

Nama : Alyssa Nadcha Syahira

NRP : 3123521018

Kelas : TI-A PSDKU - LA

1) Misalkan Sebuah algoritma Penjadwalan lebih mengutamakan Proses yang menggunakan Waktu Prosesor Paling Sedikit baru-baru ini. Mengapa algoritma ini lebih menguntungkan Program I/O-bound dan tidak membuat Program CPU-bound kelaparan secara permanen?

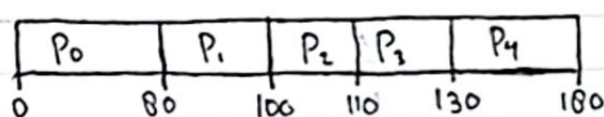
Jawab:

Algoritma Penjadwalan yang mengutamakan Proses yang telah menggunakan Waktu Prosesor Paling Sedikit akan lebih memprioritaskan Proses I/O-bound. Hal ini karena proses I/O-bound biasanya menggunakan waktu CPU yang lebih sedikit dan lebih sering menunggu operasi I/O selesai. Dengan memberi prioritas pada proses-proses ini, algoritma memastikan bahwa mereka mendapatkan waktu CPU yang dibutuhkan, mencegah terjadinya blokir dan meningkatkan efisiensi sistem secara keseluruhan. Sementara itu, proses CPU-bound yang membutuhkan lebih banyak waktu tetap mendapatkan giliran mereka, meskipun mungkin harus menunggu lebih lama.

2) Misal terdapat tugas-tugas yang harus dijalankan dengan satu proses :

i	T(pi)
0	80
1	20
2	10
3	20
4	50

(a) Diagram Gantt untuk Penjadwalan FCFS



(b) Waktu Turnaround untuk proses P3

> Proses P3 mulai pada waktu 110 (setelah P0, P1, dan P2 selesai)

> Proses P3 berjalan selama 20 unit. Jadi selesai pada waktu 130.

$$\text{Waktu turnaround} = 130 - 0 = 130 \text{ unit waktu}$$

(c) Rata-rata waktu tunggu untuk proses tersebut

$$\text{Rata-rata waktu tunggu} = \frac{\text{Total waktu tunggu}}{\text{Jumlah proses}}$$

$$\text{Rata-rata waktu tunggu} = \frac{0 + 80 - 0 + 100 - 0 + 110 - 0 + 130 - 0}{5} = \frac{420}{5} = 84 //$$

3) Diket : Rata-rata proses = 6 proses/menit

$$\text{Rata-rata waktu layanan (T)} = 8 \text{ detik per proses}$$

$$= \frac{8}{60} \text{ menit} = \frac{2}{15} \text{ menit}$$

Ditanya : Utilisasi CPU (U)....?

$$\text{Jawab : } U = \lambda \times T_s$$

$$U = 6 \text{ proses/menit} \times \frac{2}{15} \text{ menit per proses}$$

$$= 6 \times \frac{2}{15} = \frac{12}{15}$$

$$= 0,8$$

Fractions waktu CPU yang sibuk dalam sistem dgn satu prosesor adalah 0.8 / 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 adalah $n!$, yg didefinisikan sebagai Produk dari semua bil bulat positif hingga n :

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

Jdi rumus untuk menghitung Jumlah Jadwal yg mungkin dari n proses adalah $n!$

5) Hubungan antara Algoritma Penjadwalan CPU

(a) Priority dan SJF

Priority : Menjalankan proses berdasarkan Prioritas

SJF (Shortest Job First) : Memilih proses dengan waktu eksekusi terpendek

Hubungan : SJF adalah kasus khusus dari Penjadwalan Prioritas dimana prioritas didasarkan pada panjang waktu eksekusi.

(b) Multilevel Feedback Queues dan FCFS

MLFQ : Menggunakan beberapa antrian dengan Prioritas berbeda. Proses dapat berpindah antar antrian.

FCFS : Menjalankan proses berdasarkan urutan kedatangan.

Hubungan : FCFS dapat digunakan sebagai algoritma Penjadwalan dalam antrian di MLFQ.

(c) Priority dan FCFS

Priority : Menjalankan proses berdasarkan Prioritas.

FCFS : Menjalankan proses berdasarkan urutan kedatangan.

Hubungan : FCFS adalah kasus khusus dari Penjadwalan Prioritas dimana semua proses memiliki Prioritas yg sama.

(d) RR dan SJF

RR (Round Robin) : Memberikan waktu yang sama untuk setiap proses secara bergilir

SJF : Memilih proses dengan waktu eksekusi terpendek.

Hubungan : Tidak ada hubungan langsung RR dan SJF pendekatan berbeda untuk penjadwalan.

No.

Date

6.) Perbedaan antara Penjadwalan Jangka Panjang dan Jangka Pendek

- Jangka Panjang : Memutuskan Proses mana yang akan diterima ke dalam sistem (penjadwalan Pekerjaan)
- Jangka Pendek : Memutuskan Proses mana diantara yang ada di antrian siap yang akan dieksekusi berikutnya oleh CPU (penjadwalan CPU).

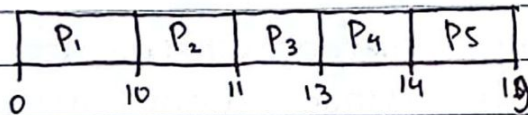
7) Pertimbangkan Serangkaian Proses berikut, dengan durasi CPU yang diberikan.

