Sistem Operasi

Penjadwalan Proses

2016

UNIVERSITAS Telkom

Outline

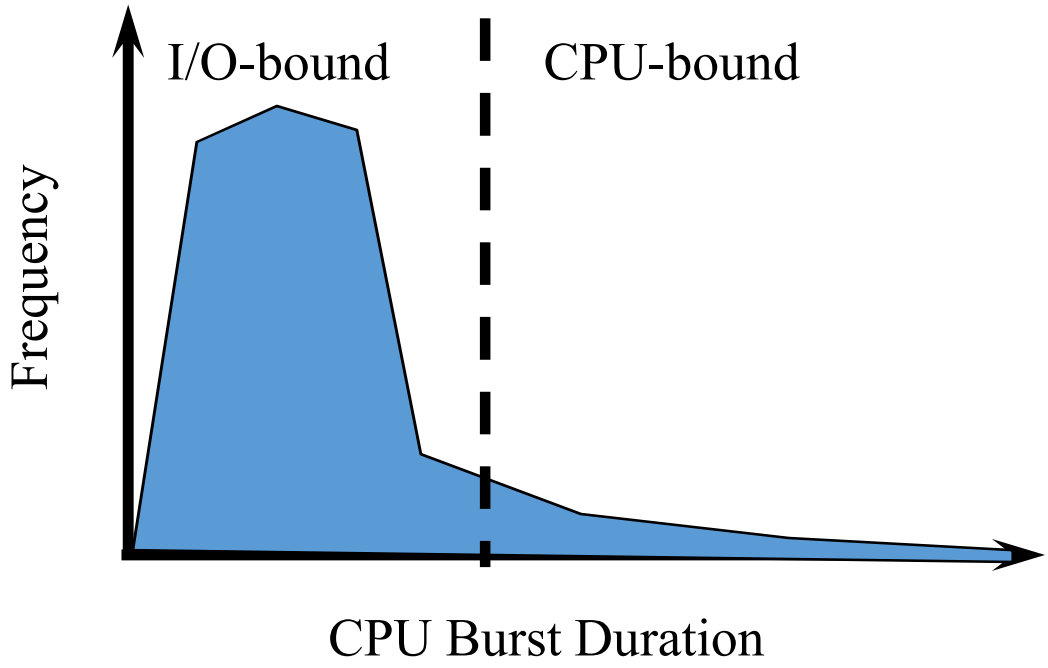
- Objektif
- Kriteria Penjadwalan
- Algorithma
- Contoh

Objektif

- **Memaksimalkan utilisasi CPU**
  - Beberapa proses run sepanjang waktu
- Sebuah proses dieksekusi sampai dia harus menunggu
  - Biasanya untuk permintaan I/O
  - CPU idle!
- Multiprogramming
  - Ketika sebuah proses menunggu, CPU dialokasikan ke proses lain.

Siklus Burst CPU-IO

```
add
read
(I/O Wait)
store
increment
write
(I/O Wait)
```



Penjadwalan Proses

- Ketika menjalankan scheduler
  1 Process create
  2 Process exit
  3 Process blocks
  4 I/O interrupt
  5 Clock interrupt
- Non-preemptive – proses jalan sampai di blok atau menyerahkan CPU (1,2,3,4)
- Preemptive – proses jalan untuk unit waktu tertentu, kemudian scheduler memilih sebuah proses untuk jalan (1-5)

Penjadwalan Proses

- Nonpreemptive
  - Ketika sebuah proses berjalan distate running, proses tersebut akan tetap berjalan hingga selesai atau memblok dirinya sendiri untuk I/O
- Preemptive
  - Proses yang sedang berjalan dapat diinterupsi dan dipindahkan ke state Ready oleh OS
  - Mengizinkan untuk mendapatkan pelayanan yang lebih baik yang dikarenakan tidak ada proses yang dapat memonopoli prosesor dengan waktu yang sangat lama

