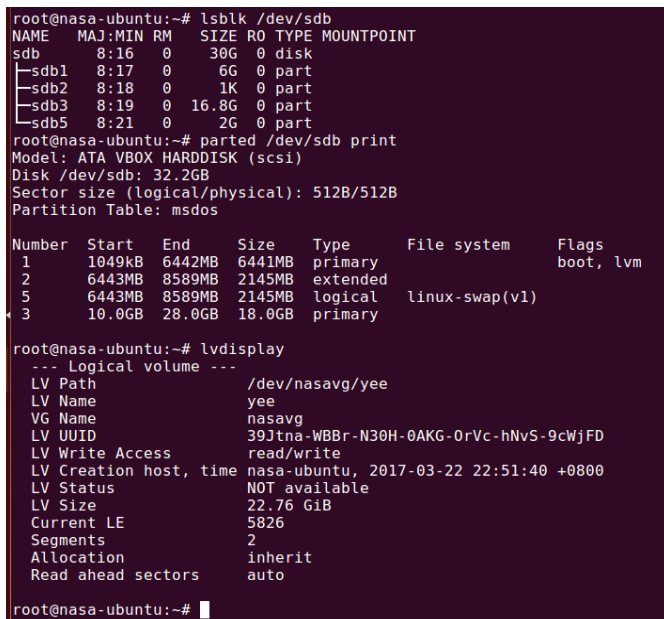
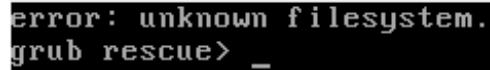


OS Spring 2017, Project 1 Report

Team28 : B04902077 江緯璿 B04902079 甯芝喲

Part 1: Failure

第一次開虛擬機時並未調整硬碟空間大小，僅有預設的 8GB，故中間 rebuild kernel 時很不幸地硬碟空間被用完了，故無法繼續 rebuild，重開機也出現 "Error: could not write bytes: Broken pipe" 錯誤訊息而無法進入桌面，只能強制進入 tty1 做事。於是上網找尋一些方法延伸虛擬磁碟空間(VBoxManage modifyhd)，也有成功，但重開機才發現，延伸出的空間在硬碟最後，且尚未被格式化。於是打算建 LVM 把 sda1(mount on /)跟後面延伸出來的硬碟合併。結果 Ubuntu 沒有內建 pvcreate，需要 apt-get，結果試圖 apt-get 的時候跳出錯誤訊息，磁碟空間滿了故無法 apt-get install lvm2。後來只好把那顆虛擬硬碟插進另一台虛擬機進行磁碟分割。原先以為會成功了，結果不幸的，設定格式化時不小心設定錯了(崩潰)，於是只好認命地重灌一整台虛擬機，然後從頭開始。



```
root@nasa-ubuntu:~# lsblk /dev/sdb
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
sdb   8:16   0   30G  0 disk
├─sdb1 8:17   0    6G  0 part
├─sdb2 8:18   0    1K  0 part
├─sdb3 8:19   0   16.8G  0 part
├─sdb5 8:21   0    2G  0 part
root@nasa-ubuntu:~# parted /dev/sdb print
Model: ATA VBOX HARDDISK (scsi)
Disk /dev/sdb: 32.2GB
Sector size (logical/physical): 512B/512B
Partition Table: msdos

Number  Start   End     Size    Type    File system  Flags
  1      1049kB  6442MB  6441MB  primary
  2      6443MB  8589MB  2145MB  extended
  5      6443MB  8589MB  2145MB  logical  linux-swap(v1)
  3      10.0GB  28.0GB  18.0GB  primary

root@nasa-ubuntu:~# lvdisplay
--- Logical volume ---
LV Path                /dev/nasavg/yee
LV Name                 yee
VG Name                 nasavg
LV UUID                 39Jtna-WBBR-N30H-0AKG-0rVc-hNvS-9cwJFD
LV Write Access         read/write
LV Creation host, time  nasa-ubuntu, 2017-03-22 22:51:40 +0800
LV Status                NOT available
LV Size                 22.76 GiB
Current LE              5826
Segments                2
Allocation              inherit
Read ahead sectors      auto
root@nasa-ubuntu:~#
```

