

Laporan Simulasi Scheduling Algorithm dengan Menggunakan Python

Kelompok 3:

- Calvin Ritchie Rumendong (0806022310003)
- Tiffany Tjandinegara (0806022310027)
- Franklin Jaya (0806022310016)
- Michael Christianto Sawitto (0806022310028)

Soal:

1. Lakukan simulasi penjadwalan dengan menggunakan algoritma:

- a. First Come First Serve**
- b. Shortest Job First (Non preemptive)**
- c. Shortest Job First (preemptive)**
- d. Longest Job First (preemptive)**
- e. Round Robin (Quantum time = 12)**

2. Tentukan yang mana algoritma terbaik dan terburuk untuk data ini lalu berikan penjelasannya dalam sebuah pdf file.

Sistem kerja Scheduling Algorithm berdasarkan code Python yang telah dibuat, selalu mengutamakan process yang terlebih dahulu sampai, bukan hanya cara kerja CPU Scheduling Algorithm saja. Adapun beberapa Algoritma Scheduling yang digunakan pada kode Python, yaitu :

- **First Come First Serve (FCFS)**
- **Shortest Job First (SJF) Non-preemptive**
- **Shortest Job First (SJF) Preemptive**
- **Longest Job First (LJF) Preemptive**
- **Longest Job First (LJF) Non-Preemptive**
- **Round Robin (RR)**

Adapun tujuan dari pembuatan laporan ini mengetahui waktu rata rata Waiting dan Turnaround Time masing masing Algoritma, serta menentukan mana Scheduling Algorithm terbaik di antara ke-5 Scheduling Algorithm yang ada.

Setelah melakukan percobaan melalui pembuatan code Python, dari percobaan tersebut kami mendapatkan perbandingan laporan mengenai rata rata Waiting Time dan Turnaround Time, sebagai berikut :

Algoritma	Rata-rata Waiting Time	Rata-rata Turnaround Time
Shortest Job First (SJF) Preemptive	476.13	488.71
Shortest Job First (SJF) Non-Preemptive	477.91	490.19
Round Robin	786.27	796.95
First Come First Serve (FCFS)	585.00	597.58
Longest Job First (LJF) Preemptive	1172.99	1185.57
Longest Job First (LJF) Non-Preemptive	723.74	736.32

Berdasarkan hasil laporan rata rata di atas, berikut adalah hasil analisis dari ke-5 scheduling algorithm yang ada :

1. Shortest Job First (SJF) Preemptive:

- **Deskripsi:** Shortest Job Preemptive adalah Scheduling Algorithm yang bekerja dengan mendahulukan burst time yang lebih kecil dari burst time yang sedang berlangsung. Tujuan SJF Preemptive adalah Mendahulukan Burst time yang lebih pendek.
- **Kelebihan:** Karena suatu proses mungkin tidak memonopoli prosesor, ini adalah metode yang lebih dapat diandalkan, Waktu respons rata-rata ditingkatkan, Sistem

operasi memastikan bahwa setiap proses yang menggunakan CPU menggunakan jumlah waktu CPU yang sama.

- **Kekurangan:** Menangguhkan proses yang sedang berjalan, mengubah konteks, dan mengirimkan proses masuk yang baru semuanya memerlukan lebih banyak waktu, proses berprioritas rendah harus menunggu jika beberapa proses berprioritas tinggi tiba pada saat yang sama.
- **Cara kerja :** Jadi, misalkan ada 3 proses :

P1: arrival time = 0, burst time = 10

P2: arrival time = 2, burst time = 4

P3: arrival time = 4, burst time = 2

P1 tiba pada dtk 0 dan berjalan selama 2 dtk. Setelah itu, pengerjaan P1 dihentikan karena terdapat P2 yang memiliki burst time lebih pendek, tetapi setelah P2 berjalan 2 detik prosesnya kembali dihentikan karena ada P3 yang memiliki burst time lebih pendek. Jadi pada detik ke 6 P3 telah selesai dan P2 kembali dilanjutkan, setelah P2 selesai pada detik ke 8, P1 kembali dilanjutkan hingga selesai.

