report

架構

- scheduler
  scheduler獨享一個core，確保排程的正確性
- process
  其他的process皆使用同個core，一次只讓最多一個process有最高的優先度可以跑。
- core control
  使用CPU_SET()跟sched_setaffinity去控制process跟scheduler。
- priority control
  使用sched_setscheduler()，因為SCHED_FIFO對於SCHED_IDLE來說有最高的優先度，所以將現在要跑的process設為SCHED_FIFO，其他的設為SCHED_IDLE。

排程設計

- FIFO
  若有人正在跑則不做任何事情，沒有人在跑就從按照生成時間順序sort過的process array中找到第一個已經產生且還沒跑完的
- RR
  用一個queue去管理即將進入的process。
  利用lastswihchtime去記錄上次contextswitch的時間，若跟現在的時間差為500或是當前process執行完則從queue拿出第一個執行
- PSJF
  每輪都找當前剩餘exectime最小的
- SJF
  若目前沒有人在跑則找一個剩餘exectime最小的，若有人正在跑則不做任何事情。

kernel版本

linux 14.14.24

與理論結果的比較

因為我的程式是由一個scheduler計算何時該換人跑，但是scheduler獨享一個core，其他process共享一個core，可能因為context switch，造成計數比scheduler慢一些，就會造成一點誤差。
在跑這個的時候，core可能還有在做其他的事情，也會造成誤差。