Part 2: Basic Requirements: hello(), show(), multiply() and min()

重新開始之後基本上就沒有遇到什麼太大的問題了，就是按部就班的照著投影片上步驟慢慢來。幸運的是，因為比較晚開始寫作業，在一開始就得知了要在 syscall(340)的部分加上一個 return_zero()的 function，因此沒有受到這部分錯誤的影響。要說真的有遇到什麼問題，大概就是在撰寫 multiply()與 min()時有點不知道該怎麼把參數送進 systemcall 裡面，查了一下，發現是直接加在 syscall 的編號後面就好(ex. syscall(341,10,20))。

Part 3: Bonus: CPU_utilization()

Ref:

<http://blog.csdn.net/jk110333/article/details/8683478>

<http://daydreamer.idv.tw/rewrite.php/read-65.html>

<https://www.sllab.cd.github.io/blog/2012/11/13/how-to-make-linux-module/>

Also discussed with 陳佳佑、楊舒瑄

實作方法：

參考 Reference，利用 `filp_open("/proc/stat", O_RDONLY, 0)` 讀取檔案，再由第一行 `cpu` 的資訊讀取 `cpu` 的總時間與 `idle` 時間。進行兩次取樣，中間利用 `msleep(2000)` 間隔兩秒，並由此計算 CPU 使用率並輸出至 `kernel message`。其中為避免浮點數運算，計算時全部乘 `100000` 並採用 `int`，輸出時再換算成小數點百分比並四捨五入至小數點第二位

因這個函式比較複雜，需要多次測試，為避免一直在 `rebuild kernel`，故先將此 `function` 包成 `kernel module`，確定正常後才寫進 `kernel code` 中。困難點在於 `kernel mode` 中，`api` 都不太一樣，平常很直覺想到的 `api`，都不太能在這次自訂的 `system call` 中使用(ex: `open()`)，另一部分則是因為原先為了 `code` 簡潔與參考，採用了許多自訂 `function`，但始終不確定那些 `funciton` 是否需要寫進其他 `header` 檔中，最後為了保險，還是先把他們全部寫進同一個函式中了。

```
root@nasa-ubuntu:/usr/src/linux-2.6.32.60/kernel/CPU_Utilization# rmmod CPU_Utilization
root@nasa-ubuntu:/usr/src/linux-2.6.32.60/kernel/CPU_Utilization# insmod CPU_Utilization.ko
root@nasa-ubuntu:/usr/src/linux-2.6.32.60/kernel/CPU_Utilization# dmesg | tail -n 5
[17696.291320] hid-generic 0003:80EE:0021.0006: input,hidraw0: USB HID v1.10 Mouse [VirtualBox USB Tablet] on usb-0000:00:06.0-l/input0
[17696.624189] e1000: eth0 NIC Link is Down
[17700.627777] e1000: eth0 NIC Link is Up 1000 Mbps Full Duplex, Flow Control: RX
[18568.686774] read file module remove successfully
[18577.744439] CPU Utilization: 10.00%
root@nasa-ubuntu:/usr/src/linux-2.6.32.60/kernel/CPU_Utilization#
```

Part 4: Result

有 `printk` 的部分直接在 `test_program` 中以 `system("dmesg|tail-1")` 的方式印出來了

```
w4a2y4@w4a2y4-VirtualBox:~/OSPJ1_Team28$ ./test.o
***SYS_HELLO:
[ 2736.223536] HELLO SYSTEM CALL

***SYS_SHOW:
[ 2736.233191] b04902077 Wei-Hsuan Chian
[ 2736.233192] b04902079 Chih-Wei Ning

***SYS_MULTIPLY(10,20)
200

***SYS_MIN(10,20)
10

***SYS_CPU_UTILIZATION
[ 2738.241235] CPU_Utilization: 100.00%

w4a2y4@w4a2y4-VirtualBox:~/OSPJ1_Team28$
```