SISTEM OPERASI

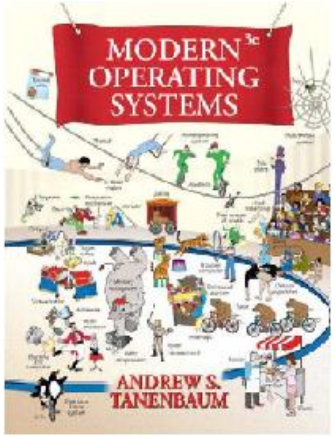
Departemen Ilmu Komputer/ Informatika
Universitas Diponegoro
Semester Gasal 2018/ 2019

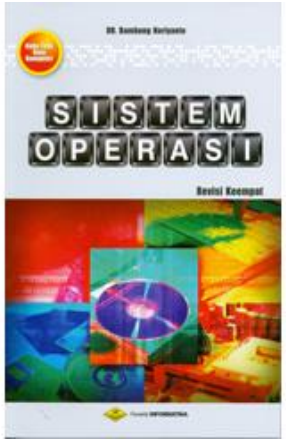
Penjadwalan (Schedulling)

- Pada komputer multiprogramming, sejumlah proses/ thread yg berada dalam **ready state** berkompetisi mendapatkan CPU.
- Seketika CPU tidak bekerja, pilihan proses mana yang **run** berikutnya harus dibuat.
- Bagian dari SO yang membuat pilihan tersebut disebut Penjadwal (**Scheduler**), dan algoritma yang digunakan disebut Algoritma Penjadwalan (**scheduling algorithm)**

Agenda

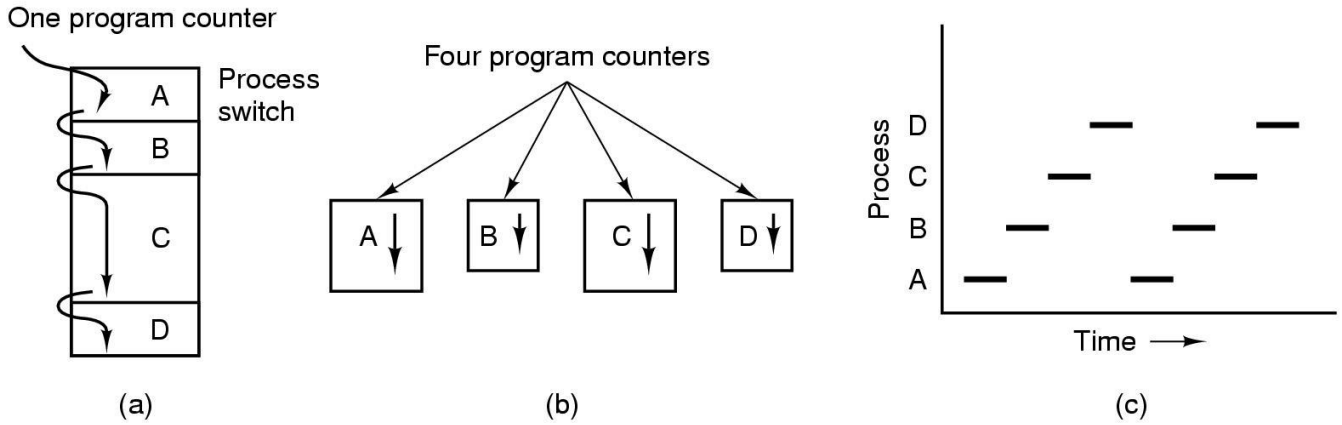
- Introduction to Schedulling
  1. Process Behaviour
  2. When to Schedule
  3. Categories of Schedulling Algorithm
  4. Schedulling Algorithm Goals
- Schedulling in batch system
- Schedulling in Interactive system
- Schedulling in realtime system

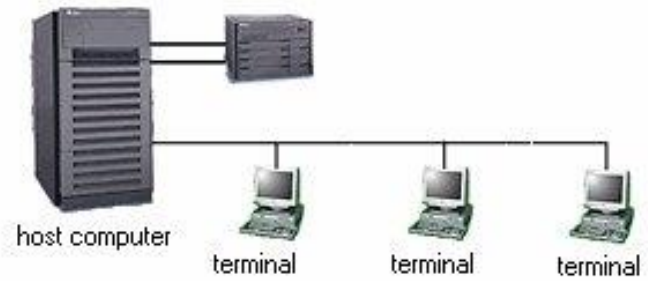




Deskripsi Penjadwalan (1)