Tujuan Penjadwalan

- Semua Sistem
  - Keadilan Proses (Fairness)
  - Penerapan kebijakan (Policy enforcement)
  - Keseimbangan (Balance for system part))
- Sistem Batch
  - Throughput
  - Waktu turnaround
  - Utilisasi CPU
- Sistem Interaktif
  - Waktu Respon
  - Proporsionalitas
- Sistem Real-time
  - Pemenuhan deadline
  - Keterkiraan (predicability)

Pengukuran Penjadwalan untuk Performansi

1 Utilitas CPU (40 ke 90)

2 Throughput (proses / jam)

3 Waktu turn-around

4 Waktu tunggu (dalam antrian)

- Maksimalkan #1, #2 Minimalkan #3, #4
- Realitas – Tidak ada kebijakan universal
  - Banyak model
  - Tentukan apa yang dioptimasi - metrik
  - Pilih yang cocok dan atur berdasarkan pengalaman

First-Come, First-Served (Batch)

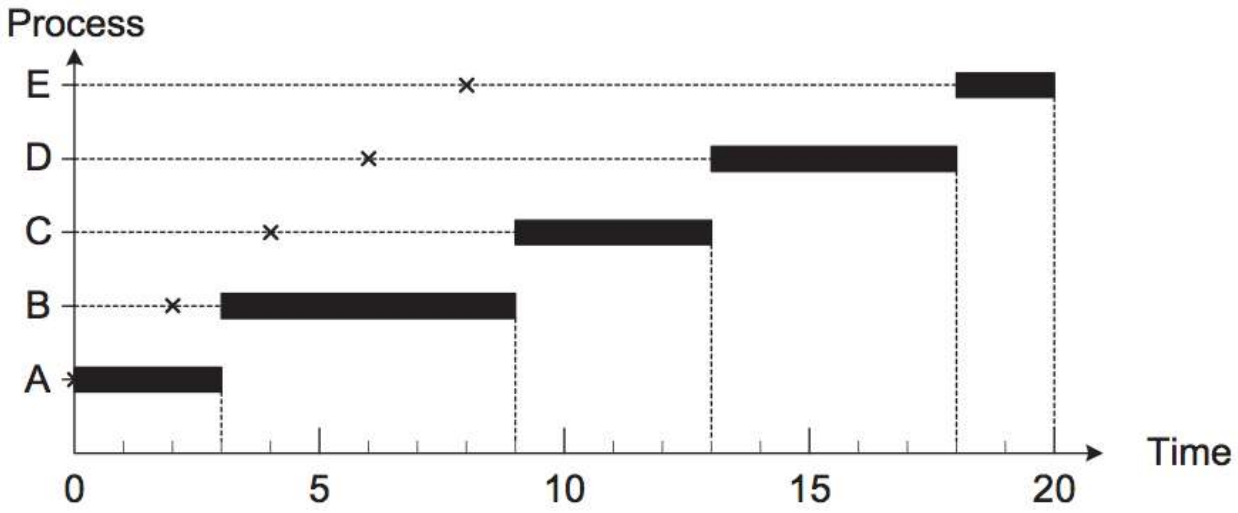
Proses	Waktu Burst
A	8
B	1
C	1

Gantt Chart

- Waktu tunggu rata-rata (AWT) (0 + 8 + 9) / 3 = 5.7
- Waktu turn-around rata-rata (ATT) (8 + 9 + 10) / 3 = 9

First-Come, First-Served (Batch)

Proses	Arrival Time	Service Time
A	0	3
B	2	6
C	4	4
D	6	5
E	8	2



First-Come, First-Served (Batch)

- Setiap proses masuk ke antrian Ready
- Ketika proses selanjutnya akan dieksekusi, dipilihlah proses yang telah menunggu di antrian yang paling lama.
- Proses yang singkat membutuhkan waktu yang lama untuk dieksekusi
- Proses I/O harus menunggu sampai proses yang lain selesai dikerjakan X_X

