

OS-Project-1

1. 設計

- FIFO
先來的process先執行，將ready time sort由小到大排序後按順序執行。
- RR (quantum = 500)
將所有process按照ready time由小到大排序。
每個process，若其ready time為當前的時間，則將該process放入一個queue中。
執行一個周期後將remaining execution time減少一個週期，未完成的process繼續放入queue中。重複動作到所有process完成。
- SJF
選擇執行時間最短的process來執行
將ready time sort由小到大排序，若ready time相同，再按照execution time由小到大排序。
每個process，若其ready time為當前的時間，則將該process放入一個heap中。heap會依照execution time由小到大排序。
如果目前沒有在執行process，並且heap中有process，則取出execution time最小的process並執行，直到該process執行結束。
- PSJF
每次選擇剩餘執行時間最短的 process 來執行，當新產生的 process 執行時間更短時，就可以插隊。
heap 頂端的process 會隨時與執行中的process比較，若其擁有更小剩餘執行時間，他就會插隊，並把原本執行中的process放回heap。

2. 核心版本

linux 4.14.25 @ubuntu 16.04

3. 比較實際結果與理論結果，並解釋造成差異的原因

順序上執行結果均與理論相同。

但存在有許多誤差，例如執行排程的時間、虛擬機的執行效率、signal傳遞的時間等都影響蠻大的。在本機上曾經遇到signal傳遞過慢的問題：process結束時，父程序一段時間後才收到SIG_CHLD導致排程亂掉。(process看起來執行的比預定時間久，導致應該先完成的process直到後面process都ready了才完成。在SJF、PSJF有重大影響。)

在換了一台硬體設備叫好後，問題就解決了，排程也正常執行。

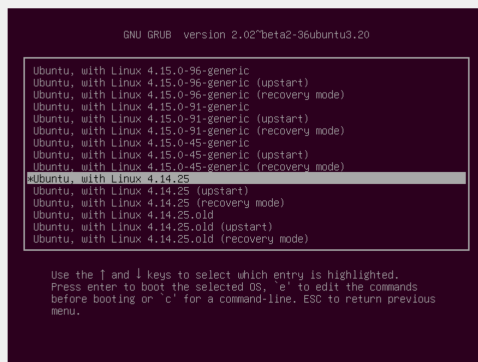
4. Difference

這次程式的實作我是用原本kernel裡的syscall去記錄時間和直接open dmesg去取代printk的功能。我有編完自己的kernel但是因為電腦ram太小所以跑不起來，匯出VM到別台電腦也打不開(不知道原因)。

權宜之計只好用最接近原本設定的方法完成此次project，希望助教體諒。

附上VM中ubuntu grub的畫面，證明我有編kernel(完成hw1)

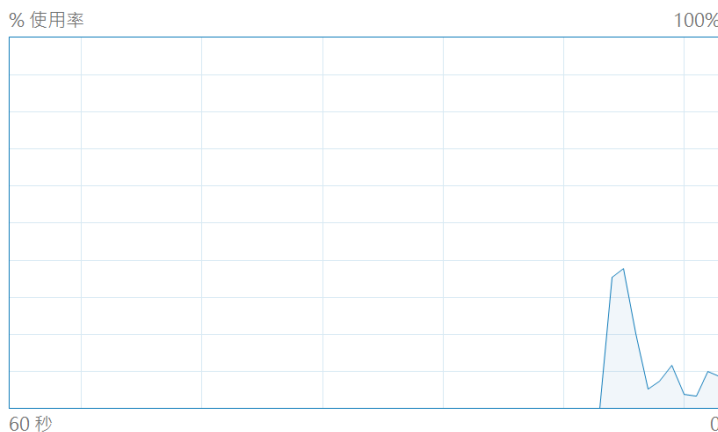
cnlab [執行中] - Oracle VM VirtualBox
檔案 機器 檢視 輸入 裝置 說明



電腦只有雙核心+4GM ram...compile kernel就等到天荒地老，最後跑不起來。

機器 使用者 詳細資料 服務

CPU Intel(R) Core(TM) i5-7300U CPU @ 2.60GHz



使用率 9% 速度 0.88 GHz
處理程序 226 執行緒 2984 控制代碼 109099
運作時間 2:09:08:42
基本速度: 2.71 GHz
插槽: 1
核心數目: 2
邏輯處理器: 4
模擬: 已啟用
L1 快取: 128 KB
L2 快取: 512 KB
L3 快取: 3.0 MB

系統 SKU Surface_Pro_1796
處理器 Intel(R) Core(TM) i5-7300U CPU @...
BIOS 版本/日期 Microsoft Corporation 234.2706.76...
SMBIOS 版本 3.1
內嵌控制器版本 255.255
BIOS 模式 UEFI
BaseBoard 製... Microsoft Corporation
BaseBoard 產... Surface Pro
BaseBoard 版... 無法使用
平台角色 平板電腦
安全開機狀態 開啟
PCR7 設定 需要提升才能檢視
Windows 目錄 C:\WINDOWS
系統目錄 C:\WINDOWS\system32
開機裝置 \Device\HarddiskVolume1
地區設定 台灣
硬體抽象層 版本 = "10.0.18362.752"
使用者名稱 DESKTOP-P2C29UN\yen
時區 台北標準時間
已安裝的實體... 4.00 GB
總共實體記憶體 3.93 GB
可用實體記憶體 296 MB
虛擬記憶體總計 11.1 GB
可用虛擬記憶體 3.87 GB
分頁檔空間 7.16 GB
分頁檔案 C:\pagefile.sys
核心 DMA 保護 關閉

5. Acknowledgement

感謝 wangyenjen 同學去年的github，以及在我遇到bug時給予我一些方向。