LKM
Assignment

授課老師：廖峻鋒
TA：莊崴宇、姚惠馨

作業附檔說明

- PG1作業說明.pdf　　// 作業引導與繳交格式之說明文件
- src資料夾　　// 進入虛擬機的ubuntu後，透過git clone指令下載
- src/simple.c , src/seconds.c , src/Makefile　// 作業中需修改的檔案
- src/hello.c　　// 範例檔

作業說明

- 這次作業中會更改範例的核心模組(kernel module) 以及載入核心模組到Linux核心運作

- 步驟：

  1. clone github上的作業檔案
  2. 完成作業後截圖，貼在從moodle下載的作業單"template.docx" word檔裡
     - 記得將 word檔 改名為 “ 學號.docx ”
  3. 按照作業繳交格式繳交

前置作業

- 於本機端打開命令列視窗，切到OS_2021目錄(lab0創建的目錄)下，執行以下指令:

  1. **vagrant up** //開啟虛擬機

  2. **vagrant ssh** //登入虛擬機

- 於ubuntu的終端機執行以下指令
  - **git clone https://github.com/C-WeiYu/src.git** //取得作業程式碼

相關指令

- ls　　//查看當前資料夾內容
- la　　//查看當前資料夾內容與隱藏內容
- cd <檔案名稱>　　//進入到目標資料夾
- cd ..　　//返回上一層資料夾
- vim <檔案名稱>　　//編輯檔案 -> 案 i 進入編輯模式、esc離開編輯模式、:q 不儲存離開、:wq 儲存離開
- nano <檔案名稱>　　//編輯檔案 -> Ctrl + X 離開
- cat <檔案名稱>　　//將文件內容印在終端機上
- top　　//顯示目前電腦狀況
- insmod <模組名稱>　　//載入核心模組
- rmmod <模組名稱>　　//移除核心模組
- lsmod　　//查看目前系統中有哪些模組(練習時可以查看模組有沒有成功載入)

練習一、載入及移除核心模組(kernel module)

練習一、載入及移除核心模組(kernel module)

✓ **作業運行方法及說明：**

- 在git clone下來的檔案裡，simple.c 為一個基本的核心模組 (kernel module)
- simple.c 可以在載入及移除時留下相對應的訊息在 kernel 內部的 ring buffer
  - kernel偶爾會產生一些有助於診斷問題的訊息，像是 I/O發生問題、USB 裝置熱插拔時 等等。這些訊息都會被寫入 kernel 內部的 ring buffer (由於 buffer 的大小固定，所以舊的 "訊息"就會被新的"訊息"蓋掉)
  - **dmesg** 可以查看目前 kernel ring buffer 的指令，**sudo dmesg -c** 可以清理這些訊息。
- 輸入 **make** 可以透過 Makefile 編譯產生多個檔案，而 simple.ko 則為被編譯過的核心模組
- 進行 模組載入 **sudo insmod simple.ko** 後，輸入 **dmesg** 就能看見 "Loading Module."
- 進行 模組移除 **sudo rmmod simple** 後，輸入 **dmesg** 就能看見多出 "Removing Module." (可以省略掉 .ko， 每次載入後要記得移除)

練習一、載入及移除核心模組(kernel module)

✓ **作業目標：**

- 練習修改核心模組，讓它在**載入時紀錄當時的 jiffies 以及 HZ** 訊息於 kernel 內部的 ring buffer，並在模組**移除時留下當時的 jiffies** 訊息在 kernel 內部的 ring buffer。
- 補充：
  - HZ 為 linux核心固定週期發出的 time interrupt，用來定義每秒發生幾次 timer interrupts。
    例如：HZ = 1000 , 代表每秒有 1000 次的 time interrupts。
  - tick 為 HZ 的倒數，即 timer interrupt 每發生一次的時間。
    例如：HZ = 250 , 代表 tick 為 4 毫秒 (1000 / 250 = 4)。
  - jiffies 為 Linux 核心變數 (unsigned long)，用來紀錄系統從開機以來，經過多少的 tick。
    即每發生一次 timer interrupt，jiffies變數會被加一。
  - unsigned long 為32位元的資料型別，數值範圍為 0 ~ 4294967295，由於沒有紀錄正負值，所以可以儲存的數值上限為“long”(-2147483648 ~ 2147483647)值得大約兩倍。在C語言裡用 “%lu” 來表示。
  - kernel無法呼叫 printf() 函式，但可以呼叫 printk() 函式，printk() 負責訊息刻在 kernel 內部的 ring buffer，printk()函式參數中使用了 “KERN_ALERT” ，這部分是 kernel 訊息的日誌級別，總共分成了 8個級別，而“KERN_INFO” 是第七個級別，負責顯示一般信息。