- Penjadwalan proses merupakan basis bagi operasi dalam sistem multiprogramming



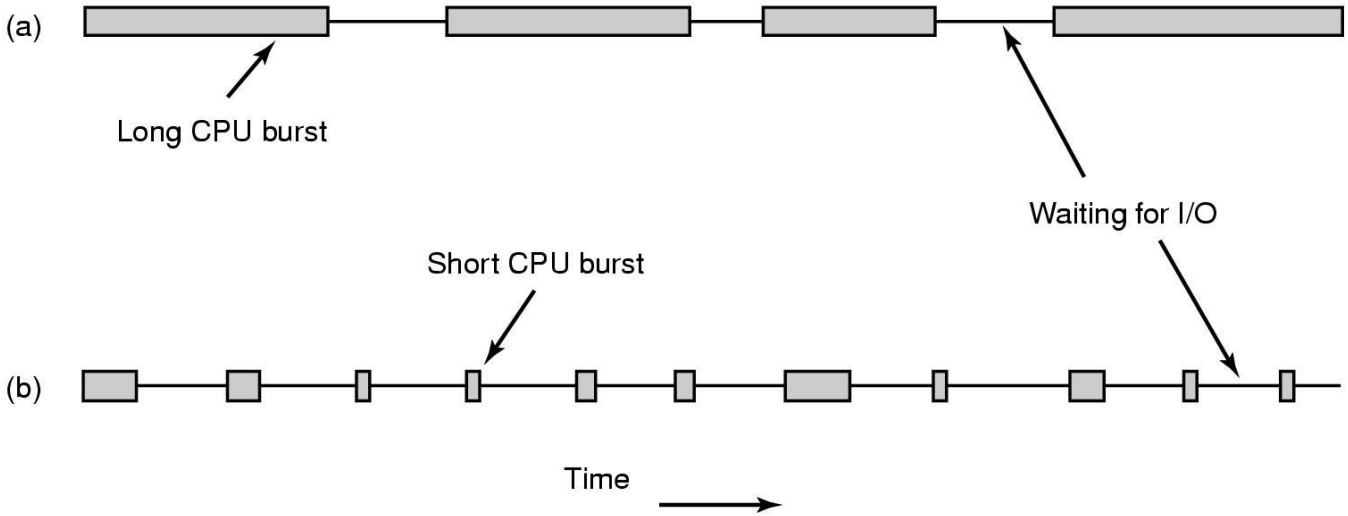


(d) Interactive - timesharing

Deskripsi Penjadwalan (2)

- Kumpulan kebijakan dan mekanisme di SO dalam mengatur urutan kerja yg dilakukan sistem komputer
- "Penjadwal" bertugas memutuskan :
  - Proses mana yg harus berjalan
  - Kapan dan selama berapa lama proses berjalan
- Ingat Diagram 3 keadaan proses!

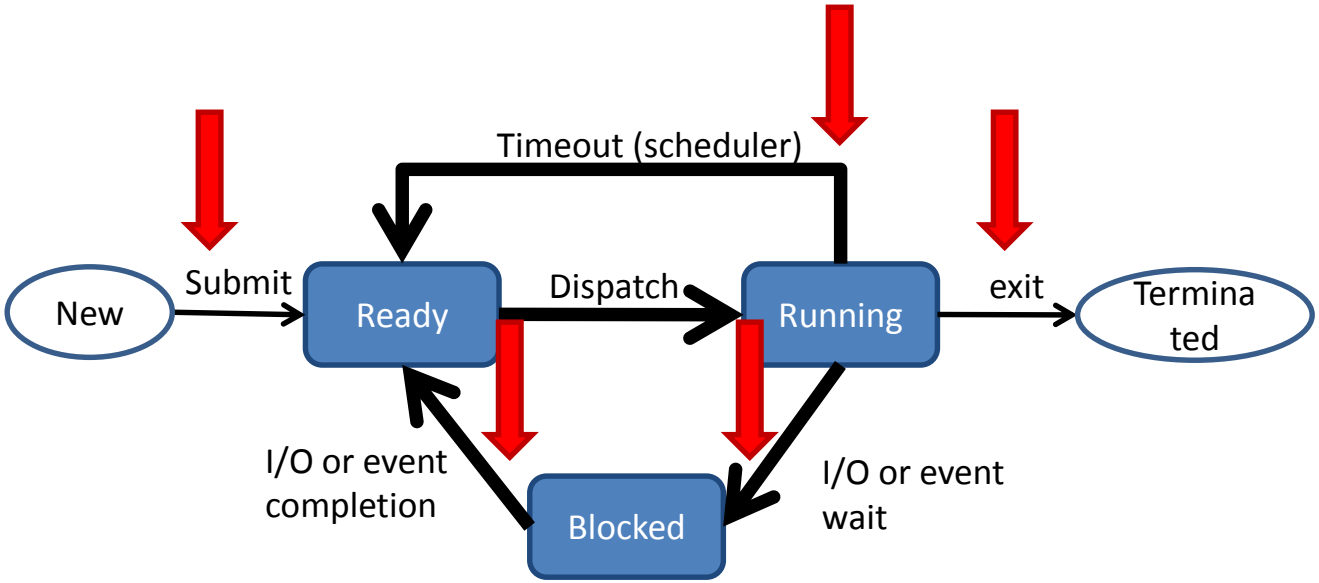
1. Karakteristik Proses



- Salah satu parameter penting bagi algoritma penjadwalan adalah karakteristik proses-proses yang akan di eksekusi
- Karakteristik proses berdasarkan lama pemakaian CPU dan I/O :
  a) Proses dengan tipe: **CPU-bound**
  b) Proses dengan tipe: **I/O bound**
- "Prioritaskan" I/O bound untuk mengoptimalkan utilisasi Sistem

2. Terjadinya Penjadwalan

1. Process creation
2. Process termination
3. Blocking system calls
4. I/O interrupt
5. Hardware clock → Preemptive or Non-preemptive ?



