

Nama : Divanadia Ramadhani

NRP : 3123521026 Kelas : D3 TI-A

No. \_\_\_\_\_

Date : \_\_\_\_\_

- ☐ 1. Misalkan sebuah algoritma penjadwalan lebih menguntungkan proses-proses yang telah menggunakan waktu prosesor paling sedikit dalam waktu dekat.
- ☐ Mengapa algoritma ini akan lebih menguntungkan program yang I/O-bound dan tidak akan selamanya membuat program yang CPU-bound?

☐ Jawab :

- ☐ Algoritma ini menguntungkan program yang I/O-bound karena mereka sering menunggu operasi I/O dan hanya membutuhkan sedikit waktu CPU. Oleh karena itu, mereka cepat keluar dari antrian dan menggunakan sedikit waktu prosesor. Program yang CPU-bound menggunakan lebih banyak waktu CPU tetapi akan mendapat giliran mereka karena algoritma ini hanya mempertahankan waktu penggunaan CPU terbaru, bukan total waktu yang telah digunakan.

- ☐ 2. Misalkan anda memiliki pekerjaan-pekerjaan berikut untuk dieksekusi dengan satu prosesor, dengan pekerjaan-pekerjaan tiba dalam urutan yang tercantum di sini :

| i | $T(p_i)$ |
|---|----------|
| 0 | 80       |
| 1 | 20       |
| 2 | 10       |





|   |    |
|---|----|
| 3 | 20 |
| 4 | 50 |

a.) Misalkan sistem menggunakan penjadwalan FCFs. Buat bagan Gantt yang menggambarkan eksekusi proses-proses ini.

Jawab :

|   | P0 | P1  | P2  | P3  | P4  |
|---|----|-----|-----|-----|-----|
| 0 | 80 | 100 | 110 | 130 | 180 |

b.) Berapa waktu penyelesaian untuk proses P3?

Jawab :

$130 - 0 = 130$ . Waktu penyelesaian untuk P3 adalah 130 milidetik.

c.) Berapa rata-rata waktu tunggu untuk proses-proses?

Jawab :

Rata-rata waktu tunggu adalah :

$$\frac{0 + 80 + 100 + 110 + 130}{5} = 84 \text{ milidetik}$$

3. Misalkan proses baru dalam sistem tiba rata-rata enam proses per menit dan setiap proses memerlukan waktu layanan rata-rata 8 detik. Perkiraan



☐ kan fraksi waktu CPU yang sibuk dalam sistem dengan satu prosesor

☐ Jawab :

☐ Diketahui :

☐ • Rata-rata kedatangan proses baru : 6 proses per menit

☐ • Rata-rata waktu layanan untuk setiap proses : 8 detik

☐ - Total waktu layanan dalam satu menit :  
 $6 \times 8 = 48$  detik

☐ - Fraksi waktu CPU yang sibuk :  
 $\frac{48 \text{ detik}}{60 \text{ detik}} = 0,8$

☐ Jadi, Fraksi waktu CPU yang sibuk adalah 0.8 atau 80%

☐ 4. Algoritma pengadwalan CPU menentukan urutan eksekusi proses-proses yang dijadwalkan. Diberikan  $n$  proses untuk dijadwalkan pada satu prosesor, berapa banyak jadwal berbeda yang mungkin ada? Berikan formula dalam istilah  $n$ .

☐ Jawab :

☐ Setiap urutan eksekusi dari  $n$  proses adalah sebuah permutasi dari  $n$  elemen. Dalam teori kombinasi, jumlah permutasi dari  $n$  elemen





☐ adalah  $n$  faktorial ( $n!$ ). Maka, jumlah urutan  
☐ berbeda yang mungkin untuk  $n$  proses adalah :  
☐  $n! = n \times (n-1) \times (n-2) \times \dots \times 2 \times 1$

☐ 5. Algoritma ini dengan demikian sebenarnya adalah  
☐ kumpulan algoritma (misalnya, kumpulan algoritma  
☐ RR untuk semua irisan waktu, dan sbgnya). Satu  
☐ set algoritma mungkin termasuk yang lain (misalnya,  
☐ algoritma FCFS adalah algoritma RR dengan  
☐ kuantum waktu tak terbatas). Apa (jika ada)  
☐ hubungan antara pasangan set algoritma berikut ?

