

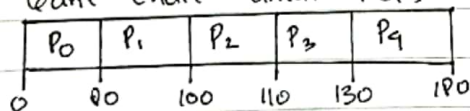
1. Misalkan Sebuah algoritma Penjadwalan lebih menyukai Proses-proses yang menggunakan waktu prosesor paling sedikit di masa lalu. Mengapa algoritma ini akan mendukung program yang terikat I/O namun tidak secara permanen membuat program yang terikat CPU-bound kelaparan?

→ Jawab : Algoritma Penjadwalan yang mengutamakan Proses-proses dengan penggunaan CPU yang sedikit dalam waktu terbaru akan lebih menguntungkan Program-program I/O karena mereka hanya membutuhkan CPU dengan waktu yang singkat sebelum menunggu I/O, sehingga mereka sering mendapatkan giliran CPU. Namun, Program-program CPU bound tidak akan kelaparan karena Program-program I/O sering melepaskan CPU untuk menunggu I/O, memberikan kesempatan bagi Program-program CPU bound untuk mendapatkan giliran CPU secara teratur. Dengan cara tersebut, algoritma ini akan menjaga keseimbangan antara responsivitas untuk program I/O dan keadilan bagi Program CPU-bound.

2. Misal terdapat tugas-tugas yang harus dijalankan dengan satu prosesor :

i	$t(p_i)$
0	80
1	20
2	10
3	20
4	50

- a. Gantt Chart untuk FCFS



- b. berapa waktu yang dibutuhkan untuk menyelesaikan proses P_3

→ turnaround P_3 : waktu selesai - waktu kedatangan

$$\text{turnaround } P_3 = 130 - 0 \\ = 130$$

Jadi, waktu yang dibutuhkan untuk menyelesaikan proses P_3 adalah 130 //

C. berapa waktu tunggu rata-rata untuk Proses tersebut?

$$\text{rata-rata waktu tunggu} = \frac{(0 + 80 + 100 + 110 + 130)}{5}$$

$$= \frac{420}{5} = 84 //$$

3. Misalkan sebuah proses baru dalam sebuah sistem tiba dengan rata-rata 6 proses per menit dan setiap proses tersebut membutuhkan rata-rata 8 detik waktu layanan. perkirakan fraksi waktu CPU sibuk dalam sistem dg prosedur tunggal.

→ diketahui : laju kedatangan rata-rata proses (λ) = 6 proses per menit
waktu pelayanan rata-rata untuk setiap proses : 8 detik per proses

ditanya : fraksi waktu dimana CPU sibuk dalam sistem dg satu prosesor

jawab : total waktu pelayanan per unit : 6 proses \times 8 detik / proses
= 48 detik

$$\text{Fraksi waktu CPU sibuk} = \frac{48 \text{ detik}}{60 \text{ detik}} = 0,8 //$$

Jadi ~~waktu~~ fraksi waktu CPU sibuk adalah 0,8 atau 80%

4. Sebuah algoritma Penjadwalan CPU menentukan urutan eksekusi berbagai proses. Jika ada proses yang akan dijadwalkan pada satu prosesor, berapa banyak jadwal yang mungkin ada? berikan rumus dalam n

→ jawab : Jumlah total permutasi dari n objek diberikan oleh $n!$, yang didefinisikan sebagai :

$$n! = n \times (n-1) \times (n-2) \times \dots \times 1$$

5. Hubungan antara Pasangan Algoritma Penjadwalan CPU

a. Priority dan SJF

- Priority : memilih proses berdasarkan ~~ke~~ prioritas yang telah ditentukan

- SJF : Memilih proses dengan waktu eksekusi terpendek berikutnya

hubungan : SJF adalah subset dari algoritma penjadwalan berbasis prioritas yang ditentukan oleh waktu eksekusi yang tersisa.

b. ~~Priority~~ Multilevel Feedback queues dan FCFS

- FCFS : Memproses pekerjaan dalam urutan kedatangan

- Multilevel feedback queues : memiliki beberapa antrian dengan prioritas

yang berbeda dan dapat menggunakan algoritma di setiap antrian hubungan; FCFS bisa menjadi salah satu algoritma yang digunakan dlm salah satu antrian multilevel feedback queues,

C. Priority dan FCFS

- Priority : memilih proses berdasarkan prioritas
- FCFS : Memproses pekerjaan dalam urutan kedatangan

Hubungan: FCFS bisa dianggap sebagai algoritma berbasis prioritas dimana semua proses memiliki prioritas yang sama & prioritas ditentukan oleh waktu kedatangan.

d. Round robin & SJF

- RR : setiap proses mendapatkan waktu yang sama dlm urutan siklus
- SJF : memilih proses dg waktu eksekusi terpendek berikutnya

Hubungan: RR dan SJF tidak memiliki hubungan langsung dalam hal parameterisasi atau konsep dasar mereka.