Preemptive (1)

- Prosesor belum selesai mengeksekusi suatu proses → dg sifat preemptive, prosesor dpt diambil alih proses lain dg prioritas lebih tinggi
- Proses yang disela beralih: **running → ready**
- Berguna pd sistem yg proses2nya perlu mdpt tanggapan prosesor scr cepat. Misal :
  - Pd sistem real time, krn kehilangan interupsi dapat fatal
  - Pd sistem interaktif timesharing, agar dapat menjamin waktu tanggap yg memadai

Preemptive (2)

- Preemptive bagus tp memberi overhead (krn byk tabel yg harus dikelola)
- Agar efektif, harus mengusahakan banyak proses di memori utama agar dpt segera *running* begitu diperlukan

Non-preemptive

- Begitu proses diberi jatah layanan prosesor maka **tidak dapat diambil alih** oleh proses lain sampai proses itu selesai.
- Eksekusi terhadap proses :

  Hingga proses tsb selesai, atau *blocked* (karena meminta layanan I/O)

3. Kategori Algoritma Penjadwalan

- Baik aplikasi maupun sistem operasi yang berbeda memiliki tujuan dan kebijakan yang berbeda-beda
- Sehingga pengaturan oleh penjadwal **tidak selalu sama** pada semua sistem
- Tiga kategori penjadwalan berdasarkan **lingkungannya** adalah:
  1. Batch.
  2. Interactive.
  3. Real time.

Lingkungan Batch

- Sistem batch masih banyak digunakan misalnya: **payroll**, **inventory**, interest calculation (at banks), claims processing (at insurance companies), dan pekerjaan lainnya yang bersifat periodik
- (User) tidak butuh respon yang cepat → cocok menggunakan Non-preemptive, atau preemptive dengan waktu yang panjang
- Mengurangi peralihan proses → meningkatkan performa
- Algoritma pada lingkungan batch bersifat **umum**, sehingga sering dipertimbangkan untuk diterapkan pada berbagai situasi

Lingkungan Interactive

- Perlu Preemptive:
  - Interactive users
  - Mencegah satu proses mendominasi pemakaian CPU
  - Mencegah satu proses memblokir lainnya karena bug
- Diterapkan pula pada Server → melayani banyak user (jarak jauh) yang meminta layanan cepat

Lingkungan Real-time

- Pada sistem real-time, preemption tidak diperlukan karena umumnya proses sadar bahwa mereka bekerja dan blok secara cepat
- Sistem real-time hanya menjalankan program tertentu untuk satu tujuan, sedangkan
- Lingkungan interaktif, bersifat general purpose sehingga dapat menjalankan berbagai jenis program yang harus diatur agar tidak mengganggu lainnya

4. Tujuan Yang Ingin dicapai Oleh Sistem Penjadwalan

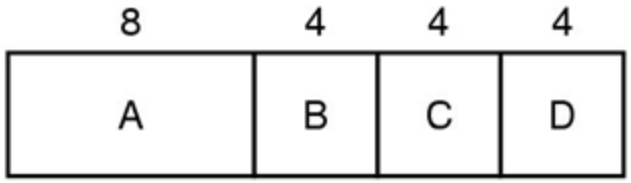
- Pada semua sistem
  - **Fairness** – semua proses mendapat layanan (CPU)
  - **Policy enforcement** – memastikan kebijakan yang diterapkan berjalan
  - **Balance** – mengupayakan semua bagian sistem dalam keadaan sibuk
- Pada sistem batch
  - **Throughput** – memaksimalkan jumlah job per jam (jobs per hour)
  - **Turnaround time** – meminimalkan waktu dari mulai (submission) hingga berakhir (termination)
  - **CPU utilization** – mengupayakan CPU selalu sibuk sepanjang waktu
- Pada sistem interaktif
  - **Response time** – merespon permintaan (request) dengan cepat
  - **Proportionality** – memenuhi harapan pengguna (user's expectation)
- Pada sistem real-time
  - **Meeting deadlines** – menghindari kehilangan data
  - **Predictablility** – menghindari penurunana kualitas dalam sistem multimedia

Agenda

- Introduction to Schedulling
- **Schedulling in batch system**
  1. FCFS (first Come First Serve)
  2. SJF (shortest job first)
  3. SRF (shortest remaining first)
- Schedulling in Interactive system
- Schedulling in realtime system

1- First Come First Serve (FCFS)

- Non-preemptive
- Tidak berprioritas
- Ketentuan :
  - Proses2 diberi jatah waktu prosesor diurutkan berdasarkan waktu kedatangan proses2 itu ke sistem
  - Begitu proses mendapat jatah waktu prosesor, proses dijalankan sampai selesai
- Penjadwalan ini dikatakan adil dalam arti harfiah,
- tapi dinyatakan tidak adil krn proses-proses yg perlu waktu lama membuat proses2 pendek menunggu.
- Proses2 tidak penting dapat membuat proses2 penting menunggu.