☐ Jawab :

☐ a) Priority dan SJF

☐ SJF adalah bentuk khusus dari Penjadwalan  
☐ prioritas ditentukan oleh panjang burst time.

☐ b) Multilevel Feedback Queues dan FCFS

☐ FCFS dapat digunakan dalam salah satu  
☐ antrian dalam Multilevel Feedback Queues.

☐ c) Priority dan FCFS

☐ Tidak ada hubungan langsung antara Priority  
☐ Scheduling dan FCFS karena mereka menggu  
☐ nakan kriteria yang berbeda untuk menentu  
☐ kan urutan eksekusi

☐ d) RR dan SJF

☐ Tidak ada hubungan langsung antara RR





dan SJF karena RR dan SJF karena mereka didasarkan pada kriteria yang berbeda.

6. Bedakan antara Penjadwalan jangka panjang dan penjadwalan jangka pendek.

Jawab :

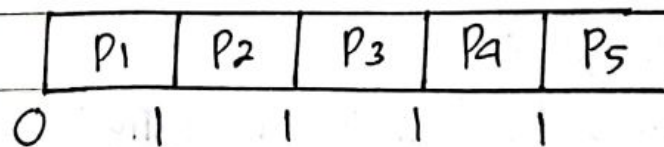
Penjadwalan jangka panjang mengatur proses mana yang akan dimasukkan ke dalam sistem untuk diproses (sering terkait dengan pekerjaan batch). Penjadwalan jangka pendek mengatur proses dimana antrian siap yang akan dijalankan selanjutnya (sering disebut sebagai Penjadwalan CPU)

7. Pertimbangkan set proses berikut, dengan panjang burst CPU yang diberikan dalam milidetik.

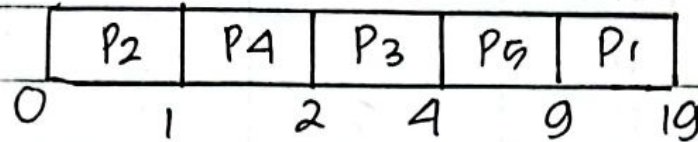
| Process        | Burst Time | Priority |
|----------------|------------|----------|
| P <sub>1</sub> | 10         | 3        |
| P <sub>2</sub> | 1          | 1        |
| P <sub>3</sub> | 2          | 3        |
| P <sub>4</sub> | 1          | 4        |
| P <sub>5</sub> | 5          | 2        |

a. Gambar empat bagan Gantt

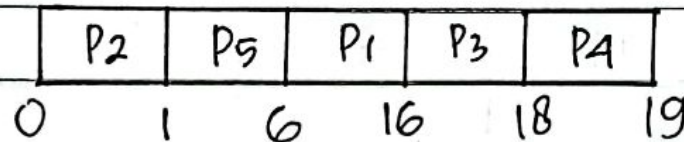
• FCFS :



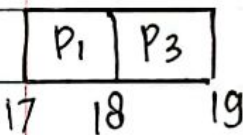
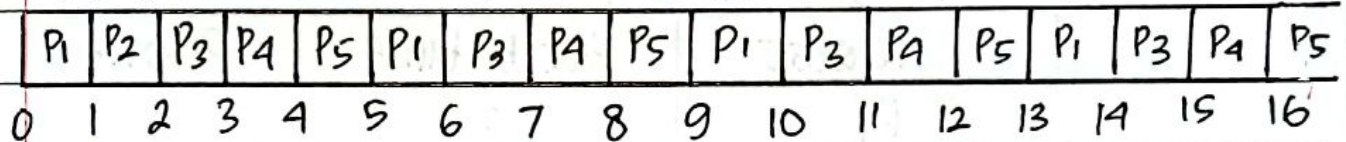
• SJF :



• Prioritas :



• RR (Kuantum = 1)



(b) Waktu penyelesaian setiap proses di bagian (a)

| Proses         | FCFS | SJF | Prioritas | RR |
|----------------|------|-----|-----------|----|
| P <sub>1</sub> | 10   | 19  | 16        | 19 |
| P <sub>2</sub> | 11   | 1   | 1         | 2  |
| P <sub>3</sub> | 13   | 4   | 18        | 13 |
| P <sub>4</sub> | 14   | 2   | 19        | 14 |
| P <sub>5</sub> | 19   | 9   | 6         | 9  |

(c) Waktu tunggu setiap proses di bagian (a)





