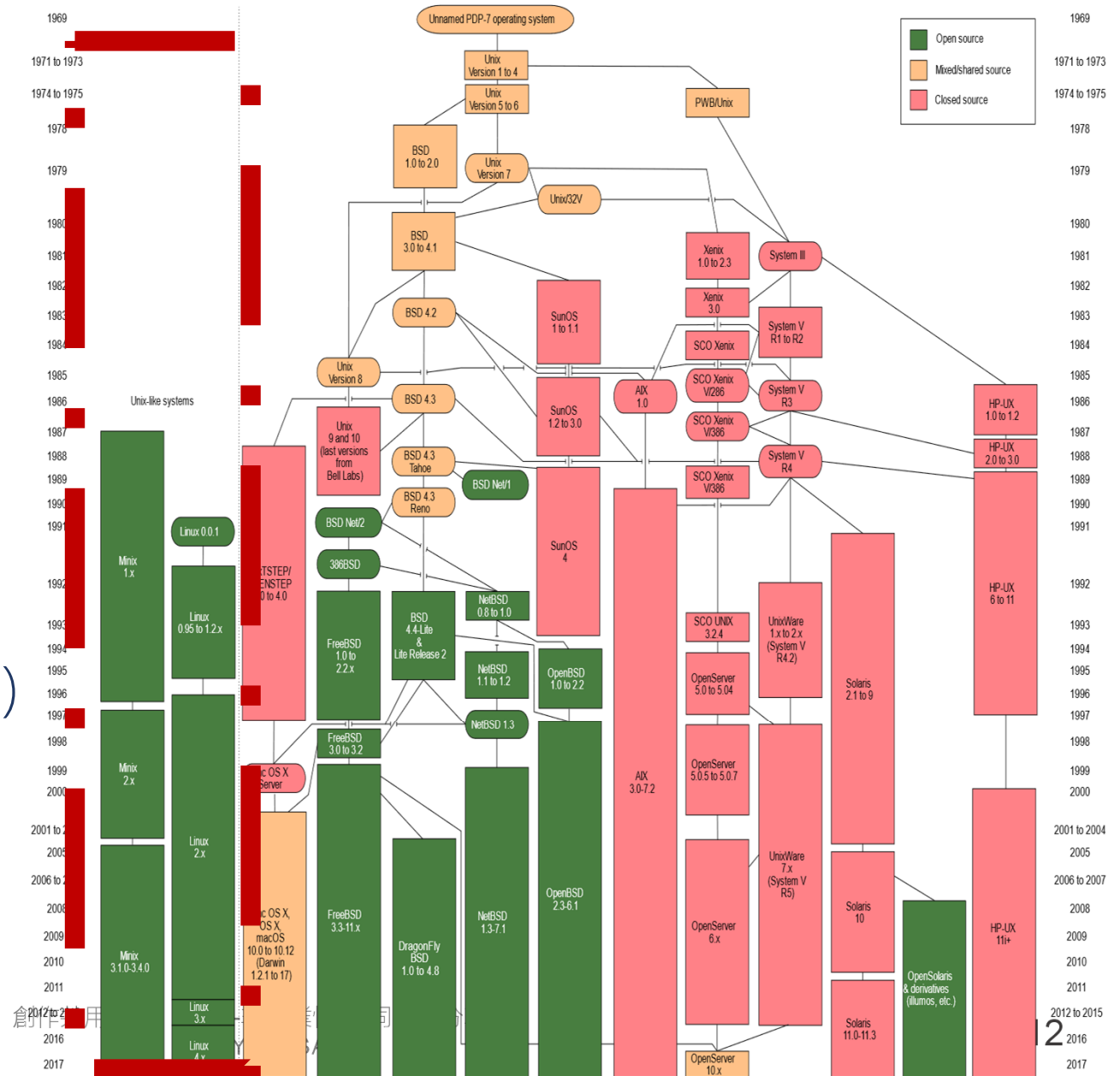


UNIX-like

🍏 並未繼承程式碼，
但大幅度的相容於
UNIX

🍏 Minix (啟發了Linux)

🍏 Linux



為什麼這門課介紹Linux

- 🍏 為了讓同學更加了解「核心精神」，我們不介紹如何寫出「跨平台」的UNIX程式
 - ♣️ 雖然同為UNIX，但BSD和Solaris會有大致一樣，但細節不太一樣的行為
 - ♣️ 同一份規範，不同的實作依然有差異性。每個廠商的解讀不完全相同
 - ♣️ 例如：signal，雖然在POSIX和SUS內增加了規範，但喪失了「簡易性」
- 🍏 GNU/Linux在市佔率上大幅度的領先
 - ♣️ 限縮於GNU/Linux，我們可以使用一些還未進入規範的功能
 - ♣️ 例如：file hole



專屬於這門課的部分

這門課的進化

- 🍏 第一代，於階梯教室上課
 - ♣️ 教科書：Advanced Programming in the UNIX® Environment (APUE)
 - ♣️ 評分方式：作業 + 點名 + 期中、期末考
 - ♣️ 缺點：用投影片講解程式碼，不容易具體了解
- 🍏 第二代，改到電腦教室上課
 - ♣️ 評分方式：作業 + 點名 + 期中、期末考 + 上機考
 - ♣️ 缺點：教科書十分強調「跨平台」，造成初學者理解困難
- 🍏 第三代，改用自編教材
 - ♣️ 於電腦教室上課，「並且」上課中直接修改、執行
 - ♣️ 缺點：上課時間往往不夠用
- 🍏 第四代，自編教材 + youtube
 - ♣️ 可能的缺點：同學們並沒有先看過youtube
 - ♣️ 可能需要在這堂課上花更多時間
- 🍏 第五代，加強程式碼的「可讀性」，增加更有趣的作業

這學期...