FCFS

- FCFS jarang digunakan mandiri,
- Dapat dikombinasikan dg skema yg lain, mis:
  - Pada seleksi awal dilakukan berdasarkan nilai prioritas proses, sedangkan
  - Untuk proses2 yang berprioritas sama akan diputuskan menggunakan FCFS

FCFS
Evaluasi

- Mengacu kriteria penilaian penjadwalan:
  - Fairness → Adil dalam arti resmi
  - Efisiensi → Sangat efisien dlm penggunaan prosesor, karena tidak ada *switching process*
  - Response time → Sangat tidak memuaskan krn proses bisa menunggu lama, tidak cocok utk sistem interaktif.
  - Turn around time → Tidak bagus
  - Throughput → Tidak bagus

FCFS

Penggunaan

- Cocok untuk sistem batch yg sangat jarang melakukan interaksi dg pemakai scr langsung.
  - Contoh : aplikasi analisis numerik, pembuatan tabel
- Tidak berguna untuk sistem interaktif krn tidak memberi **waktu tanggap** yg bagus
- Tidak dapat digunakan untuk sistem *realtime*
- Contoh FCFS

2-Shortest Job First (SJF)

- Non-preemptive
- Serasa berprioritas → prosesor dialokasikan ke proses berprioritas tertinggi. Proses2 dg prioritas yg sama akan dijadwalkan scr FCFS
- Penjadwalan ini mengasumsikan waktu lamanya proses diketahui sebelumnya (dapat diramalkan).
- Mekanisme SJF :

  mendahulukan proses dg **waktu jalan terpendek** sampai selesai, setelah proses itu selesai, maka proses dg waktu jalan terpendek berikutnya dijadwalkan. Dst.

Shortest Job First

- (+) efisiensi tinggi, turn arround time rendah
- Bandingkan (asumsi waktu kedatangan bersamaan):

Cara I → FCFS

A	B	C	D
8	7	6	5

Cara II → SJF

D	C	B	A
5	6	7	8

Proses	Waiting time	Waiting time
	FCFS	SJF
A	0	18
B	8	11
C	15	5
D	21	0
Rata-rata	**11**	**8,5**

Proses	turn arround	turn arround
	FCFS	SJF
A	8	26
B	15	18
C	21	11
D	26	5
Rata-rata	**17,5**	**15**

Shortest Job First

- Masalah :
  - Tidak dapat mengetahui ukuran proses saat proses masuk → asumsi/ pendekatan → perilaku historis sistem
  - Proses yang tidak datang bersamaan, shg penetapan prioritasnya hrs dinamis

- Penggunaan
  - Sangat jarang digunakan, mrpk kajian teoritis utk perbandingan **TA time**
- Contoh SJF

3-Shortest Remaining-Time First (SRF)

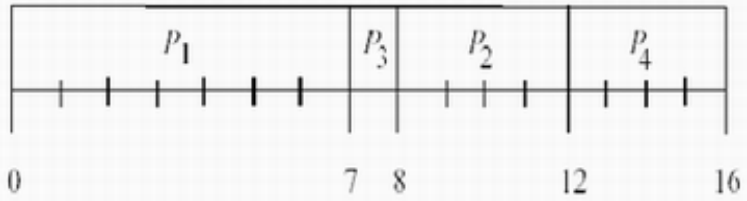
- Preemptive, dinamis
- Perbaikan dari SJF
  - SJF → non preemptive
  - SRF → preemptive → dpt digunakan utk sistem timesharing
- Ketentuan :
  - Mengestimasi waktu proses terendah untuk dijalankan, tmsk proses2 yg baru tiba
- SJF vs SRF :
  - SJF → begitu proses dieksekusi, proses jalan sampai selesai
  - SRF → proses yg Running dpt diambil alih oleh proses baru (yg sisa waktu jalan lbh rendah)

SJF VS SRF

Non-Preemptive VS Preemptive

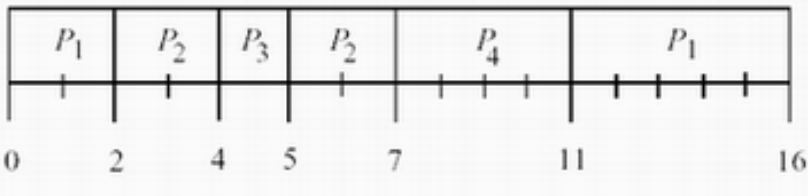
Process	Arrival Time	Burst Time
P_1	0.0	7
P_2	2.0	4
P_3	4.0	1
P_4	5.0	4

- SJF (non-preemptive)



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

- SRTF (preemptive)