2. Shortest Job First (SJF) Non-preemptive:

- **Deskripsi:** SJF Non-preemptive adalah Scheduling yang berlawanan dengan SJF Preemptive karena jika SJF preemptive mendahulukan burst time terpendek SJF Non-preemptive juga mendahulukan burst time paling pendek, tetapi tidak dapat menghentikan eksekusi yang telah berjalan. Tujuan SJF Non-preemptive adalah mendahulukan burst time terpendek meski sesuai waktu kedatangan.
- **Kelebihan:** Beban penjadwalannya minimal, komputasi yang digunakan lebih sedikit, memiliki tingkat throughput yang tinggi.
- **Kekurangan:** Waktu responnya terhadap proses itu super, bug dapat mengakibatkan komputer macet.
- **Cara kerja :** Jadi, misalkan ada 3 proses :

P1: arrival time = 0, burst time = 10

P2: arrival time = 2, burst time = 4

P3: arrival time = 4, burst time = 2

P1 yang pertama kali tiba akan diselesaikan dahulu selama 10 detik, setelah itu terdapat 2 proses yang terdapat pada antrian yaitu P2, dan P3. Pada detik ke 10, P3 akan dikerjakan hingga selesai terlebih dahulu sesuai algoritma Shortest Job First (SJF) Non-preemptive, setelah P3 selesai dieksekusi P2 akan diselesaikan hingga detik ke 16.

3. Round Robin (RR):

- **Deskripsi:** Round Robin (RR) adalah Scheduling Algorithm yang dapat memotong proses jika telah melewati Quantum time yang telah ditetapkan, maksud dari

memotong adalah mendahulukan proses lain yang lebih dulu sampai jika lewat dari Quantum time yang ditetapkan.

- **Kelebihan:** Efektif untuk sistem multiuser, Memiliki *overhead* yang kecil jika ukuran proses yang rata-rata lebih kecil dibandingkan slot waktunya.
- **Kekurangan:** Jika tempatnya terlalu kecil, maka sebagian proses tidak bisa diselesaikan oleh satu waktu saja, memiliki performa yang buruk jika *quantum time* nya lebih besar dari pada prosesnya di banding FCFS.
- **Cara kerja :** Jadi, misalkan ada 3 proses :

P1: arrival time = 0, burst time = 13

P2: arrival time = 2, burst time = 4

P3: arrival time = 4, burst time = 2

Quantum time = 12

P1 yang pertama kali tiba akan diselesaikan dahulu selama 12 detik, jadi tersisa 1 detik lagi karena (12-13). Nah, setelah itu proses P2 akan dilanjutkan hingga selesai (12-4), setelah sampai pada detik 16, P1 akan dilanjutkan kembali selama 1 dtk / waktu yang tersisa (12-1). Setelah itu, P1 akan langsung diselesaikan.

Note : cth (12 - 13). 12 itu Quantum timenya

4. First Come First Serve (FCFS):

- **Deskripsi:** FCFS adalah Scheduling Algorithm yang mendahulukan proses lain yang lebih dulu sampai dan akan diselesaikan, lalu dilanjutkan oleh proses yang lebih dulu hingga selesai. Tujuan FCFS adalah menyelesaikan proses yang pertama kali sampai, setelah itu lanjut lagi ke proses yang sampai terlebih dahulu selanjutnya.
- **Kelebihan:** Sederhana dan mudah dipahami, FCFS memberikan keadilan dengan memperlakukan semua proses secara setara dan memberikannya kesempatan yang sama untuk dijalankan.
- **Kekurangan:** Algoritma FCFS khususnya bermasalah untuk sistem multiprogramming, yang mengharuskan setiap pengguna mendapat jatah CPU secara berkala, lebih mengutamakan proses yang terikat CPU daripada proses yang terikat I/O.
- **Cara kerja :** Jadi, misalkan ada 3 proses :

P1: arrival time = 0, burst time = 13

P2: arrival time = 2, burst time = 4

P3: arrival time = 4, burst time = 2

P1 pertama kali akan diselesaikan karena memiliki arrival time terkecil, lalu P2 akan diselesaikan kedua. Setelah itu, P3 akan diselesaikan. Intinya sama seperti siapa cepat dia dapat

