

Mata Kuliah : Sistem Operasi

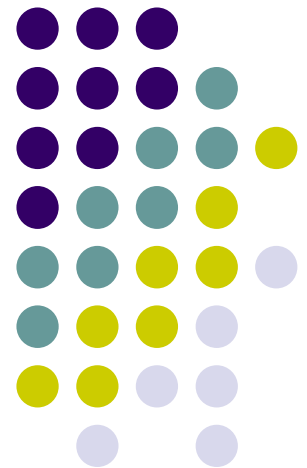
Kode MK : IK393330

6

Penjadualan CPU

Tim Teaching Grant
Mata Kuliah Sistem Operasi

Made Agung Raharja.,M.Cs





Standar Kompetensi :

- Mengetahui konsep dari Manajemen Proses

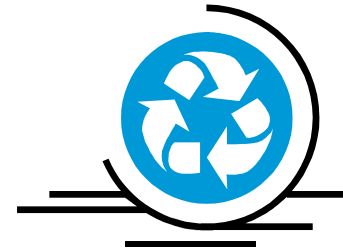
Pengalaman Belajar :

- Menjelaskan tentang manajemen proses antara lain penjadwalan (FCFS, SJF, RR),



Indikator Pencapaian :

- Mahasiswa mampu memahami dan menjelaskan tentang manajemen penjadwalan CPU.





Penjadualan CPU

- Konsep Dasar
- Kriteria Penjadualan
- Algoritma Penjadualan
- Evaluasi Algorithm



Penjadual CPU

- Algoritma scheduling:
 - **Memilih** dari proses-proses yang berada di memori (ready to execute) dan memberikan jatah CPU ke salah **satu** proses tersebut.
- Kapan keputusan untuk algoritma dilakukan:
 - Saat suatu proses:
 - 1.Switch dari status running ke waiting.
 - 2.Switch dari status running ke ready.
 - 3.Switch dari status waiting ke ready.
 - 4.Terminates.
 - Penjadualan 1 dan 4 termasuk nonpreemptive
 - Penjadualan lainnya termasuk preemptive



Jenis Penjadualan

- Preemptive: OS dapat mengambil (secara interrupt, preempt) CPU dari satu proses setiap saat.
- Non-preemptive: setiap proses secara sukarela (berkala) memberikan CPU ke OS.
- Contoh:
 - Penjadualan untuk switch dari running ke wait atau terminate: *non-preemptive*.
 - Penjadualan proses dari running ke ready: *pre-emptive*.
 - *Prasyarat untuk OS real-time system.*



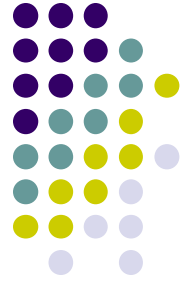
Kriteria Penjadualan

- Utilisasi CPU: menjadikan CPU terus menerus sibuk (menggunakan CPU semaksimal mungkin).
- Throughput: maksimalkan jumlah proses yang selesai dijalankan (per satuan waktu).
- Turn around time: minimalkan waktu selesai eksekusi suatu proses (sejak di submit sampai selesai).
- Waiting time: minimalkan waktu tunggu proses (jumlah waktu yang dihabiskan menunggu di ready queue).
- Response time: minimalkan waktu response dari sistim terhadap user (interaktif, time-sharing system), sehingga interaksi dapat berlangsung dengan cepat.

Kriteria Penjadualan yang Optimal



- Memaksimumkan utilisasi CPU
- Memaksimumkan throughput
- Meminimumkan turnaround time
- Meminimumkan waiting time
- Meminimumkan response time



Algoritma Penjadualan

- First-come, first-served (FCFS)
- Shortest-Job-First (SJF)
- Priority
- Round-Robin (RR)

First-Come, First-Served (FCFS)



- Algoritma:
 - Proses yang request CPU pertama kali akan mendapatkan jatah CPU.
 - Sederhana – algoritma maupun struktur data: menggunakan FIFO queue (ready queue).
- FIFO: Non preemptive
 - Timbul masalah “waiting time” terlalu lama jika didahului oleh proses yang waktu selesainya lama.
 - Tidak cocok untuk time-sharing systems.
 - Digunakan pada OS dengan orientasi batch job.



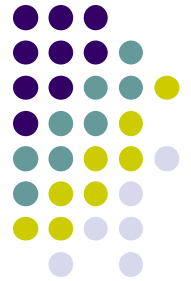
FCFS (Cont.)