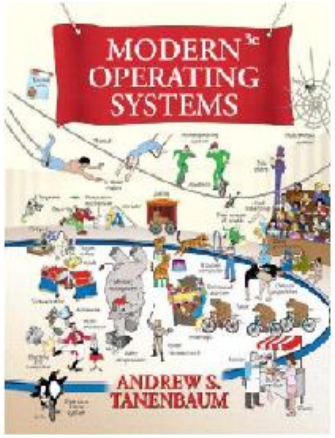
Average waiting time = (9 + 1 + 0 + 2)/4 = 3

SRF

- Kelemahan:
  - Overhead SRF > Overhead SJF
  - SRF butuh penyimpanan waktu layanan yg telah dihabiskan proses
  - Proses2 kecil yg tiba akan segera dijalankan
  - Proses2 lama, berarti lama dan variasi waktu tunggu lebih besar dibanding SJF
- Secara teoritis SRF memberi waktu tunggu minimum
- Contoh SRF

Agenda

- Introduction to Schedulling
- Schedulling in batch system
- **Schedulling in Interactive system**
  1. RR (round robin)
  2. PS (priority schedulling)
  3. MFQ (multiple feed back queues)
  4. HRN (highest ratio next)
  5. GS (Guaranteed Schedulling)
- Schedulling in realtime system



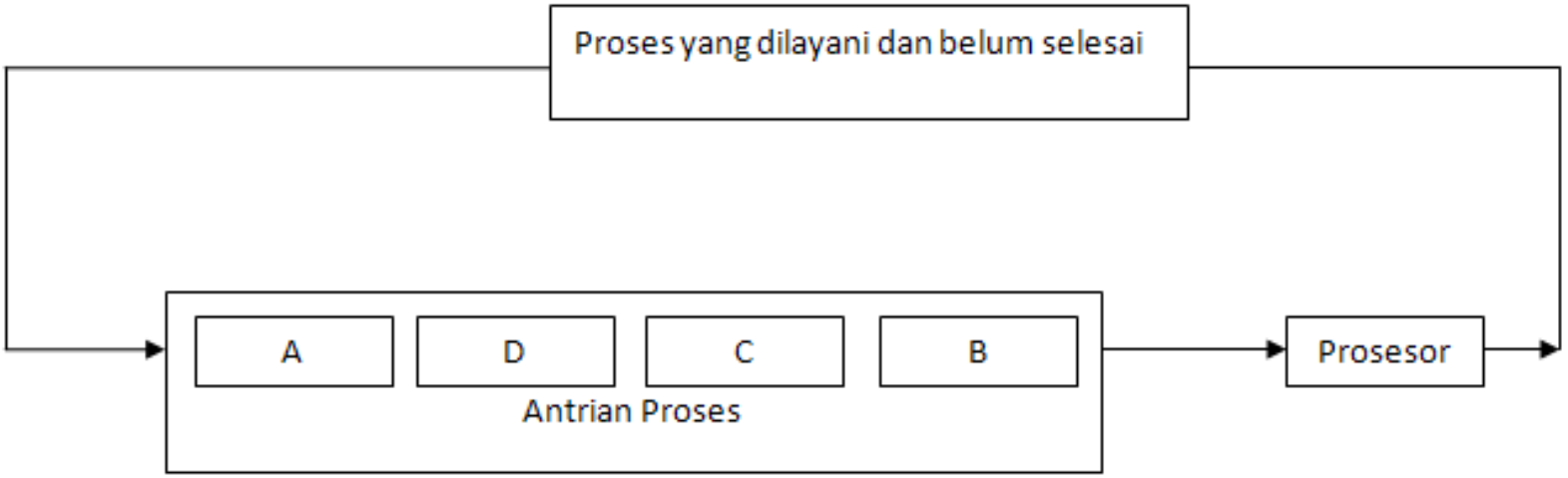
1. Round Robin (RR)

- Preempt-by-time
- Penjadwalan tanpa prioritas
- Semua proses dianggap penting dan diberi sejumlah waktu prosesor yg disebut kwanta (quantum).

Round robin

- Aturan:
  1. Jika kwanta habis dan proses belum selesai maka proses Running itu menjadi Ready (runnable) dan prosesor dialihkan ke proses lain
  2. Jika kwanta belum habis dan proses menunggu suatu kejadian, maka proses Running itu mjd Blocked dan prosesor dialihkan ke proses lain
  3. Jika kwanta belum habis tapi proses telah selesai maka proses Running itu diakhiri dan pemroses dialihkan ke proses lain.