Shortest Job First (Batch)

Proses	Waktu Burst
A	8
B	1
C	1

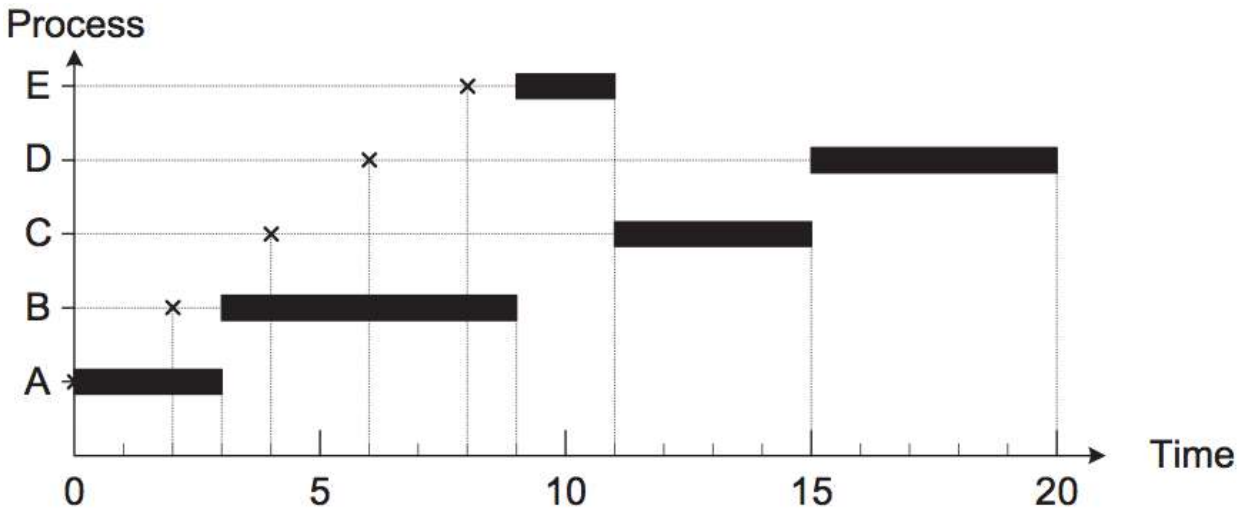
- Waktu tunggu rata-rata (0 + 1 + 2) / 3 = 1
- Waktu tunggu rata-rata optimal?
- Waktu turn-around rata-rata (1 + 2 + 10)/ 3 = 4.3

Shortest Job First (Batch)

- Nonpreemptive
- Proses yang dikerjakan yang lebih cepat diselesaikan
- Proses yang lebih singkat dapat meloncati proses yang lebih lama
- Jika waktu estimasi untuk sebuah proses tidak tepat, OS dapat membatalkannya
- Kemungkinan terjadi ‘starvation’ untuk proses yang lama

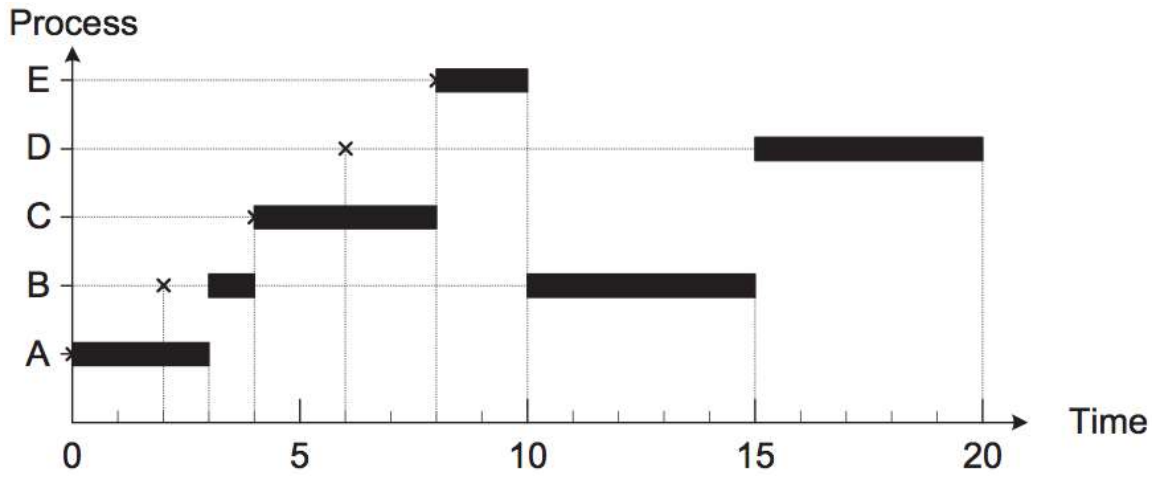
Shortest Job First (Batch)