- Example:

Process	Burst Time
P_1	24
P_2	3
P_3	3
- Diketahui proses yang tiba adalah P_1 , P_2 , P_3 . Gant chart-nya adalah :

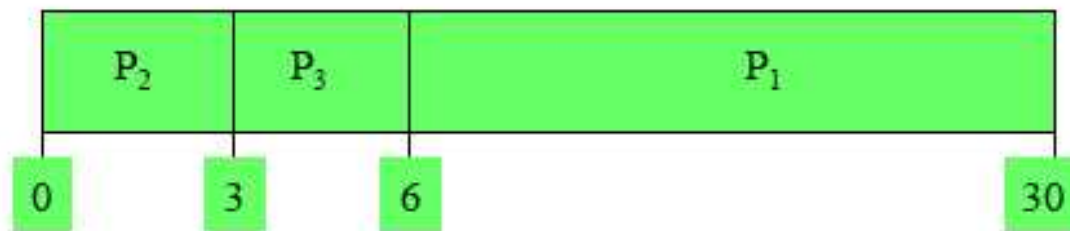


- Waiting
- Average waiting time: $(0 + 24 + 27)/3 = 17$



FCFS (Cont.)

- Diketahui proses yang tiba adalah P_2 , P_3 , P_1 .
Gant chart-nya adalah :



- Waiting time untuk $P_1 = 6$; $P_2 = 0$; $P_3 = 3$
- Average waiting time: $(6 + 0 + 3)/3 = 3$
 - Lebih baik dari kasus sebelumnya
- *Convoy effect* proses yang pendek diikuti proses yang panjang



Shortest-Job-First (SJF)

- Penggabungan setiap proses merupakan panjang dari burst CPU berikutnya. Panjang tersebut digunakan untuk penjadualan proses pada waktu terpendek
- Terdapat 2 skema :
 - *nonpreemptive* – CPU hanya satu kali diberikan pada suatu proses, maka proses tersebut tetap akan memakai CPU hingga proses tersebut melepaskannya
 - *preemptive* – jika suatu proses tiba dengan panjang CPU burst lebih kecil dari waktu yang tersisa pada eksekusi proses yang sedang berlangsung, maka dijalankan preemptive. Skema ini dikenal dengan Shortest-Remaining-Time-First (SRTF).
- SJF akan optimal, ketika rata-rata waktu tunggu minimum untuk set proses yang diberikan



Contoh Non-Preemptive SJF

	<u>Process</u>	<u>Arrival Time</u>	<u>Burst Time</u>
--	----------------	---------------------	-------------------

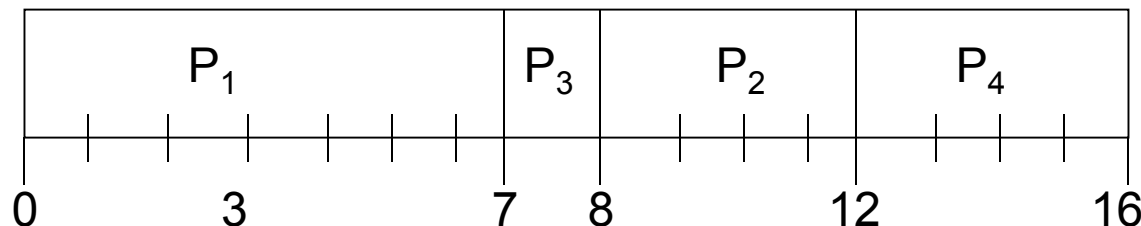
	P_1	0.0	7
--	-------	-----	---

	P_2	2.0	4
--	-------	-----	---

	P_3	4.0	1
--	-------	-----	---

	P_4	5.0	4
--	-------	-----	---

- SJF (non-preemptive)



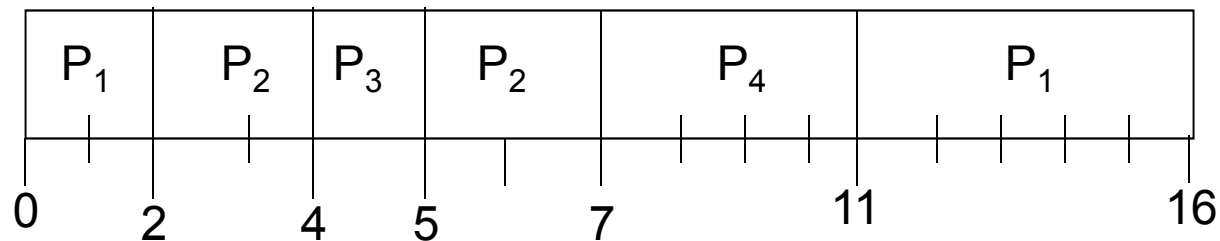
- Average waiting time = $(0 + 6 + 3 + 7)/4 - 4$



Contoh Preemptive SJF

Process	Arrival Time	Burst Time
P_1	0.0	7
P_2	2.0	4
P_3	4.0	1
P_4	5.0	4

- SJF (preemptive)