Round robin
- Implementasi



- Sistem mengelola senarai proses Ready (runnable) sesuai urutan kedatangan
- Sistem mengambil proses yang berada di ujung depan antrian Ready mjd Running
- Bila kwanta belum habis dan proses selesai maka sistem mengambil proses di ujung depan antrian proses Ready.
- Jika kwanta habis dan proses belum selesai maka tempatkan proses Running ke ekor antrian proses Ready dan sistem mengambil proses di ujung depan antrian proses Ready

Round Robin
Permasalahan

- Penentuan besar kwanta:
  - Kwanta terlalu besar menyebabkan waktu tanggap besar dan turn arround time tinggi.
  - Kwanta terlalu kecil mengakibatkan peralihan proses (*switching time*) terlalu banyak sehingga menurunkan efisiensi prosesor
  - Contoh :
    - 1 ms switching time, 4 ms kuantum → 20% waktu terbuang
      - Tidak efisien, terlalu banyak waktu terbuang untuk process switching
    - 1 ms switching time, 100 ms kuantum → 1% waktu terbuang
      - Namun tidak nyaman krn bila ada 10 user, maka user terakhir dilayani setelah 1000 ms !!
- Besar kwanta waktu yg optimal hrs ditetapkan berdasarkan kebutuhan sistem, terutama dari hasil percobaan atau data historis sistem.

Round Robin

Evaluasi

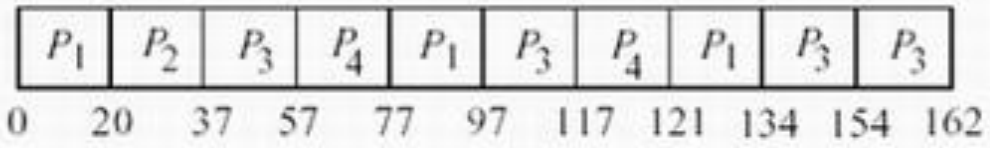
- Mengacu kriteria penilaian penjadwalan
  - Fairness
    - Dipandang adil → dari kesamaan layanan oleh prosesor
  - Efisiensi
    - Cenderung efisien bagi sistem interaktif
  - Response time
    - Memuaskan utk sistem interaktif, tidak memadai utk sistem real time
  - Turn around time
    - Cukup bagus
  - Throughput
    - Cukup bagus

Round Robin

Contoh

Process	Burst Time
P_1	53
P_2	17
P_3	68
P_4	24

- The Gantt chart is:



- Waktu kuanta = 20

Round robin

Penggunaan

- Cocok untuk sistem interactive-timesharing (krn sebagian besar waktu dipergunakan untuk menunggu kejadian eksternal).
  - Contoh text-editor, kebanyakan waktu program adalah utk menunggu kejadian dari keyboard shg prosesor dapat digunakan untuk mengeksekusi proses2 lain.
- Tidak cocok untuk sistem waktu nyata

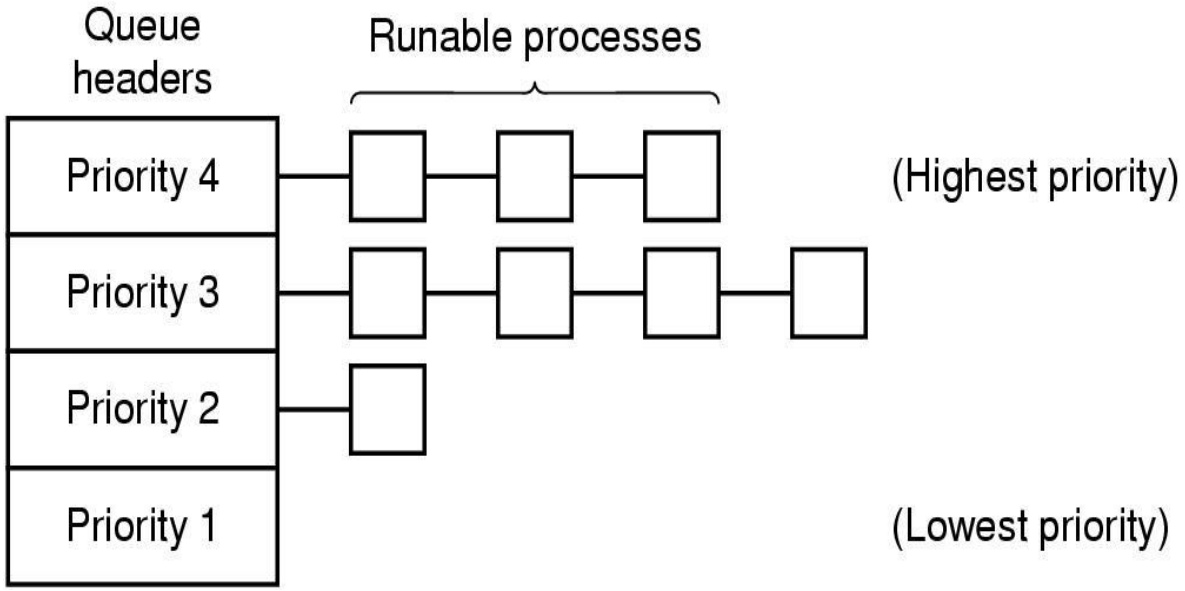
2. Penjadwalan Berprioritas

- Masing-masing proses diberi prioritas dan proses yg berprioritas tertinggi mjd **Running**
- Prioritas dpt diberikan secara:
  - **Statis**: prioritas tidak berubah
    - (+) mudah diimplementasikan
    - (+) overhead relatif kecil
    - (-) tidak tanggap perubahan lingkungan yg menghendaki penyesuaian prioritas
  - **Dinamis**: mekanisme menanggapi perubahan lingkungan sistem saat beroperasi di lingkungan nyata.
    - Prioritas awal yang diberikan ke proses mungkin hanya berumur pendek → sistem dapat menyesuaikan nilai prioritasnya ke nilai yg lebih tepat sesuai lingkungan.
    - (+) response time bagus
    - (-) implementasinya lebih kompleks dan overhead besar