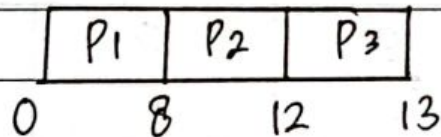
| Proses         | FCFS | SJF | Prioritas | RR |
|----------------|------|-----|-----------|----|
| P <sub>1</sub> | 0    | 9   | 6         | 9  |
| P <sub>2</sub> | 10   | 0   | 0         | 1  |
| P <sub>3</sub> | 11   | 2   | 16        | 10 |
| P <sub>4</sub> | 13   | 1   | 18        | 13 |
| P <sub>5</sub> | 14   | 4   | 1         | 4  |

8. Pertimbangkan rangkaian proses berikut dengan lamanya CPU burst dan waktu kedatangannya.

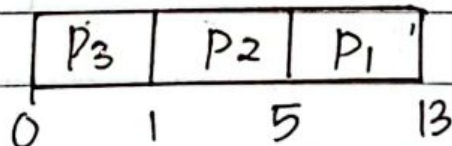
| Process        | Burst time | Priority |
|----------------|------------|----------|
| P <sub>1</sub> | 8          | 0        |
| P <sub>2</sub> | 4          | 0.4      |
| P <sub>3</sub> | 1          | 1        |

a.) Gantt chart

• FCFS :



• SJF :



☐ b.) Turnaround Time

| <input type="checkbox"/> | Proses         | FCFS | SJF |
|--------------------------|----------------|------|-----|
| <input type="checkbox"/> | P <sub>1</sub> | 8    | 13  |
| <input type="checkbox"/> | P <sub>2</sub> | 12   | 5   |
| <input type="checkbox"/> | P <sub>3</sub> | 13   | 1   |

☐ c.) Waiting time

| <input type="checkbox"/> | Proses         | FCFS | SJF |
|--------------------------|----------------|------|-----|
| <input type="checkbox"/> | P <sub>1</sub> | 0    | 5   |
| <input type="checkbox"/> | P <sub>2</sub> | 8    | 1   |
| <input type="checkbox"/> | P <sub>3</sub> | 12   | 0   |



9. Jelaskan Perbedaan dalam sejauh mana algoritma penjadwalan berikut menguntungkan proses yang pendek.

a.) First come First served (FCFS)

Dalam algoritma FCFS, proses dieksekusi dalam urutan kedatangan mereka. FCFS tidak menguntungkan proses pendek karena tidak mempertimbangkan burst time dari proses. Proses yang datang lebih awal akan dieksekusi terlebih dahulu, terlepas dari durasinya.

b.) Round Robin (RR)

Dalam algoritma Round Robin, setiap proses mendapatkan waktu eksekusi yang sama dalam unit waktu tertentu (time quantum) secara bergilir. Round Robin lebih menguntungkan proses pendek daripada FCFS karena setiap proses mendapatkan kesempatan yang sama untuk dieksekusi.

c.) Multilevel Feedback Queues (MLFQ)

Dalam algoritma MLFQ, terdapat beberapa tingkat antrian dengan prioritas berbeda, dan proses bisa pindah antrian berdasarkan perilaku mereka. MLFQ sangat menguntungkan proses pendek karena proses yang menggunakan CPU dalam





☐ dalam waktu singkat dapat dipromosikan ke antrian dengan prioritas lebih tinggi. ☐ MLFQ akan lebih sering mendapatkan akses ke CPU. ☐

☐ 10. Tulis catatan singkat tentang :

☐ a.) Waiting time

☐ Waiting time adalah jumlah total waktu yang dihabiskan oleh proses dalam antrian sebelum mendapat kesempatan untuk dieksekusi. ☐ Waiting time penting karena berpengaruh pada performa sistem secara keseluruhan. ☐ Semakin lama menunggu, semakin rendah efisiensi sistem. ☐

☐ b.) Response time

☐ Response time adalah waktu yang diambil dari saat proses mengirimkan permintaan eksekusi hingga pertama kali proses tersebut dieksekusi. ☐ Response time penting dalam sistem interaktif, dimana pengguna mengharapkan respon cepat setelah mengirimkan permintaan. ☐

☐ c.) Throughput

☐ Throughput adalah jumlah proses yang





dapat diselesaikan dalam satu satuan waktu. Throughput meng kur produktivitas sistem. Semakin tinggi throughput, semakin banyak pekerjaan yang dapat diselesaikan dalam periode waktu tertentu.