Proses	Arrival Time	Service Time
A	0	3
B	2	6
C	4	4
D	6	5
E	8	2



Shortest Remaining Time

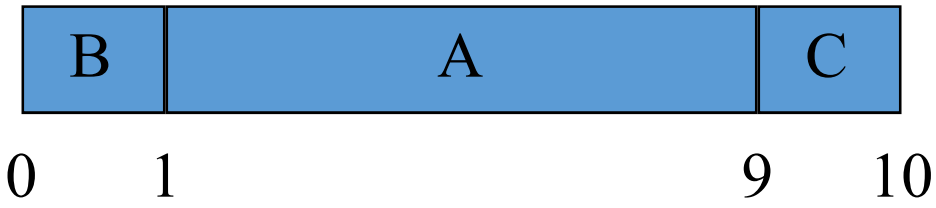
- SJF versi preemptive
- Harus mengestimasi waktu proses



Penjadwalan Prioritas (Interaktif)

- SJF merupakan kasus khusus

Proses	Waktu Burst	Prioritas
A	8	2
B	1	1
C	1	3

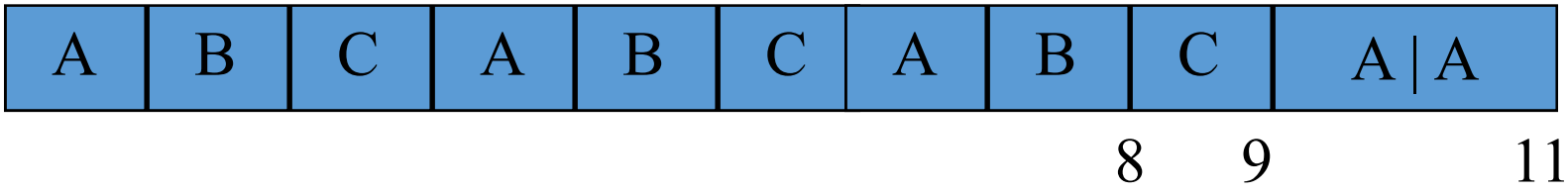


- AWT $(0 + 1 + 9) / 3 = 3.3$
- ATT $(1 + 9 + 10) / 3 = 6.7$

Round Robin (Interaktif)