Contoh Dinamis

- Pada kasus *I/O bound process*
  - Dlm rangka memenuhi tujuan sistem komputer → mis. Memberi prioritas tinggi pada I/O bound process
  - Waktu tunggu (blocked), CPU dipakai oleh proses2 lainnya
  - Memori tidak perlu menyimpan I/O bound process
- **Rumus 1/f**, dg f adalah rasio kwanta terakhir yg digunakan proses, contoh:
  - Proses yg hanya menggunakan 2 ms kwanta 100 ms maka prioritasnya 50
  - Proses yg telah berjalan selama 50 ms sblm blocked berprioritas 2
  - Proses yg menggunakan seluruh kwanta berprioritas 1.
  - Prioritas : 50 > 2 > 1

Kombinasi: Prioritas - RR



- Dg mengelompokkan proses2 kedalam kelas2 prioritas.
- Penjadwalan berprioritas diterapkan antar kelas2 proses2 tsb, penjadwalan RR atau FCFS pd proses2 di dlm suatu kelas
- Alokasi prioritas dpt menggunakan statis dan dinamis, misal 1/f
- Prioritas harus di update secara berkala agar prioritas rendah tidak mengalami *starvation*

3. Multiple Feedback queues (MFQ)

- Bahwa memberi kuanta yang besar pada CPU-bound process lebih menguntungkan krn kuanta yang kecil berarti menyebabkan banyak swapping → akses ke disk = lambat
- Namun kuanta besar berdampak buruk pada response time
- Sasaran : mencegah aktivitas swapping yg berlebih
- Solusi : membuat kelas-kelas prioritas

MFQ (2)

- Preemptive , dinamis
- Algoritma:
  1. Proses2 yg banyak memakai prosesor diberi alokasi waktu (kuantum) lebih banyak
  2. Memberi prioritas : kelas tertinggi 1 kwanta, kmdn 2 kwanta, 4 kwanta, 8 kwanta, dst
- Aturan:
  - Jalankan proses2 yg berada pd kelas prioritas tertinggi
  - Jika proses telah menggunakan seluruh kwanta yg dialokasikan maka proses itu diturunkan kelas prioritasnya
  - Proses yg masuk utk pertama kali ke sistem langsung diberi kelas tertinggi

MFQ (3)

contoh

- Suatu proses memerlukan proses komputasi kontinyu selama 100 kuanta
- Dengan round robin 1 kuanta → 100 swap
- Dengan MFQ → 7 swap
  - Kelas 1 : kwanta = 1 : sisa 99
  - Kelas 2 : kwanta = 2 : sisa 97
  - Kelas 3 : kwanta = 4 : sisa 93
  - Kelas 4 : kwanta = 8 : sisa 85
  - Kelas 5 : kwanta = 16 : sisa 69
  - Kelas 6 : kwanta = 32 : sisa 37
  - Kelas 7 : kwanta = 64 : 0

MFQ (4)

- (+) Mampu mencegah:
  - Swapping berulang kali thd proses yg lama
  - Proses2 interaktif singkat menunggu terlalu lama

- Penggunaan cocok untuk:
  - Sistem dg proses2 yg banyak proses lambat, memerlukan waktu lama, dan banyak proses singkat

4. Highest-Ratio Next (HRN)

- Non preemptive, dinamis
- Terinspirasi dari SJF, untuk digunakan pada lingkungan interaktif
- Prioritas proses berdasarkan fungsi waktu proses (burst time) dan jumlah waktu tunggu proses
- Rumus:

  Prioritas = $(t_{tunggu} + t_{proses})/ t_{proses}$

- Sasaran → memperoleh waktu tanggap yg baik
- HRN mendahulukan proses pendek, namun prioritas proses panjang akan turut meningkat melalui peningkatan waktu tunggunya
- Contoh HRN

5. Guaranteed Schedulling (GS)

- Preemptive, dinamis
- Berupaya memberi masing2 pemakai daya prosesor yg sama → menjanjikan tiap user mdpt jatah yg sama (adil)
- Contoh:

  Jika ada N pemakai, masing2 mdpt 1/N daya, maka
  - Dilakukan pemeriksaan berkala thd berapa banyak tiap proses telah mengkonsumsi CPU (T).
  - T kmdn dibandingkan dg jatah yg dialokasikan (1/N) → **T : 1/N**

