

## hw2

110590002 王燊竑

### 4.8

1. write file
2. task is quick than split new thread

### 4.10

not shared: (a) Register, (d) Stack memory shared: (b) heap memory, (c) global memory

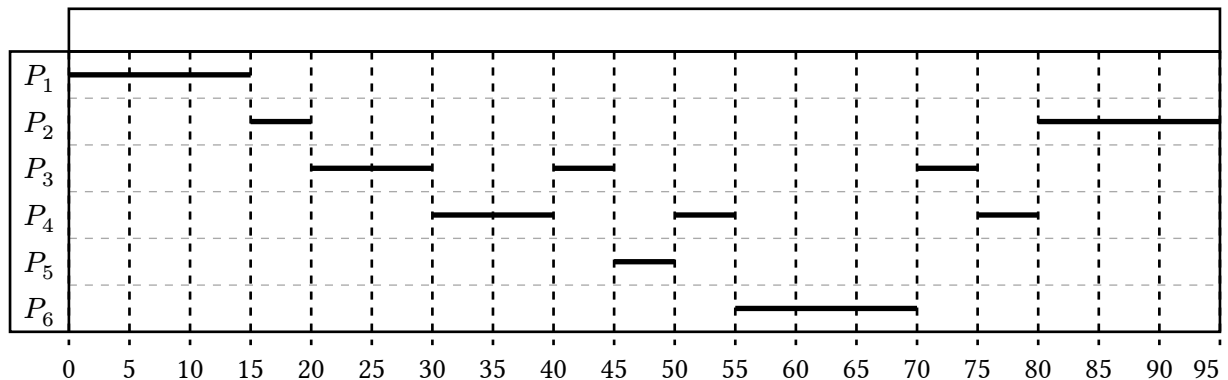
### 4.16

1. 1 thread will be created for input, 1 thread will be created for output.
  - many thread use I/O will be blocked, so it is better to use 1 thread for I/O
2. 4 thread will be created for processing
  - four thread can process 4 task in same time, so it is better to use 4 thread for processing

### 5.14

1. each processing core has its own run queue:
  - advantage:
    - each core can process task in parallel
  - disadvantage:
    - need to implement a mechanism to balance the load of each core
2. a single run queue is shared by all processing cores:
  - advantage:
    - easy to implement
  - disadvantage:
    - if many task in run queue, it will be blocked

### 5.18



1. turnaround time:

- $P_1$ : 15
- $P_2$ : 95
- $P_3$ : 55
- $P_4$ : 55
- $P_5$ : 5
- $P_6$ : 15

2. waiting time:

- $P_1$ : 0
- $P_2$ : 75

- $P_3$ : 35
- $P_4$ : 35
- $P_5$ : 0
- $P_6$ : 0

### 5.22

1. the time quantum is 1 milisecond:  
the CPU utilization is  $\frac{1+10}{(1+0.1)+(1+0.1)*10} = 90.90\%$
2. the time quantum is 10 milisecond:  
the CPU utilization is  $\frac{10+10}{(10+0.1)+(1+0.1)*10} = 94.78\%$

### 5.25

1. FCFS  
short process will be blocked by long process
2. RR  
all process have same fixed time to run, then short process want run need to wait front process is queue
3. Multilevel feedback queues  
all process process in first level queue, if the process is too long to run, it will be moved to next level queue. so short process can run in first level queue

### 6.7

- (a) when the task length is 1, and two thread run pop in same time, will happen race condition, and segfault
- (b) let the stack function is atomic, so the race condition will be fixed

### 6.15

1. loss of responsiveness: disabling interrupts will let single-processor system can't response to other task. In user level will look like freeze, it is not good for user experience.
2. system stability: if the interrupt is disabled, the system can't response to hardware interrupt, so the system will be unstable.

### 6.18

the acquire a lock request will be enqueue to a waiting queue. when acquired the lock, the thread will be blocked by waiting queue, when unlock the lock, make next request have request.