- Slot waktu tetap dan Preemptiv

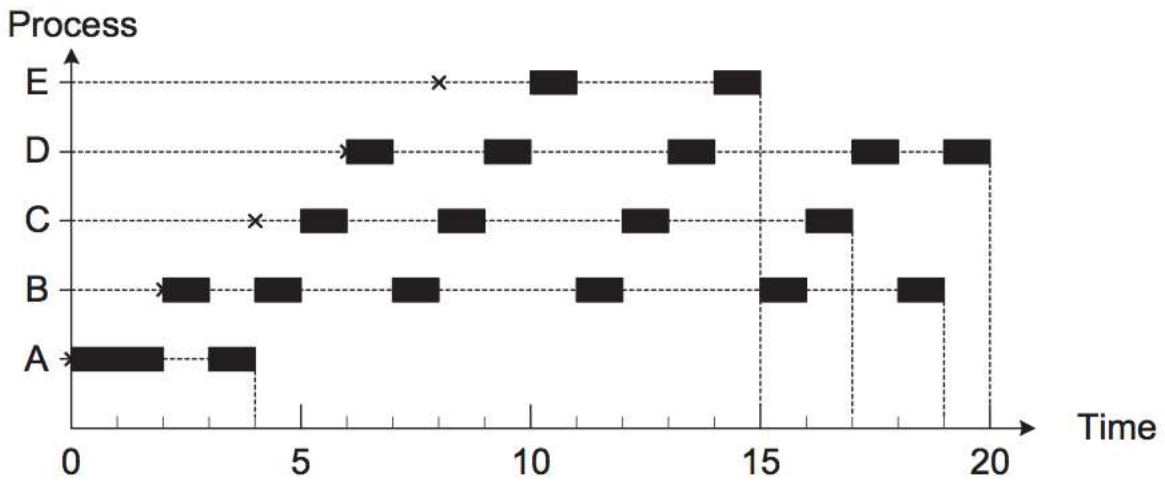
Proses	Waktu Burst
A	5
B	3
C	3



- AWT = (0 + 1 + 2) / 3 = 1
- ATT = (11 + 8 + 9) / 3 = 9.3
- FCFS? SJF?

Round Robin (Interaktif)

Proses	Arrival Time	Service Time
A	0	3
B	2	6
C	4	4
D	6	5
E	8	2



Perbadingan algoritma

	RR	FCFS	SJF
AWT	1	4.3	3
ATT	9.3	8	6.7

Round Robin Fun

Proses	Waktu Burst	Arrival Time
A	11	3
B	13	5
C	10	11

- Waktu giliran?
  - q = 10
  - q = 5
  - q = 1
  - q = 0,5 q = 0.25 q = 0.1
- AWT? ATT?

Pengaruh Q

	Q = 10	Q =5	Q = 1	Q = 0.5
AWT	13	8	6,67	3,8
ATT	34,67	34,67	34	34,1
∑ giliran	5	8	34	68

Pengaruh Q

AWT

Q	A	B
1	1	1
2	2	2
3	3	3
4	3.7	4
5	4.3	4

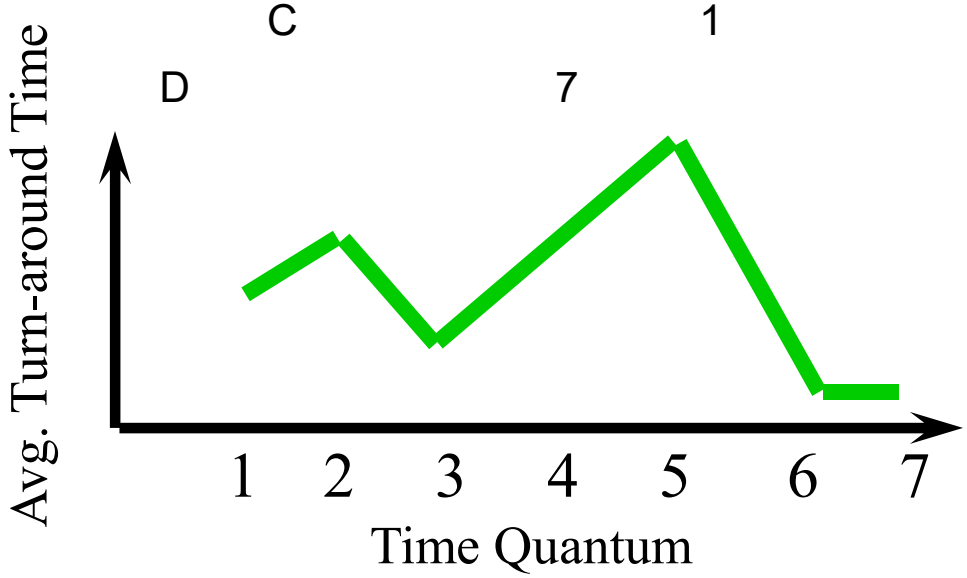
ATT

Q	A	B
1	9.3	10.3
2	10	10.3
3	8.6	10.3
4	9.3	9
5	8	8.3

Q menentukan AWT dan ATT, Q yang lebih tinggi akan menaikkan AWT tetapi akan mengurangi ATT