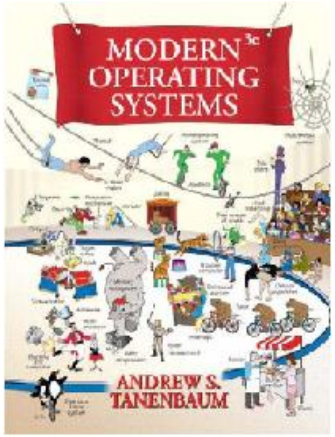
    Contoh:

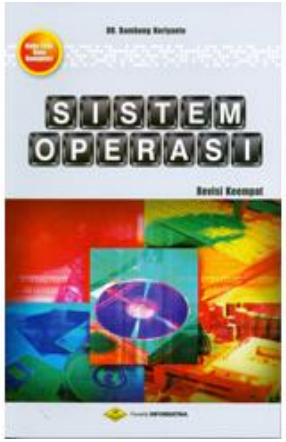
    0,5 → baru memakai separuh jatahnya

    2 → sudah melebihi 2x porsi jatahnya
  - Kemudian, penjadwal **menjalankan proses dg rasio terendah hingga rasio proses ini melampaui kompetitor terdekatnya**

Agenda

- Introduction to Schedulling
- Schedulling in batch system
- Schedulling in Interactive system
- **Schedulling in realtime system**





Real time system (1)

- Dibahas lebih lanjut pada kajian real-time multimedia system (Chapter 7. MOS. Tanenbaum)
- Sistem real-time tidak mentolerir keterlambatan dalam merespon
- Contoh: memutar lagu dari CD, pemantauan pasien UGD, autopilot pada pesawat, robot pada pabrik otomatis
- Tipe sistem real-time:
  - hard real-time
    Harus memenuhi absolute deadline
  - soft real-time
    Ada toleransi tertentu terhadap deadline

Latihan 1

Dengan **FCFS**, buatlah Gantt Chart dan hitung Response time serta turn around time nya

Nama Proses	Waktu tiba	Lama eksekusi
A	0	5
B	0	2
C	0	6
D	0	8
E	0	3

Nama Proses	Waktu tiba	Lama eksekusi
A	0	5
B	1	2
C	2	6
D	2	8
E	5	3

Hint:

Nama Proses	Waktu tiba	Lama eksekusi	Waktu Tunggu	Mulai Eksekusi	Selesai Eksekusi	Turn Around TIme

Latihan 2

Dengan **SJF**, buatlah Gantt Chart dan hitung Response time serta turn around time nya

Nama Proses	Waktu tiba	Lama eksekusi
A	0	10
B	0	5
C	0	7
D	0	1
E	0	3

Nama Proses	Waktu tiba	Lama eksekusi
A	9	10
B	5	5
C	7	7
D	0	1
E	2	3

Latihan 3

Dengan **SRF**, buatlah Gantt Chart dan hitung Response time serta turn around time nya

Nama Proses	Waktu tiba	Lama eksekusi
A	0	7
B	2	3
C	4	9
D	5	4

Latihan 4

Dengan **RR**, buatlah Gantt Chart dan hitung Response time serta turn around time nya

Nama Proses	Waktu tiba	Lama eksekusi
A	0	7
B	0	5
C	0	8
D	0	2
E	0	6
Kuantum (Q):	3	

Nama Proses	Waktu tiba	Lama eksekusi
A	0	5
B	1	3
C	5	7
D	6	1
E	7	6
Kuantum (Q):	2	

Latihan 5

Diberikan beberapa proses dibawah ini dengan panjang CPU burst (dalam milidetik). Semua proses diasumsikan datang pada saat t=0 (datang bersamaan)

Proses	Burst Time	Prioritas
P1	10	3
P2	1	1
P3	2	3
P4	1	4
P5	5	2

a. Gambarkan gantt chart 4 eksekusi dari proses-proses tersebut menggunakan FCFS, SJF, prioritas nonpreemptive, dan round robin dengan kuanta 2.
b. Hitung **troughput** (hingga t = 10) dan rata-rata **respon time** dari tiap algoritma yang digunakan
c. Berdasarkan b, algoritma mana yang terbaik untuk lingkungan batch?
d. Berdasarkan b, algoritma mana yang terbaik untuk lingkungan interaktif?

Review

1. Lima proses tiba secara bersamaan pada saat t=0 dengan urutan P1, P2, P3, P4, dan P5. Bandingkan (rata-rata) *turn-around time* dan *response time* dari ke lima proses tersebut jika mengimplementasikan algoritma penjadwalan FCFS, SJF, dan RR (*Round Robin*) dengan kuantum 4 (empat) satuan waktu. *Burst time* kelima proses tersebut berturut-turut (10, 8, 6, 4, 2) satuan waktu.
2. Bila diketahui kelima proses (pada soal 1) tiba berturut-turut pada saat (0, 2, 3, 5, 9) ms, gunakan algoritma *Shortest Remaining Time First* (SRF) untuk mengilustrasi *time-line* (gant chart) eksekusinya. Hitunglah rata-rata *turn-around time* nya.
3. Berikan analisa anda berdasarkan b dan c, mana yang paling baik untuk sistem batch, dan untuk sistem interaktif?