Sistem Operasi

Pertemuan 7 – CPU Scheduling

Mengenal Scheduling

Pada dasarnya, satu CPU hanya bisa mengerjakan satu process sekali waktu

Process lain akan dikerjakan bergantian begitu CPU free

Tujuan : memaksimalkan penggunaan CPU dengan multiprogramming

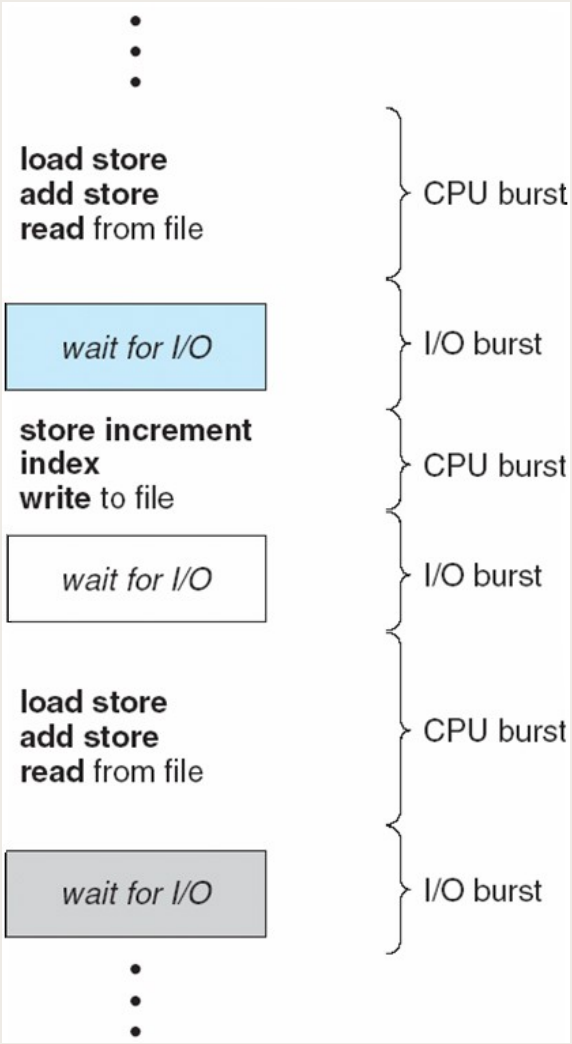
Bagaimana?

- Sebuah process (P_1) dieksekusi sampai masuk state waiting (misal karena I/O request)
- CPU dalam kondisi free
- OS membawa proses lain (P_2) untuk diolah CPU

Siklus Proses

Siklus pemrosesan ada dua :

- Siklus eksekusi CPU disebut CPU burst
- Siklus I/O wait disebut I/O burst



CPU Scheduler

Tugas scheduler : memilih process di ready queue untuk dieksekusi di CPU

Keputusan untuk scheduling diambil ketika sebuah proses :

- Pindah state dari running ke waiting
- Pindah state dari running ke ready. e.x. interrupt
- Pindah state dari waiting ke ready. e.x. I/O completion
- Terminate

CPU Scheduler

Scheduling pada 1 dan 4 disebut nonpreemptive scheduling (tidak bisa interrupt)

Ekesekusi bisa berpindah tanpa harus menunggu proses yang sedang dieksekusi Terminate atau Waiting.

Scheduling selainnya disebut preemptive scheduling

Non preemptive

- Dibawah non-preemptive scheduling, ketika CPU sudah mengalokasikan proses maka proses akan tetap di sana hingga CPU melepaskan dengan mematikan atau menukar ke Waiting State

- Scheduling ini digunakan oleh Microsoft Windows 3.1 dan Apple Macintosh

- Ini adalah satu-satunya metode yang dapat digunakan di beberapa perangkat tertentu, karena tidak memerlukan hardware khusus (timer) yang diperlukan untuk preemptive

Preemptive

- Tipe scheduling ini task/proses biasanya mempunyai prioritas. Ada kalanya sebuah tugas atau proses harus dijalankan dengan prioritas lebih tinggi sebelum tugas yang lainnya dijalankan.