Round Robin Fun

Proses	Waktu Burst
A	6
B	3
C	1
D	7



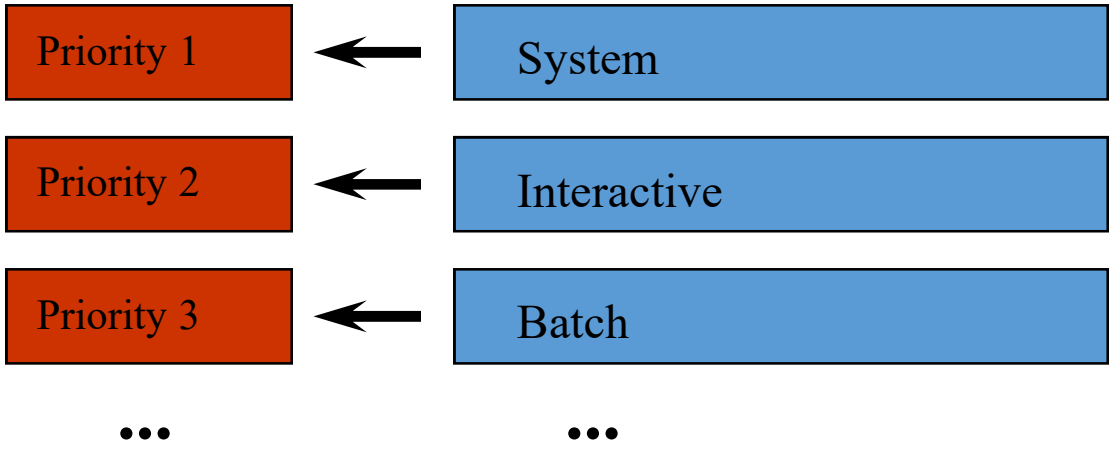
Mencoba Penjadwalan

Proses	Waktu Burst	Prioritas
A	10	2
B	1	1
C	2	3

- Gantt Charts:
  - FCFS
  - SJF
  - Prioritas
  - RR (q=1)
- Performansi:
  - (Throughput)
  - Waktu tunggu
  - Waktu turn-around

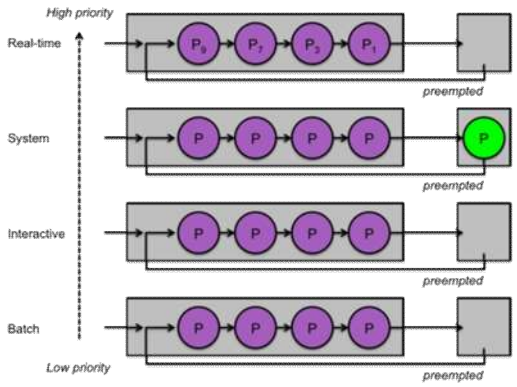
Antrian Multi-Level (Interaktif)

- Kategori proses



- Jalankan di 1 dahulu, lalu 2 …
- Kelaparan (Starvation)??
- Bagi antar antrian: 70% 1, 20% 2 …

