Penjadwalan CPU

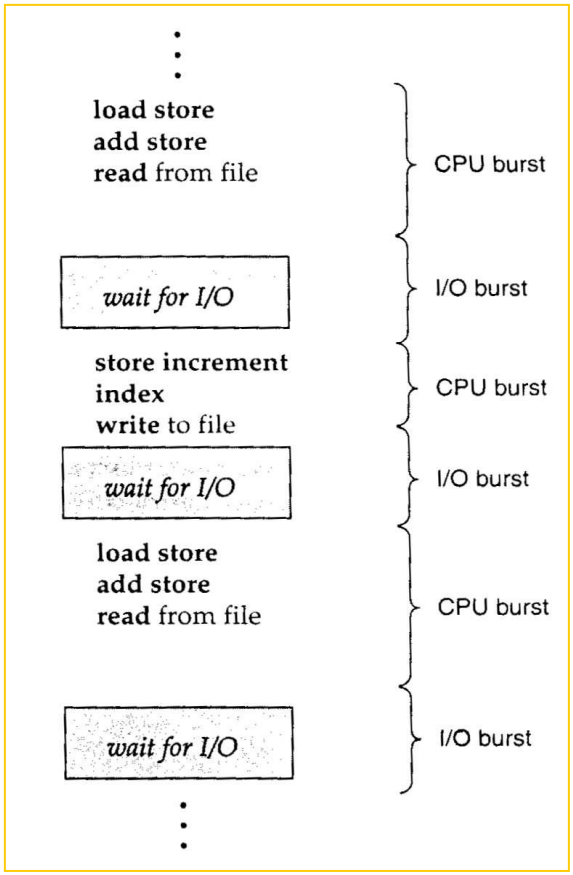
Konsep Dasar

- Penjadwalan CPU adalah fungsi OS yang fundamental, dimana hampir semua resource komputer dijadwalkan sebelum digunakan
- CPU adalah satu dari resource komputer utama, penjadwalan CPU adalah pusat desain OS
- Penjadwalan CPU dibutuhkan pada sistem multiprogramming untuk memaksimalkan utilitas CPU
- Pada sistem uni-processor, tidak pernah lebih dari satu proses yang running, sehingga jika terdapat banyak proses, sisanya akan menunggu sampai CPU bebas dan dapat dijadwal ulang

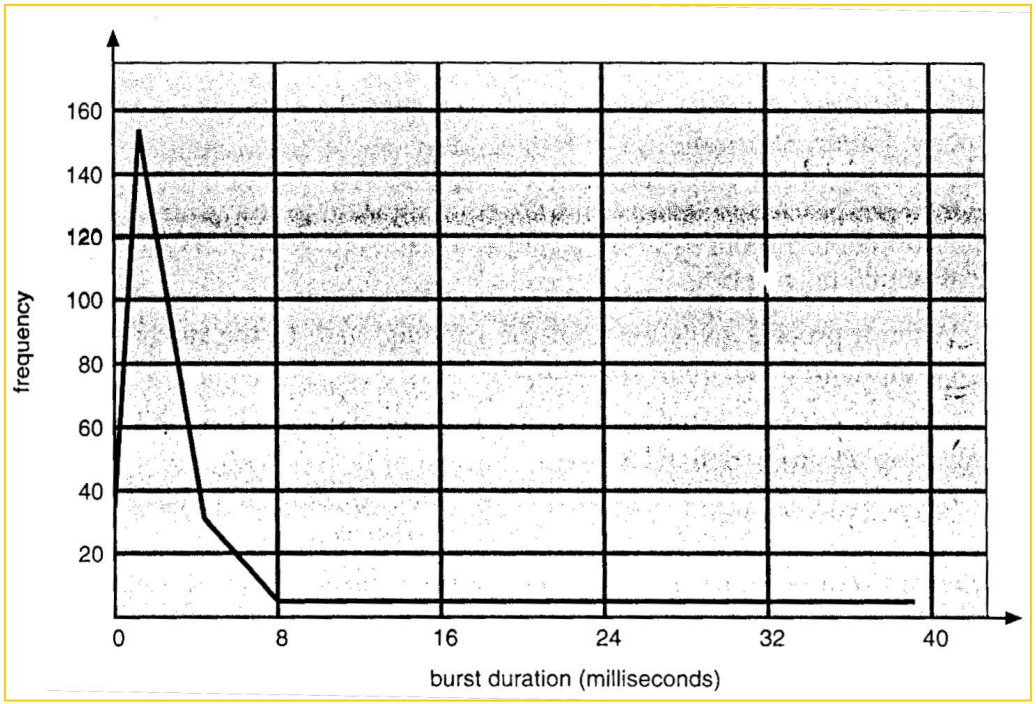
Siklus CPU-I/O Burst (1)

