Report

一、設計

- **Makefile** :
  編譯所有程式
- **main.c** :
  讀入所有input，並放進scheduler
- **scheduler.c** :
  - get_next() :
    - 若policy是FIFO:
      - 若現在的process尚未結束，則繼續執行
      - 若現在無process，則找下一個process
    - 若policy是RR
      - 若現在無process，則pop()出一個
      - 若現在的process尚未結束，則每500 unit time都pop()一個
    - 若policy是SJF或policy是PSJF
      - 若是SJF且現在的process尚未結束，則繼續執行
      - 每次有新的process ready，則檢查所有ready的process，並執行execu_time最小的
  - scheduler()
    assign給自己CPU0並給自己最高priority
    - while(1):
      - 檢查pocess是否完成:
        若proc[running_id].execu_time==0則wait它，並將已結束的process數加一
      - 檢查是否有process ready:
        若有process ready則fork child去執行並將priority降至最低
        若是RR則push至queue
      - 檢查是否有process 要執行:
        若get_next拿到的process和現在正在跑的不同，則將現在正在跑的降至低prority，將新的process升至最高priority
  - 跑unit time
- **process.c**:
  - assign_CPU():
    指派CPU給process
  - proc_execu():
    fork child並執行process，且計算總時間
    assign CPU1給該process
  - proc_wake():
    將process設至高priority
  - proc_block():
    將process設至低priority
- **process.h**:
  funcion和structure的宣告

二、核心版本

三、比較與結論

1. scheduler的計時器不會完全和process的同步
2. scheduler除排程之外還有調整process的priority等的時間
3. 跑一個一單位時間的process需要的時間不只一個單位，一次跑多個單位做平均較恰當