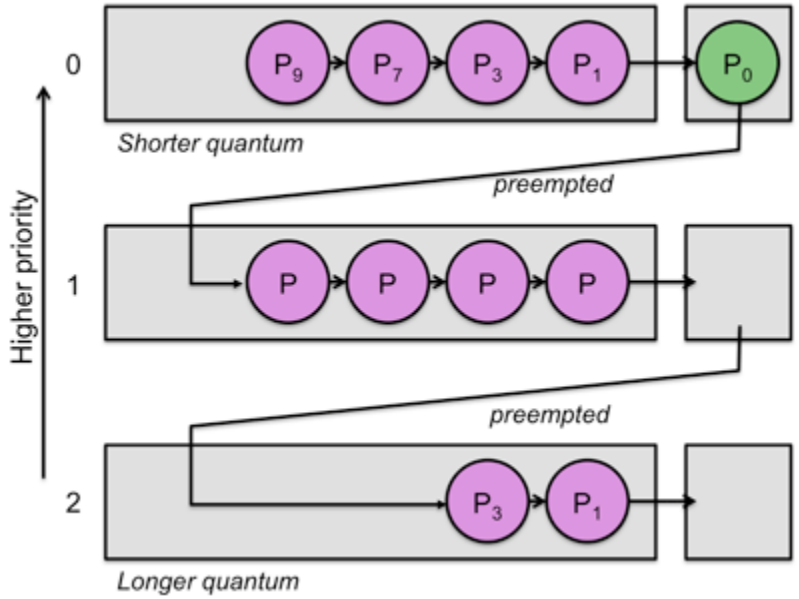
Antrian Multi-Level



- Tiap level algoritmanya bisa berbeda
- Yang umum digunakan per level : Round Robin dengan quantum yang berbeda (optional)
- Low priority, quantum lebih lama
- Biasa digunakan pada top-level schedulling, seperti real-time, kernel threads, interactive, dan background process

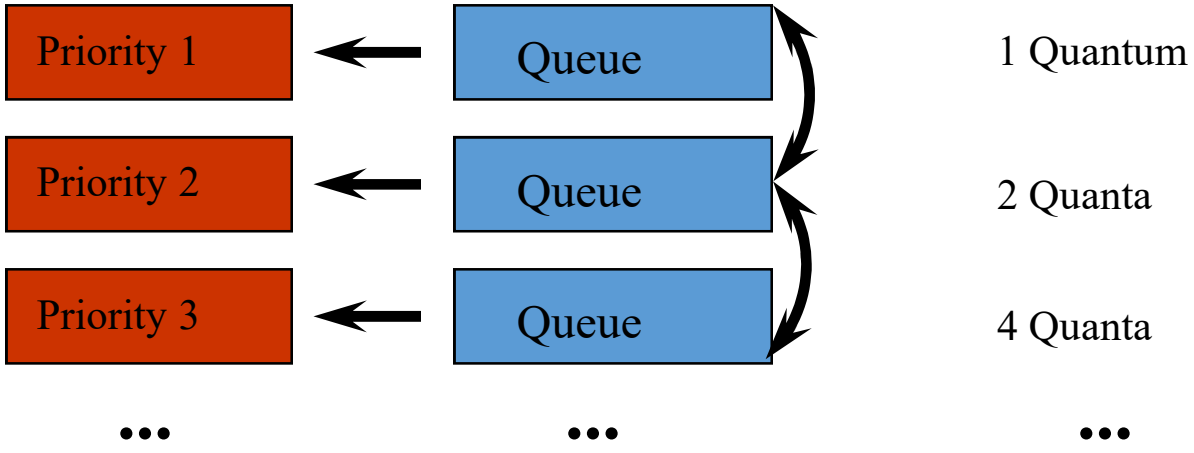
UNIVERSITAS Telkom

Antrian Multi-Level Feedback



Antrian Multi-Level Feedback

- Berbasis slot waktu tapi ingin bisa interaktif



- Bayangkan suatu proses yang memerlukan 100 quanta
  - 1, 2, 4, 8, 16, 32, 64 = 7 swaps!
- Bagaimana dengan interaktifitas untuk pemakai?