練習一、載入及移除核心模組(kernel module)

✓ 作業運行步驟：

- simple.c 修改完後，輸入 **make** ，透過 Makefile 檔案編譯產生 simple.ko 檔
- 輸入 **sudo insmod simple.ko** 將模組載入後，再輸入 **sudo rmmod simple** 移除
- 輸入 **dmesg** 終端機將顯示:
  - Loding Module
    init_jiffies : XXXXXX
    HZ : XXX
    exit_jiffies : XXXXXX
    Removing Module
- 截圖終端機上的輸出畫面為作業截圖一
- 最後輸入 **make clean** 清除檔案，留下要上傳的.c檔

練習二、/proc 檔案系統

練習二、/proc 檔案系統

✓ 介紹：

Linux 核心提供了一種通過 /proc 檔案系統，在執行時訪問核心 內部資料結構、改變核心設定的機制。 proc檔案系統是一個偽檔案系統，它只存在記憶體當中，而不佔用外存空間。它以檔案系統的方 式為訪問系統核心 資料的操作提供介面。 使用者和應用程式可以通過 proc得到系統的資訊，並可以 改變核心的某些引數。由於 系統的資訊，如程序，是動態改變的，所以使用者或應用程式讀取 proc檔 案時，proc檔案系統是動態從系統 核心讀出所需資訊並提交的。

✓ 範例檔案介紹：

- hello.c 為一個核心模組，能創出/proc/hello檔案，透過 **cat /proc/hello** 能印出檔案內容
- 在模組入口proc_init() 中，使用proc_create()函式創建新的 /proc/hello 項目。此函式透過proc_ops傳遞，其中包含 struct file_operations的reference。該struct將 .owner和.read初始化。.read的值是每次讀取 /proc/hello 時都會呼叫的函數式 proc_read()。 在proc_read()函式中，可以看到字串 “Hello World\n” 被寫入在Buffer，該 Buffer在核心記憶體 (kernel memory)中。因為要可以從 user space 訪問 /proc/hello，所以要使用核心函式 raw_copy_to_user() 將Buffer的內容複製到user space。此函式將核心記憶體緩衝區的內容複製到用戶空間中存在的變數usr_buf。每次讀取 /proc/hello文件時，都會重複呼叫proc_read()函式，直到返回0，因此必須有邏輯確保該函式在收集到數據後返回 0 (在這種情 況下，字串 “Hello World\n”)，該字串將進入相應的 /proc/hello 文件中。

練習二、/proc 檔案系統

✓ 練習：

1. 修改Makefile，第一行修改成 (obj-m += hello.o)
2. 輸入指令 **make** 編譯，透過Makefile編譯檔案產生出hello.ko檔
3. 載入模組 **sudo insmod hello.ko**
4. 輸入 **cat /proc/hello**
5. 終端機將顯示/proc/hello檔案裡的("Hello World")內容
6. 輸入 **sudo rmmod hello** 移除模組
7. 輸入 **make clean** 清除檔案