6. Perbedaan antara Penjadwalan Jangka Panjang dan Jangka Pendek

- Jawab:
- Penjadwalan jangka panjang bertujuan untuk memilih proses untuk dimasukkan ke memori utama
 - Penjadwalan jangka pendek bertujuan untuk memilih proses untuk dimasukkan ke CPU.

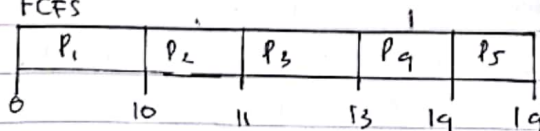
7. Perbandingan serangkaian proses berikut dengan durasi CPU yang diberikan

Process	burst time	priority
P_1	10	3
P_2	1	1
P_3	2	3
P_4	1	4
P_5	5	2

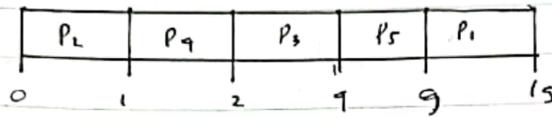
Proses diasumsikan tiba dalam urutan P_1, P_2, P_3, P_4, P_5 semua pada waktu 0

a. diagram gantt

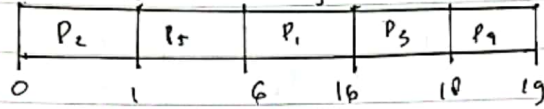
1. FCFS



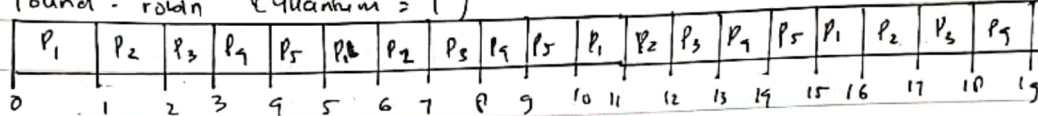
2. SJF



3. non preemptive Priority



4. round - robin (Quantum = 1)



b. turnaround time

1. FCFS

Process	Turnaround time
P ₁	10
P ₂	11
P ₃	13
P ₄	14
P ₅	19

2. SJF

Process	turnaround
P ₂	1
P ₄	2
P ₃	4
P ₅	9
P ₁	19

3. non preemptive Priority

process	turnaround
P ₂	1
P ₅	6
P ₁	16
P ₃	18
P ₄	19

4. RR (Quantum = 1)

Process	turnaround
P ₁	10
P ₂	2
P ₃	7
P ₄	9
P ₅	12

C. Waiting time for each process

1. FCFS

Process	Waiting time
P ₁	0
P ₂	10
P ₃	11
P ₄	13
P ₅	19

2. SJF

Process	Waiting time
P ₂	0
P ₄	1
P ₃	2
P ₅	9
P ₁	9

3. non Preemptive Priority

Process	Waiting time
P ₂	0
P ₅	1
P ₁	6
P ₃	16
P ₄	18

4. RR (quantum = 1)

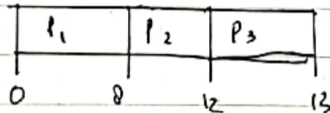
Process	Waiting time
P ₁	6
P ₂	1
P ₃	5
P ₄	3
P ₅	7

D. Pertimbangkan rangkaian Proses berikut, dengan lamanya CPU burst dan waktu kedatangannya

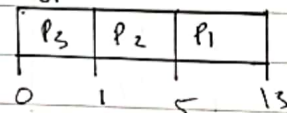
Process	burst time	Priority
P ₁	8	0
P ₂	9	0.9
P ₃	1	1

a. Gantt Chart

1. FCFS



2. SJF



b. turnaround time

1. FCFS

Process	t-time
P ₁	8
P ₂	12
P ₃	13

2. SJF

Process	t-time
P ₂	1
P ₂	5
P ₁	13

C. Waiting time

1. FCFS

Process	w. time
P ₁	6
P ₂	8
P ₃	12

2. SJF

Process	w. time
P ₃	6
P ₂	1
P ₁	5

9. Jelaskan Perbedaan Sejalan mana algoritma penjadwalan berikut ini

a. first come first ~~served~~ served

Proses eksekusi sesuai urutan kedatangan dalam antrian. tidak ada diskriminasi khusus untuk proses pendek. FCFS tidak membedakan proses berdasarkan panjangnya.

b. round robin

Setiap proses diberi quantum waktu tetap sebelum dipindahkan ke belakang antrian. Round robin memberikan kesempatan yang sama bagi semua proses.

c. Multilevel feedback queues

memberikan prioritas tinggi pada proses pendek dengan menempatkannya di antrian prioritas tinggi.

10. tulis Catatan singkat tentang

a. Waiting time

Jumlah waktu yang dihabiskan oleh sebuah proses dalam ~~antrian~~ ~~ready~~ ready queue sebelum mulai dieksekusi oleh CPU.

b. response time

Waktu yang diperlukan oleh sistem untuk merespons permintaan awal pengguna.

c. throughput

Mengacu pada jumlah pekerjaan yang selesai dalam satu unit waktu oleh sistem / komponen sistem tertentu.