Penjadwalan Proses di Linux

- FIFO Real-time
  - Prioritas tertinggi
  - Tidak bisa preemptiv
- Round-Robin Real-time
  - Kedua tertinggi
  - Sama seperti FIFO Real-time kecuali bisa preemptiv
- Timesharing

Prioritas and Quanta

- Prioritas
  - Menentukan thread mana yang dipilih untuk dijalankan
  - Setiap thread diberi prioritas
  - Prioritas berubah sepanjang umur thread
- Quanta
  - Menentukan berapa lama sebuah Thread berjalan ketika dipilih
  - Bervariasi sesuai jenis aplikasi

Penentuan Prioritas

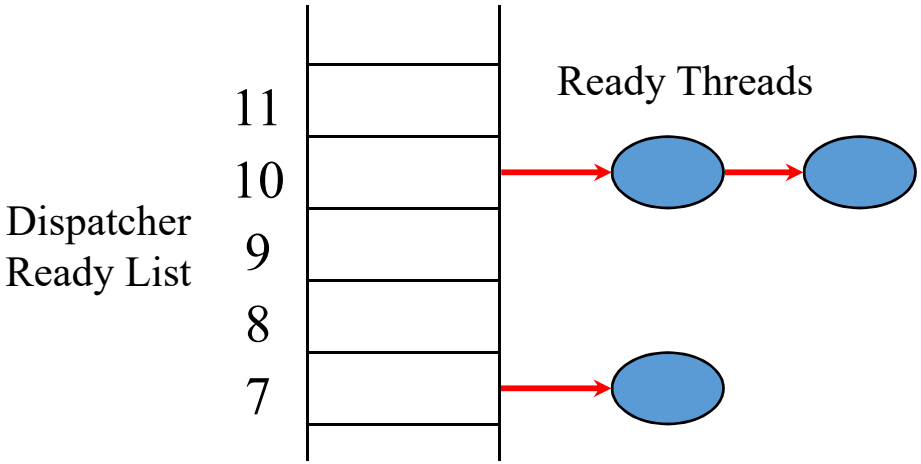
- Kernel Windows menggunakan 32 level prioritas
  - 31 tertinggi; 0 thread system idle
  - Prioritas realtime: 16 - 31
  - Prioritas dinamis: 1 - 15
- Pemakai menentukan *priority class*:
    - realtime (24) , high (13), normal (8) dan idle (4)
  - dan sebuah prioritas relatif:
    - tertinggi (+2), diatas normal (+1), normal (0), dibawah normal (-1), dan terendah (-2)
  - untuk menetapkan *starting priority*
- Threads juga mempunyai *current priority*

Prioritas Win XP

	real-time	high	above normal	normal	below normal	idle priority
time-critical	31	15	15	15	15	15
highest	26	15	12	10	8	6
above normal	25	14	11	9	7	5
normal	24	13	10	8	6	4
below normal	23	12	9	7	5	3
lowest	22	11	8	6	4	2
idle	16	1	1	1	1	1

Figure 5.12 Windows XP priorities.

Daftar Dispatcher Ready



- Mencatat semua thread yang siap di eksekusi
- Antrian thread di berikan ke setiap level
- Temukan ReadyThread

Pencegahan Starvation

- Thread prioritas rendah mungkin tidak pernah dieksekusi
- “Kebijakan Anti-CPU starvation”
  - thread yang tidak dieksekusi selama 3 detik
  - di boost prioritasnya ke 15
  - quanta ganda

Penjadwalan di Windows

- Unit dasar penjadwalan adalah sebuah thread
  - (Anggap dulu thread adalah proses)
- Penjadwalan berdasarkan prioritas per thread
- OS preemptiv
- Quanta variabel

Review

- Kenapa diperlukan multiprogramming?
- Apa perbedaan penjadwalan preemptive dan non-preemptiv?
- Pastikan anda mengerti betul jawaban pertanyaan 2
- Apa masalah yang terjadi di antrian banyak level, bagaimana mengatasinya?