練習二、/proc 檔案系統

✓ **作業目標：**

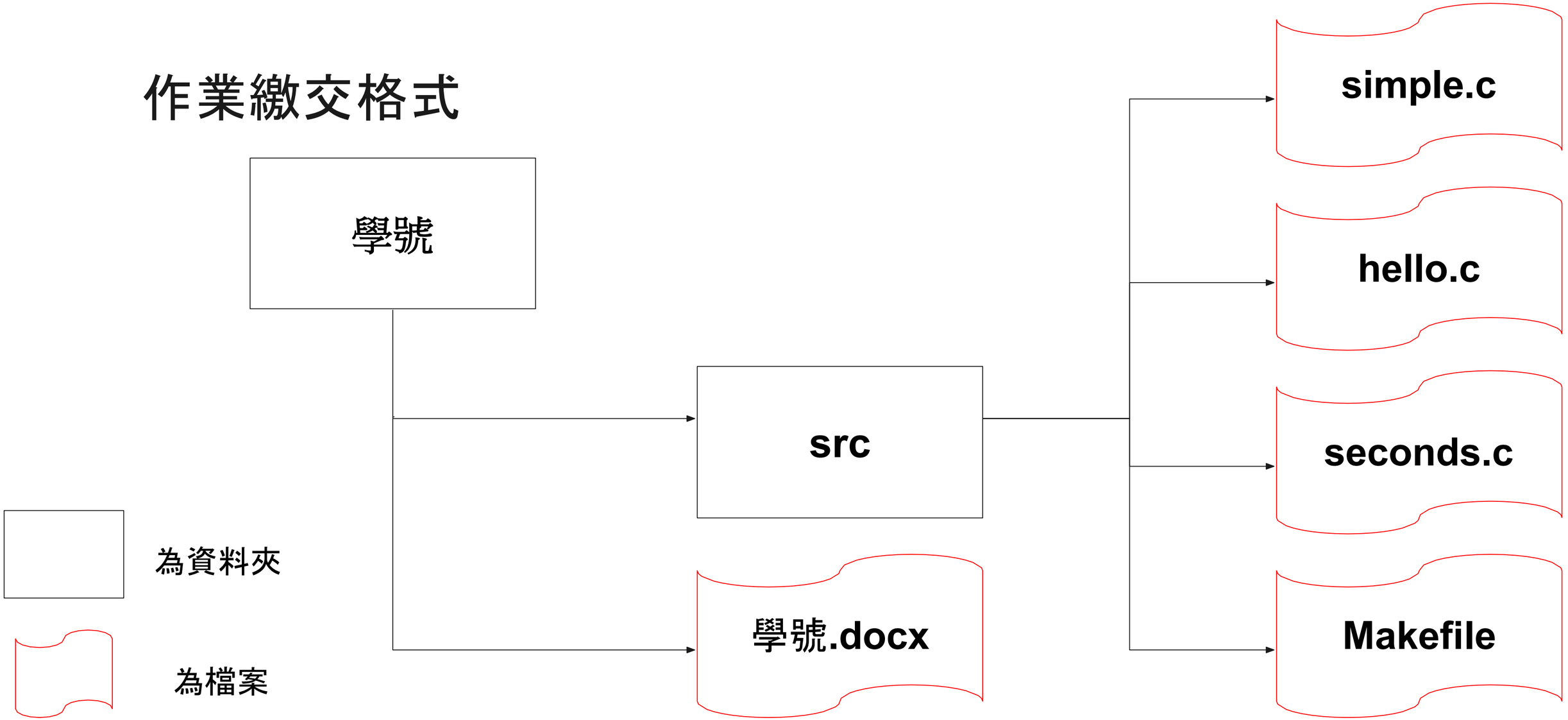
- 練習二的檔案 seconds.c 為 hello.c 的拷貝版，我們要加上一個功能，在終端機輸入 **cat /proc/seconds** 時，顯示此核心模組從載入到輸入指令間，中間所經過的時間秒數。

✓ **作業運行方法：**

- **作法：**因為 jiffies 是不斷變動的時間標記，我們先記錄載入時的數值，與在 /proc/seconds 被讀取時的 jiffies 相減，再除上每秒250次的 HZ，就能得到目前所運行經過的時間。
- 將 Makefile 的第一行修改成 (obj-m += seconds.o) 再使用指令 **make**
- 輸入**sudo insmod seconds.ko** 載入模組
- 輸入 **cat /proc/seconds** 後終端機將顯示
  - Module has been running for XX seconds
- 截圖 輸入五次 cat /proc/seconds 後，在終端機上的輸出為作業截圖二
- 最後 **sudo rmmod seconds** 加上 **make clean**

請將檔案壓縮成.zip檔繳交!!

作業繳交格式



About Programming Assignment

1. 作業評分規則
    a. 有依照作業要求繳交:40分
    b. 程式碼compile無誤:60分
    c. 未按規定繳交作業:一項扣10分
    d. **遲交0分**
2. 程式作業請獨立完成，請勿抄襲同學之程式碼
3. **繳交word作業內容:**
    a. 作業截圖X2
    b. 心得

Thanks for Listening