Process	Burst time	Priority
P ₁	10	3
P ₂	1	1
P ₃	2	3
P ₄	1	4
P ₅	5	2

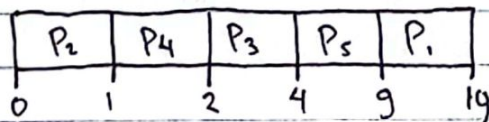
Proses diasumsikan tiba dalam urutan P₁, P₂, P₃, P₄, P₅ Semua Pd waktu 0

(a) Diagram Gantt

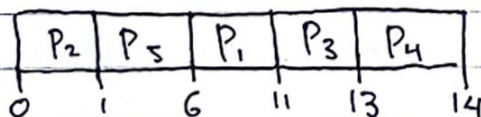
1. FCFS (First-Come, First-Served)



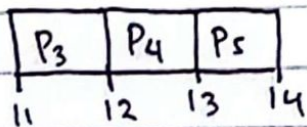
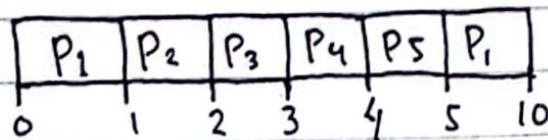
2. SJF (Shortest Job First):



3. Non-Preemptive Priority:



4. RR (Round-Robin) dengan quantum = 2:



(b) turnaround time

> FCFS:

- P₁ : 19 ms
- P₂ : 1 ms
- P₃ : 11 ms
- P₄ : 13 ms
- P₅ : 19 ms

> SJF:

- P₁ : 19 ms
- P₂ : 1 ms
- P₃ : 9 ms
- P₄ : 14 ms
- P₅ : 14 ms

> Non-preemptive Priority:

- P₁ : 13 ms
- P₂ : 1 ms
- P₃ : 11 ms
- P₄ : 14 ms
- P₅ : 19 ms

> RR

- P₁ : 10 ms
- P₂ : 14 ms
- P₃ : 13 ms
- P₄ : 14 ms
- P₅ : 14 ms

(c) Waiting time

> FCFS:

- P₁ : 0 ms
- P₂ : 0 ms
- P₃ : 9 ms
- P₄ : 11 ms
- P₅ : 4 ms

> SJF:

- P₁ : 9 ms
- P₂ : 0 ms
- P₃ : 0 ms
- P₄ : 12 ms
- P₅ : 1 ms

> Non-preemptive Priority

- P₁ : 6 ms
- P₂ : 0 ms
- P₃ : 0 ms
- P₄ : 12 ms
- P₅ : 4 ms

> RR

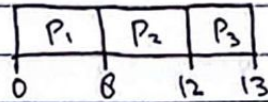
- P₁ : 9 ms
- P₂ : 0 ms
- P₃ : 1 ms
- P₄ : 12 ms
- P₅ : 9 ms

8.) Pertimbangkan Rangkaian Proses berikut, dengan ramanya CPU burst dan waktu kedatangannya.

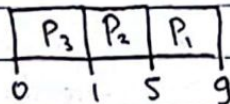
Process	Burst time	Priority
P ₁	8	0
P ₂	4	0.4
P ₃	1	1

(a) Diagram Gantt

1. FCF (First - Come, First - Served);



2. SJF (Shortest Job First)



(b) Turnaround time

> FCFS

- P₁ : 13 ms
- P₂ : 9 ms
- P₃ : 12 ms

> SJF:

- P₁ : 9 ms
- P₂ : 8 ms
- P₃ : 12 ms

(c) Waiting time

> FCFS:

- P₁ : 5 ms
- P₂ : 0 ms
- P₃ : 0 ms

> SJF

- P₁ : 1 ms
- P₂ : 0 ms
- P₃ : 0 ms

9.) Jelaskan Perbedaan sejauh mana algoritma Penjadwalan berikut ini:

(a) FCFS (First Come First Served)

Tidak memberikan Preferensi terhadap Proses-Proses Pendek. Proses dieksekusi sesuai urutan kedatangan, tanpa memperhatikan durasi eksekusi

(b) Round Robin (RR)

Memberikan Sedikit Preferensi terhadap Proses-Proses Pendek dengan memberikan Potongan Waktu CPU yang sama kepada Setiap Proses. Namun, Proses-Proses Pendek tetap bisa mengalami Penundaan jika kuantitas Waktu terlalu besar.

(c) Multilevel Feedback Queues (MLFQ)

- Memberikan Preferensi besar terhadap Proses-Proses Pendek dengan menempatkan mereka di antrian Prioritas tinggi. Proses-Proses Pendek dapat menyelesaikan eksekusi lebih cepat karena mendapatkan lebih banyak Waktu CPU di Prioritas tinggi.

10) Tuliskan Catatan Singkat tentang

(a) Waiting time

Waktu yang dihabiskan oleh suatu proses dalam antrian siap sebelum mulai dieksekusi oleh CPU

(b) Response time

Waktu yang dibutuhkan oleh sistem untuk merespons permintaan awal dari pengguna atau proses yang meliputi persiapan proses dan alokasi sumber daya.

(c) Throughput

Jumlah pekerja yang diselesaikan oleh sistem dalam satu unit waktu tertentu.