- Oleh karena itu, tugas yang diperlukan akan diinterupsi untuk beberapa waktu dan kemudian dilanjutkan kemudian ketika tugas prioritas selesai

Kriteria Scheduling

CPU utilization – buat CPU sesibuk mungkin. Jangan biarkan CPU idle

Throughput – Jumlah proses yang selesai dieksekusi per unit waktu

Turnaround time – jumlah waktu untuk mengeksekusi sebuah proses

Waiting time – jumlah waktu sebuah proses harus menunggu di ready queue

Response time – jumlah waktu yang dibutuhkan dari pertama request eksekusi

dikirim sampai respon pertama muncul (bukan output akhir)

Indikator Optimal

Makin tinggi makin bagus:
Max CPU utilization
Max throughput

Makin renda makin bagus:
Min turnaround time
Min waiting time
Min response time

Algoritma Scheduling

Terdapat beberapa algoritma dasar yang digunakan sistem operasi dalam mengontrol proses-proses yang berjalan.

Beberapa algoritma scheduling

- First-Come, First-Served (FCFS) Scheduling
- Shortest-Job-First (SJF) Scheduling
- Priority Scheduling
- Round Robin (RR)
- Multilevel Queue

First Come First Serve Scheduling

- First Come First Serve mirip dengan FIFO struktur data, yang di mana data elemen yang ditambahkan pertama ke dalam antrian yang akan pertama keluar

- Digunakan dalam sistem Batch

- Sangat mudah dipahami dan diimplemetasikan menggunakan antrian struktur data

- Contoh nyata dari ini adalah antrian tiket

Ilustrasi

Process	Burst Time
P1	24
P2	3
P3	3

- Waiting time for P_1 = 0; P_2 = 24; P_3 = 27
- Average waiting time: (0 + 24 + 27)/3 = 17

Shortest-Job-First (SJF) Scheduling

Process dengan CPU burst time lebih kecil akan dieksekusi lebih dulu

SJF memberi hasil scheduling yang optimal

Berdasarkan kapan keputusan scheduling dilakukan, SJF dibagi menjadi :

- Non-preemptive SJF scheduling
- Preemptive SJF scheduling

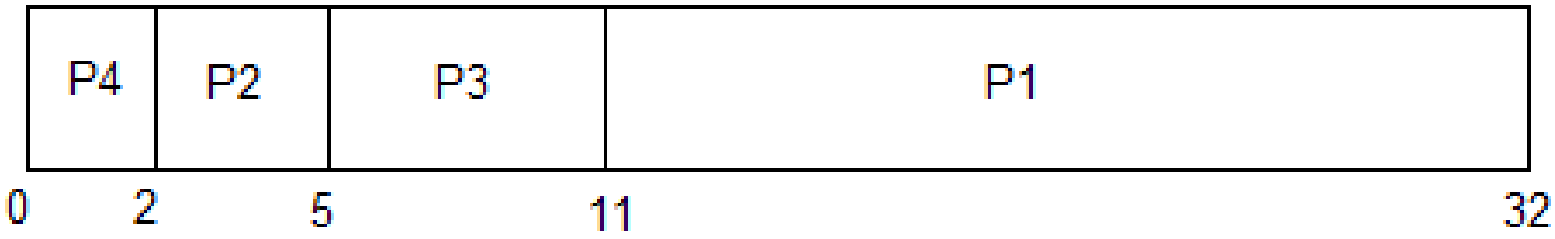
Non-Preemptive SJF Scheduling

Algoritma
1) Jalankan process dengan burst time terkecil sampai masuk state waiting
2) Pilih dan jalankan process dengan burst time terkecil berikutnya
Contoh

PROCESS	BURST TIME
P1	21
P2	3
P3	6
P4	2

Non-Preemptive SJF Scheduling

Average waiting time =
$T_1 + T_2 + T_3 + T_4 = (0 + 2 + 3 + 6 + 21) / 4 = 4.5$



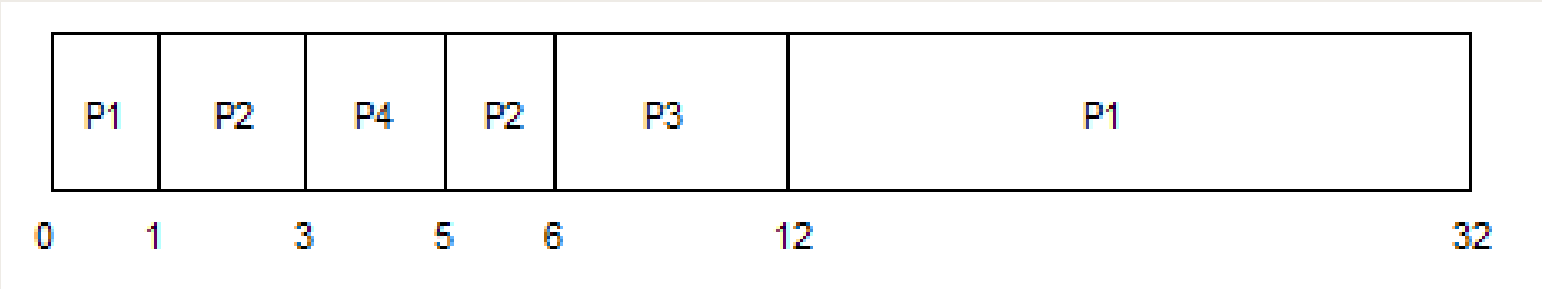
Preemptive SJF Scheduling

- Jalankan process dengan burst time terkecil
- TANPA menunggu state berubah waiting, cek burst time dari process yang baru masuk (T_{new})
- Jika Tnew < sisa burst time process sekarang, pindah eksekusi ke process baru