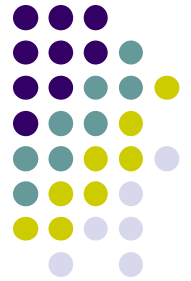
- Average waiting time = $(9 + 1 + 0 + 2)/4 = 3$



Penjadualan Prioritas

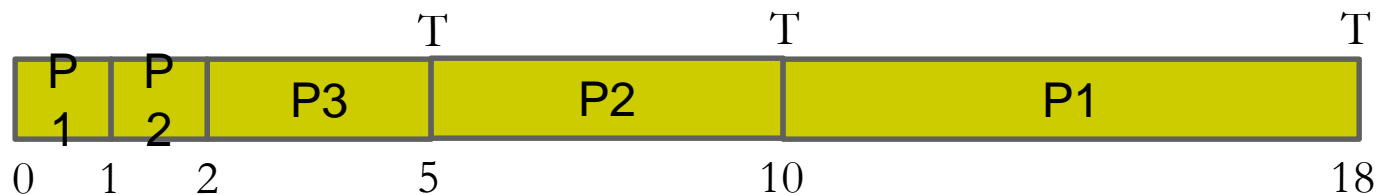
- Algoritma:
 - Setiap proses akan mempunyai prioritas (bilangan integer).
 - CPU diberikan ke proses dengan prioritas tertinggi (smallest integer ⁰ highest priority).
 - Preemptive: proses dapat di interupsi jika terdapat prioritas lebih tinggi yang memerlukan CPU.
 - Nonpreemptive: proses dengan prioritas tinggi akan mengganti pada saat pemakaian time-slice habis.
 - SJF adalah contoh priority scheduling dimana prioritas ditentukan oleh waktu pemakaian CPU berikutnya.
- Problem = Starvation
 - Proses dengan prioritas terendah mungkin tidak akan pernah dieksekusi
 - Solution = Aging
 - Prioritas akan naik jika proses makin lama menunggu waktu jatah CPU.

Contoh Kasus Priority Preemptive



PROSES	BRUST TIME	WKT KEDATANGAN	PRIORITY
P1	9	0	3
P2	6	1	2
P3	3	2	1

Gant Chart :



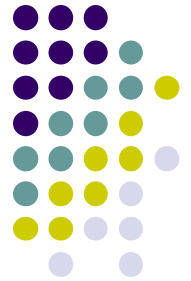
Waiting Time $P1 = 0 + (10 - 1) = 9$ | $P2 = 0 + (5 - 2) = 3$ | $P3 = 0$

Average Waiting Time $= \frac{9 + 3 + 0}{3} = 4$

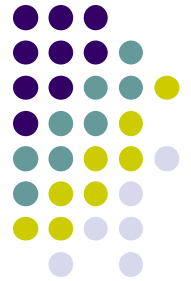
Turn Around Time $P1 = 18$ | $P2 = 10 - 1 = 9$ | $P3 = 5 - 2 = 3$

Average Turn Around Time $= \frac{18 + 9 + 3}{3} = 10$

CONTOH



Proses	AT	BT	Size (kb)
P1	0	10	100
P2	2	8	150
P3	3	12	175
P4	5	5	100



Round Robin (RR)

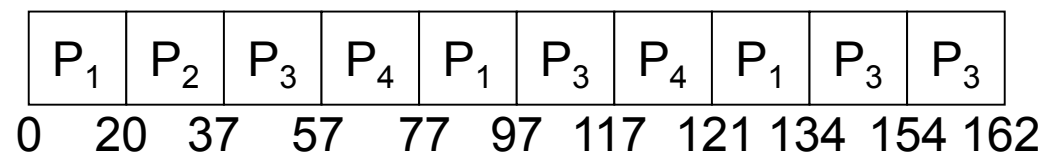
- Setiap proses mendapat jatah waktu CPU (time slice/quantum) tertentu misalkan 10 atau 100 milidetik.
 - Setelah waktu tersebut maka proses akan di-preempt dan dipindahkan ke ready queue.
 - Adil dan sederhana.
- Jika terdapat n proses di “ready queue” dan waktu quantum q (milidetik), maka:
 - Maka setiap proses akan mendapatkan $1/n$ dari waktu CPU.
 - Proses tidak akan menunggu lebih lama dari: $(n-1) q$ time units.
- Performance
 - q besar \Rightarrow FIFO
 - q kecil $\Rightarrow q$ harus lebih besar dengan mengacu pada context switch, jika tidak overhead akan terlalu besar



Contoh RR (Q= 20)

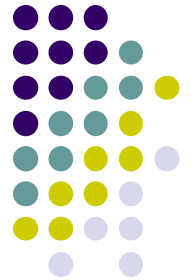
<u>Process</u>	<u>Burst Time</u>
P_1	53
P_2	17
P_3	68
P_4	24

- Gantt Chart



- Tipikal: lebih lama waktu rata-rata turnaround dibandingkan SJF, tapi mempunyai response terhadap user lebih cepat.

