

## Design

---

原理很簡單，分配兩個 cores，一個 run processes，另外一個 scheduling。設置一個 process queue 決定 processes 的順序，每當 process 開始或交換的時候，就會 enqueue 當前的 process。之後，當 core 0(running processes)閒置的時候，就會 dequeue。Operations 分三種，分別是 dequeue, enqueue, low execution time priority enqueue。

FIFO: 開始 fork 的時候 enqueue，執行的時候 dequeue。

PSJF: 當新的 process 開始的時候，會 yield 當前的 process，然後執行兩個 low execution time priority enqueue，分別 enqueue 開始和當前的 process。

RR: 基本上，與 FIFO 類似，但會讓 running process 跑一個 quantum time 之後 yield，然後 enqueue 再 dequeue 直到跑完。

SJF: 當新的 process 開始的時候，執行 low execution time priority enqueue。與 PSJF 只差在 SJF 並沒有 yield 當前在跑的 process。

## Kernel Version

---

Ubuntu 4.14.176

## Practice versus theory

---

在 TIME\_MEASUREMENT.txt 測試的時候，start time 和 end time 差距有時候會突然暴漲。總體來說，雖然閒置與跑 process 都是 500 個 unit time，但是實際閒置 core 的時間會比跑 process 的還要少。照理來說，core 1(scheduling core) 會延遲，因為做比較多的 operation，造成相反的結果，一種可能是 syscall 需要花很多時間，延遲了 process 結束的時間。