5. Longest Job First Preemptive (LJF Preemptive):

- **Deskripsi:** LJF Preemptive adalah Scheduling Algorithm yang mendahulukan burst time tertinggi terlebih dahulu sesuai waktu arrivalnya, tetapi eksekusinya akan dihentikan jika ada proses yang lebih tinggi burst timenya (kurang lebih sama seperti SJF Preemptive).
- **Kelebihan:** Semua proses selesai kira-kira pada waktu yang sama, yang dapat bermanfaat dalam skenario tertentu.
- **Kekurangan:** Algoritma LJF dapat menyebabkan waktu tunggu rata-rata dan waktu penyelesaian rata-rata yang tinggi untuk serangkaian proses tertentu, yang dapat berdampak negatif pada kinerja sistem, efek konvoi dapat terjadi, yang berarti proses pendek harus menunggu proses panjang untuk selesai, sehingga menyebabkan penundaan yang tidak perlu.
- **Cara kerja :** Jadi, misalkan ada 3 proses :

P1: arrival time = 0, burst time = 13

P2: arrival time = 2, burst time = 14

P3: arrival time = 4, burst time = 4

Pertama - tama P1 akan masuk pada detik 0 sampai detik ke-2, P2 akan masuk dan dieksekusi karena memiliki burst time yang lebih besar, hingga pada detik ke-16 P1 akan dilanjutkan hingga selesai pada detik ke-27, P3 akan masuk menggantikan P1 hingga selesai.

6. Longest Job First Non-Preemptive (LJF Non-Preemptive):

- **Deskripsi:** LJF Non-Preemptive adalah Scheduling Algorithm yang mendahulukan burst time tertinggi terlebih dahulu sesuai waktu arrivalnya, tetapi tidak seperti Preemptive ia tidak dapat dibatalkan sekali eksekusi.
- **Kelebihan:** Semua proses selesai kira-kira pada waktu yang sama, yang dapat bermanfaat dalam skenario tertentu.
- **Kekurangan:** Algoritma LJF dapat menyebabkan waktu tunggu rata-rata dan waktu penyelesaian rata-rata yang tinggi untuk serangkaian proses tertentu, yang dapat berdampak negatif pada kinerja sistem, efek konvoi dapat terjadi, yang berarti proses pendek harus menunggu proses panjang untuk selesai, sehingga menyebabkan penundaan yang tidak perlu.
- **Cara kerja :** Jadi, misalkan ada 3 proses :

P1: arrival time = 0, burst time = 13

P2: arrival time = 2, burst time = 2

P3: arrival time = 4, burst time = 4


P1 akan masuk pertama pada detik ke 0 karena sampai pertama, setelah P1 selesai pada detik 13, maka pada detik ke 14 akan masuk P3 karena algoritma LJF menerapkan burst time tertinggi terlebih dahulu. Selanjutnya, P2 akan dilanjutkan hingga selesai

Langkah-Langkah agar code python pada google collab dapat dijalankan :

1. masukkan file excel yang ingin dibuat scheduling algorithm nya ke folder sample_data.
2. lalu run code sesuai scheduling algorithm yang butuh.

Setelah selesai melakukan pembuatan code python dan menganalisis apa saja kelebihan, kekurangan dari masing masing scheduling algorithm, dapat kita tarik kesimpulan dari percobaan yang kita lakukan sebagai berikut :

- SJF Preemptive adalah Scheduling Algorithm terbaik, karena memiliki rata - rata waiting dan turnaround time terpendek diantara Scheduling Algorithm lain.
- LJF Preemptive menjadi Scheduling Algorithm paling buruk dan tidak disarankan untuk digunakan, karena memiliki rata rata waiting dan turnaround time, serta ia tidak cocok digunakan untuk sistem real-time karena lambat.
- Scheduling Algorithm LJF Preemptive memiliki rata rata waiting dan turnaround yang sangat berbeda dengan LJF Non-preemptive, karena efek preemptive pada LJF yang mendahulukan proses dengan burst time lebih besar pada CPU yang menyebabkan banyak proses yang seharusnya bisa selesai dengan cepat jadi macet / bertumpuk, sedangkan meskipun LJF Non-Preemptive juga mendahulukan proses burst time terbesar, tetapi ia tetap menyelesaikan proses yang telah dimulai sehingga proses tidak bertumpuk.
- SJF Preemptive lebih baik dari SJF Non-Preemptive, karena meski SJF preemptive dapat mengganti proses yang dieksekusi sama seperti LJF Preemptive, tetapi itu adalah hal yang baik karena dengan terus menerus bertukar dengan yang lebih pendek burst timenya ia dapat menyelesaikan banyak proses.

Link Google Collab :  Tugas OS.ipynb

<https://colab.research.google.com/drive/1befSnll6Aik05YrHsk9yj8TFzP7NCjhR?usp=sharing>