Latihan Soal...

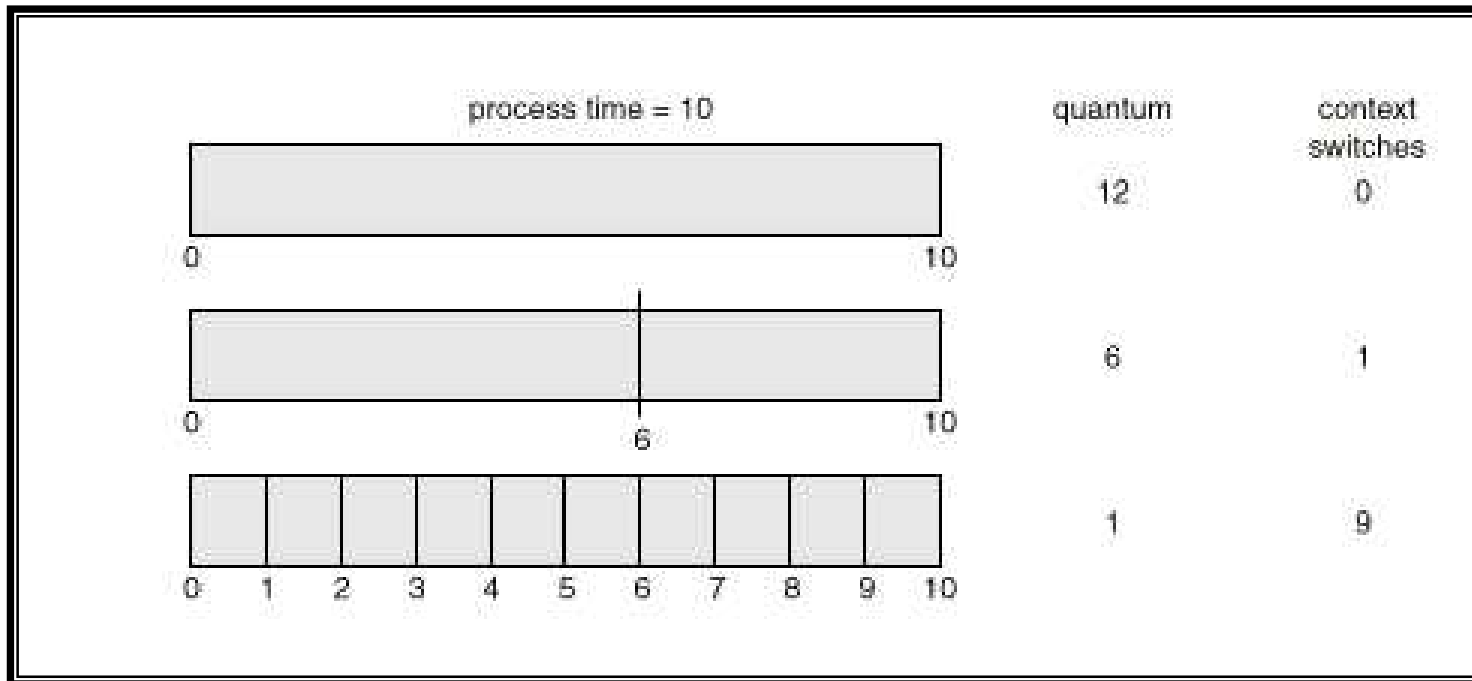
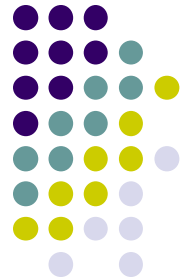


PROSES	BURST TIME	PRIORITY	WKT KEDATANGAN
P1	10	3	0
P2	4	1	2
P3	5	2	4

Gambar Gant Chart dan Tentukan Avg. Waiting Time serta Avg. Turn Around Time dari ketiga proses di atas dengan algoritma :

- a) FCFS
- b) SJF (preemptive)
- c) Round-Robin, $q = 2$
- d) Round-Robin, $q = 5$
- e) Priority (preemptive)

Waktu Kuantum dan Waktu Context Switch





Rangkuman :

- Algoritma penjadwalan yang terdiri atas First-come, first-served (FCFS), Shortest-Job-First (SJF), Priority dan Round-Robin (RR) merupakan beberapa algoritma yang digunakan dalam suatu sistem operasi.
- Setiap model algoritma penjadwalan memiliki kelebihan dan kekurangan masing-masing tergantung kasus yang digunakan.

Q & A