- 🍎 「授課老師」的穿著很隨性
 - ✿ T shirt + 短褲 + 布鞋 + 亂亂的頭髮
 - ✿ 授課老師只有電腦很乾淨、體面
- 🍎 上課時以實際操作為主
 - ✿ 使用zoom.us，不使用廣播系統，同學們可以自由的使用電腦
 - ✿ 「原則上」上課不講解投影片，主要是回答問題，幫同學解決程式問題
 - ✿ 同學們應該在家先看過youtube
- 🍎 上課如需「滑手機」、「睡覺」、打瞌睡，請選擇教室後面的位置
- 🍎 如果有無法控制的理由，如：焦慮、過動、AS，請寫信告知授課老師。如果同學影響到其他人或者授課老師，將被請出教室

這學期...

🍏 每星期都要到『教室上課』或『網路上課』！上課點名

🍀 教室上課，授課老師可以幫同學互動

🍀 網路上課，同學們的發問情況比較好

🍏 請在上課前將當週的youtube看完

🍏 上課的方式可能因著同學們的學習而有所改變

🍏 如果沒有特別問題可能會提早下課

上課教材

- 🍏 課程網站: <https://www.cs.ccu.edu.tw/~shiwulo/course/2021-sp/>
- 🍏 論壇: https://groups.google.com/d/forum/ccu_system_programming
- 🍏 課程投影片: <https://goo.gl/X7mUPx>
- 🍏 youtube: <https://goo.gl/NnbLWM>
- 🍏 程式碼: <https://github.com/shiwulo/system-programming>
- 🍏 上課視訊: <https://zoom.us/j/6456450000>

🍀 zoom.us也有APP, 你們也可以用手機看直播, 請在應用程式商店搜尋「zoom」

評分方式（暫定，視疫情等）

- 🍎 點名成績（10%）
- 🍎 每週的小考（30%，以作業的形式繳交）
 - ♣️ 小考公布後七天內繳交（例如：3/3號公布作業，3/10日晚上11:59前繳交）
 - ♣️ 繳交後助教會在**二天內立即公布所有人作業**，作業必須能在Ubuntu 20.04上使用gcc、make編譯（程式作業）。文件可以用pdf瀏覽器觀看
 - ♣️ 在繳交作業後『原則上』二個星期內公布成績
 - ♣️ 如果發現其他同學疑似抄襲，可以寫信告知授課老師，直接由學校處理
- 🍎 期中考、期末考（60%）
 - ♣️ 考試包含筆試（有考古題）及上機考，分別各佔本項成績的40%及60%
 - ♣️ 筆試時間3~4小時。上機考6小時

有二個開發平台

- 🍏 可以使用virtual machine或者雙系統開機，安裝Ubuntu 20.04
 - ♣️ 較為方便
 - ♣️ 可以使用各種開發環境（各種整合式開發介面）
 - ♣️ 缺點：硬體需求較高，建議4核心以上，記憶體8GB以上
- 🍏 高階伺服器
 - ♣️ 實體主機為：Intel Xeon 48 core, 320GB DRAM, 200G HDD（預計選課結束後二星期後開放）
 - ♣️ 帳號：s學號，密碼：email。例如：687410007，那麼帳號是：s687410007
 - ♣️ 缺點：伺服器可能當機。太多同學同時使用，效能可能會較低

web版的伺服器平台



<http://103.124.72.224:7777/>



沒有帳號和密碼，大家可以先試用看看

```
home > guest1 > system-programming > ch04 > flock.c > main(int, char * [])
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <sys/file.h>
4
5  int main(int argc, char* argv[]) {
6      int fd;
7      int ret;
8      char opt;
9      fd = open (argv [1], O_WRONLY);
10     printf("fd = %d is opened\n", fd);
11
12     sscanf(argv[2], "%c", &opt);
13     switch (opt) {
14         case 's':
15             ret = flock(fd, LOCK_SH);
16             break;
17         case 'e':
18             ret = flock(fd, LOCK_EX);
19             break;
```

DEBUG CONSOLE

Loaded '/lib/x86_64-linux-gnu/libc.so.6'. Symbols loaded.

Breakpoint 2, main (argc=1, argv=0x7fffffffde58) at /home/guest1/system-programming/ch04/flock.c:9

9 fd = open (argv [1], O_WRONLY);

Execute debugger commands using "-exec <command>", for example "-exec info registers" will list registers in use (when GDB is the debugger)

關於加簽

- 🍏 請繳交給助教
- 🍏 授課老師依照底下原則
 - ♣️ 為鼓勵女生讀資工系，女生優先
 - ♣️ 資工系二年級優先
 - ♣️ 資工系優先
 - ♣️ 其他
 - ♣️ 也可以寫上20個字，一定要選這門課的理由（例如：因為 X X 原因，無法修另一門 S P）

關於通過率

- 🍏 依照同學的成績，直接送出成績
- 🍏 前學年SP通過略約為80%，平均分約70分
- 🍏 上學期OS通過率約為92%，平均分約80分
- 🍏 通過率每年都會不同

本週進度

🍏 熟悉room.us

🍏 安裝Linux

🍀 <https://tinyurl.com/u1v7jhvn>

🍀 不用繳交

請所有人學習zoom.us

- 🍏 發問（如果可以準確描述問題，用此功能）：
 - 🍀 使用發問的功能，將你的學號「發表你的學號：# # # #」
- 🍏 使用箭頭和繪圖（可以指出問題）
 - 🍀 使用箭頭和繪圖，注意：必須使用半透明的顏色
 - 🍀 第一排同學先畫，第二排，然後。。。
- 🍏 螢幕分享：
 - 🍀 請先使用任何軟體，在螢幕上顯示你的學號
 - 🍀 然後使用螢幕分享功能