05. CPU Scheduling

Created	@March 24, 2022 6:15 PM
Tags	Lee Edith.D

Concepts

Scheduling Algorithm

- maximum CPU utilization

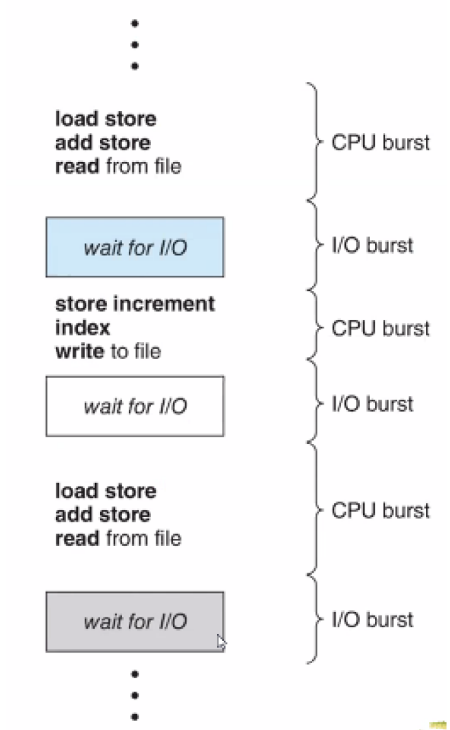
Time Concepts

- ATT (Average Turnaround Time) 周转: $\sum(time\ when\ process\ finish - time\ when\ process\ begin)/N$
- AWT (Average Waiting Time) 等待: $\sum(Start - Arrive)/N$
  - Arrival Time:
  - Start to Execute
  - Wait Time

- Execute Time
- End Time

Burst

- CPU burst
- I/O burst
- most computers is I/O bound program



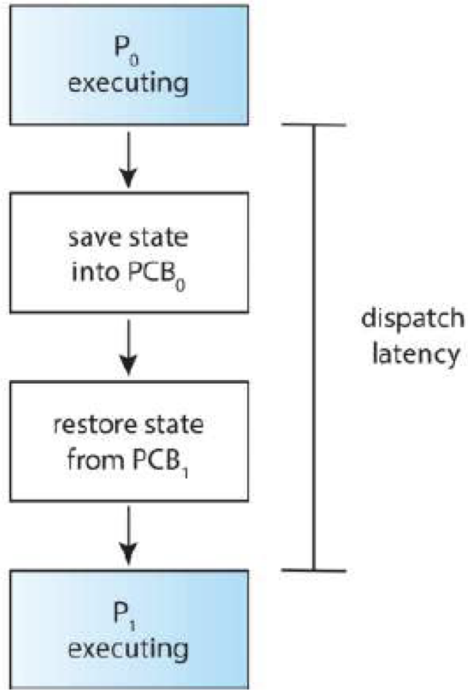
CPU Scheduler

Scheduler

- select a process to be executed

Dispatcher

- context switch
- kernel mode →
  usr mode



- 分派器模块将CPU的控制权交给了短期调度程序选择的进程。
  - 切换上下文
  - 切换到用户态
  - 跳到用户程序中适当的位置以重新启动该程序

- what a CPU scheduler do:
  - load new PCB
  - save old PCB
  - switch between kernel mode and usr mode
  - read PC
- 指标
  - response time:
    - **submit** → first response
  - **turn around time:**
    - submit→complete
  - **waiting time:**

        - time in ready queue, waiting to be sent to running
        - submit - arrive
    - throughput:
        - amounts of processes in a per unit time
    - CPU utilization:

First-Come First-Served (FCFS)

process	waiting time	arrive time
P1	0	0
P2	24	0
P3	27	0

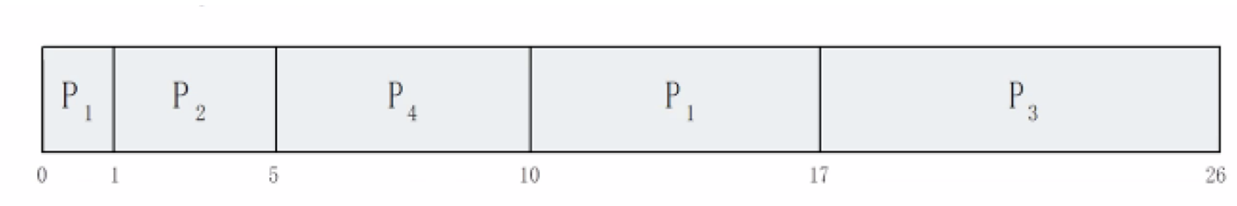
- cons:
    - ATT, AWT is long

Shortest-Job-First (SJF)