PROCESS	BURST TIME	ARRIVAL TIME
P1	21	0
P2	3	1
P3	6	2
P4	2	3

Preemptive SJF Scheduling

- Average waiting time =
- $T_1 + T_2 + T_3 + T_4 = ((5-3) + (6-2) + (12-1) / 4 = 4.25$



Priority Scheduling

- CPU mengeksekusi dulu proses dengan prioritas tertinggi (proses dengan priority number terkecil = process dengan prioritas tertinggi)
  - Preemptive
  - Nonpreemptive
- SJF adalah bentuk priority scheduling dimana prioritasnya adalah inverse dari estimasi nilai burst berikutnya
- Problem = **Starvation** – process dengan prioritas rendah bisa jadi tidak pernah dieksekusi
- Solution = **Aging** – seiring berjalannya waktu, tingkatkan prioritas dari sebuah process

Round Robin

- Setiap proses mendapat sebuah alokasi unit waktu CPU (quantum waktu q) biasanya sekitar 10-100ms.

- Pemrosesan dimulai dari process yang pertama masuk

- Setelah quantum waktu q berlalu, eksekusi berpindah ke process berikutnya sampai waktu quantum q nya berakhir, dan seterusnya. Process sebelumnya ditempatkan di akhir ready queue

Round Robin

- Jika waktu quantum = q dengan n process di ready queue, maka setiap process mendapat menunggu paling laa (n-1)q unit waktu.

- Butuh timer untuk interrupt setiap quantum agar proses berikutnya bisa dijadwalkan

- Performance
  - q besar = FIFO
  - q kecil = q must be large with respect to context switch, otherwise overhead is too high

Multi Prosesor Scheduling

Scheduling dengan Multi Prosesor mengatur prosesor mana yang tersedia untuk diberikan tugas, indikator :

Processor Affinity
- Soft Affinity
- Hard Affinity

Load Balancing
- Push Migration
- Pull Migration

Multicore Processors
- Coarse-Grained Multithreading
- Fine-Grained Multithreading

Processor Affinity

Processor Affinity

- Soft Affinity
  - SO punya kebijakan untuk menjaga proses tetap dalam prosesor yang sama

- Hard Affinity
  - Beberapa SO seperti Linux memungkinkan untuk migrasi antar prosesor

Load Balancing

Load Balancing

Sebuah fenomena membuat beban menjadi seimbang antar prosesor

- Push Migration
  - Proses akan melakukan pengecekan beban, jika terjadi ketidak seimbangan maka dia akan menyeimbangkan
- Pull Migration
  - Prosesor yang idle (menganggur) akan menarik tugas menunggu dari prosesor yang sibuk

Multicore Processor

Multicore Processors
Sebuah teknologi di mana prosesor fisik memiliki banyak inti (core), sehingga kinerja lebih efisien dan hemat energi

- Coarse-Grained Multithreading
  - Sebuah thread di eksekusi di prosesor sampai latency panjang seperti Memory Stall terjadi
- Fine-Grained Multithreading
  - Multithreading ini menukar antara thread ke level yang lebih bagik

Scheduling Frekuensi

- CPU Scheduling tidak hanya mengatur proses, threading dan lain-lain. CPU scheduler dapat mengatur jangkauan frekuensi yang sistem operasi dapat gunakan untuk meraih kinerja optimal

- Pengatur frekuensi CPU bisa disebut juga dengan CPU Governor.

- CPU Governor mudah ditemukan di SO UNIX/Linux/Android

CPU Governor Standar SO

- CPU Governor standar yang disediakan oleh kernel sistem operasi adalah
  - Ondemand
  - Powersave
  - Performance
  - Userspace
  - Conservative
  - Schedutil
  - PegasusQ (Android)
  - Lulz (Android)
  - SmartAss (Android)
  - DLL
- Konfigurasi ini mengatur jangkauan frekuensi yang berbeda-beda dengan kondisi yang berbeda