

作業系統 OS Report

姓名: 劉正鴻

學號: d07944009

1. 設計:

首先，在虛擬機器，設定的 Core 至少要兩個，因為這樣 parent process 和 child process 才能分開，否則會混亂。在程式裡面，設定一個 parent process 在一個 CPU 裡面能夠獨立運行，在 Ready queue 裡，他要所有的 process 都是 child process，因為到了 Ready time 的時間就會排到 Ready queue 裡，而這些 child process 的運作，會跟原本的 parent process 的計算分開，因此，可以確保運作的時間不會受到影響。

2. 核心版本:

我使用的虛擬機器為 Ubuntu 16.04 with Linux kernel 4.14.25

CPU: Intel(R) Core (TM) i7-9750H CPU @ 2.60GHz

照作業 1 的方式 call system call，就可以 call sys_process_time.c

先到先做 (First-in First-out, FIFO) scheduling: 是一個最簡單的不可搶先的 scheduling。它選擇下一個 process 的方式是根據 process 要求使用 CPU 的順序，先發出要求的 process 可以先取得 CPU 的使用權。實作 FIFO scheduling 時，我們可以使用 FIFO queue 來完成。當一個 process 進入 ready queue 時，process 的 PCB 會被連結至 queue 的末端，當 CPU 閒置時，scheduler 只需將 queue 中第一個 process 取出，再把 CPU 的使用權交給他。

最短工作優先 Shortest Job First (SJF) Scheduling: 根據 process 下一次 CPU 暴衝的長短選出下一個 process。CPU 暴衝最短的 process 可優先取得

CPU 的使用權；但是若兩個 process 下一次的 CPU 暴衝時間等長的話，則可以使用 FIFO scheduler 來做 scheduling。

可搶先的最短工作優先 Preemptive Shortest Job First: 就是 SJF 加上可以搶先。

優先權 Priority Scheduling: 每個 process 都有一個優先權，scheduler 依 process 的優先權來分配 CPU 的使用順序，優先權愈高的 process 可優先使用 CPU。當 process 的優先權相同時，可使用 FIFO scheduling 來決定執行的順序。SJF 也可以視為是一種優先權 scheduling，而其 process 的優先權由 process 下一次 CPU 暴衝時間的長短來決定，CPU 暴衝愈短的 process 優先權愈高。

循環分時 (Round-Robin) Scheduling: 是特別為分時系統所設計的。基本上 RR 與 FIFO scheduling 很相似，一樣是先到的 process 先做，不過 RR scheduling 是可搶先的。另外還有一個不同點是 RR scheduling 定義了很小的時間單位，稱為時間切片 (time slice) 或是時間量 (time quantum)，一個時間切片通常為 10~100 毫秒。RR scheduling 將時間等切成一小片一小片，每一片時間切片就是每個 process 每次得到 CPU 使用權後可執行的時間，當時間用完時，scheduler 就會由 ready queue 中取出下一個 process 做內文交換。使用 RR scheduling，可以使用 FIFO queue 來時做 ready queue，當一個新 process 進入 ready queue 時，會被放在 queue 的最末端，而 scheduler 會由 queue 的最前端取出 process，並將計時器設置成在一個時間切片後產生中斷來重新 scheduling，最後將 CPU 的使用權遞交給取出的 process。

3. 實際結果 vs 理論結果:

照實驗出來的結果，放在 output 資料夾裡，有分 stdout 跟 dmesg，依照輸出的單位時間結果，dmesg 輸出的結果誤差可以看出 scheduling 會受到執行時間的長度的影響以及準備時間的影響，導致每個 process 在順序上處理的時間有不一樣的結果。像是 RR scheduling，會有固定的時間中斷，將正在跑的 process 中斷掉，然後排到後面，輪到下一個 process 跑，也是執行在固定的單位時間，跑到超過單位時間的時候，也會強制中斷，跑下一個 process，以此類推，直到 process 執行時間結束。從 dmesg 以及 stdout 可以看得出來 process scheduling 的順序有根據 RR scheduling 結束的時間輸出，Terminal 印出的結束時間跟 dmesg 一致。