- ideal
- AWT is the shortest

Shortest-Remaining-Time-First (SRTF)

Process	*Arrival* Time	Burst Time
P_1	0	8
P_2	1	4
P_3	2	9
P_4	3	5



- 抢占式 (preemptive)
- AVT = [(10-1)+(0)+(17-2)+(5-3)]/4
- cons:
  - Starvation （饿死）
  - long process can’t execute

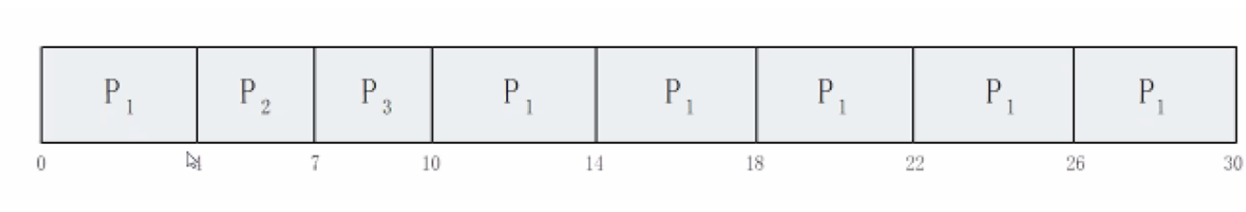
Priority-Scheduling

Process	Burst Time	Priority
P_1	10	3
P_2	1	1
P_3	2	4
P_4	1	5
P_5	5	2

- each process is allocated with a priority(integer)
- implement way to do scheduling
- solution to starvation
    - aging
- static priority setting consideration
    - resources in use
    - execution time
    - process state(kernel / usr, front-end / back-end, IO/CPU bound)

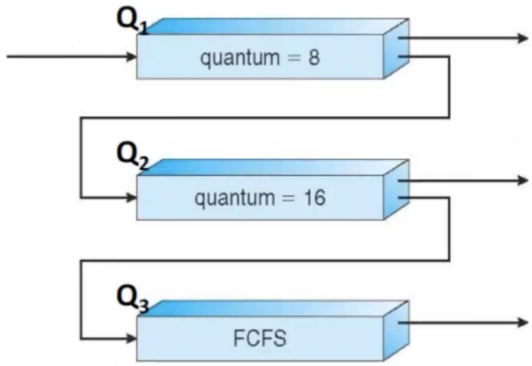
Round-Robin

Process	Burst Time
P_1	24
P_2	3
P_3	3

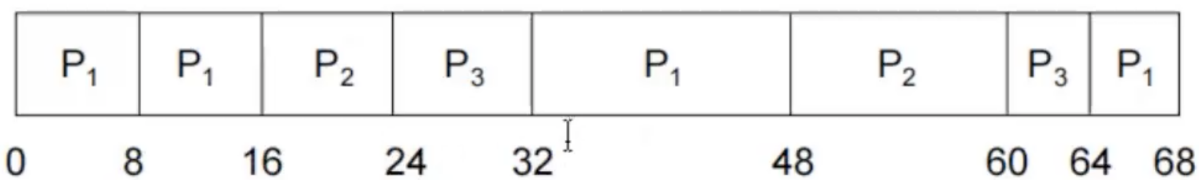


- FCFS
- preemptive
- time quantum q
    1. each process gets a small unit of CPU time (q)
    2. if not finished, the process is preempted and added to the end of ready queue
    - q > 80% processes

Multilevel Feedback Queue



Process	*Arrival* Time	Burst Time
P_1	0	36
P_2	16	20
P_3	20	12



- different levels have different priorities
- different levels have an isolated queue

1. a process enters Q1
    a. if not finish within $q1$
    b. enters Q2
2. priority of Q2 is lower than Q1

-