- Penjadwalan CPU tergantung properti proses sbb :
  - Eksekusi proses terdiri dari siklus eksekusi CPU dan menunggu I/O
  - Proses berpindah antara kedua state tersebut
  - Eksekusi proses dimulai dengan CPU burst, diikuti I/O burst, kemudian diikuti dengan CPU burst lain, I/O burst lain dan seterusnya
  - CPU burst terakhir akan berakhir dengan sistem meminta terminasi eksekusi bukan karena I/O burst yang lain
- Program I/O bound mempunyai banyak CPU burst yang sangat pendek
- Program CPU bound mempunyai beberapa CPU burst yang sangat panjang

Siklus CPU-I/O Burst (2)



Urutan CPU dan I/O Burst



Histogram waktu CPU Burst

Preemptive scheduling

- Keputusan penjadwalan CPU mempertimbangkan 4 keadaan sbb :
  1. Ketika proses berpindah dari state running ke state waiting (contoh : permintaan I/O, atau menunggu terminasi satu dari proses child)
  2. Ketika proses berpindah dari state running ke state ready (contoh : saat terjadi interrupt)
  3. Ketika proses berpindah dari state waiting ke state ready (contoh : menyelesaikan I/O)
  4. Ketika proses diterminasi
- Skema penjadwalan disebut "non-preemptive" jika penjadwalan berada pada keadaan 1 dan 4 dan disebut "preemptive" pada keadaan lain
- Pada penjadwalan non-preemptive, ketika CPU dialokasikan ke suatu proses, proses memegang CPU sampai dilepaskan CPU karena proses diterminasi atau karena berpindah ke state waiting. MS Window mengadopsi skema ini

Dispatcher

- "Dispatcher" adalah modul yang memberikan kontrol pada CPU terhadap proses yang dipilih dengan short-term scheduling.
- Fungsi-fungsinya :
  - Switching context
  - Switching ke user-mode
  - Melompat ke lokasi tertentu pada user program untuk memulai program
- Karena dispatcher digunakan setiap berpindah proses, dispatcher harus secepat mungkin
- Waktu yang dibutuhkan dispatcher untuk menghentikan suatu proses dan memulai menjalankan proses yang lain disebut "dispatch lantency"

Kriteria penjadwalan

- Algoritma penjadwalan CPU yang berbeda mempunyai properti yang berbeda dan menyerupai satu class dari proses dengan proses lain
- Beberapa kriteria yang digunakan untuk membandingkan algoritma penjadwalan CPU adalah :
  - CPU utilization. Idenya adalah menggunakan CPU sesibuk mungkin. Pada sistem real, berkisar antara 40 – 90%
  - Throughput : banyaknya proses yang selesai dikerjakan dalam satu satuan waktu
  - Turnaround time : jumlah waktu yang diperlukan untuk mengeksekusi proses, dari menunggu untuk meminta tempat di memori utama, menunggu di ready queue, eksekusi oleh CPU dan mengerjakan I/O
  - Waiting time : jumlah waktu yang diperlukan suatu proses untuk menunggu di ready queue
  - Response time : jumlah waktu yang dibutuhkan suatu proses dari minta dilayani hingga ada respon pertama dari permintaan tsb
- Diantara semua kriteria diatas, perlu memaksimalkan utilitas CPU dan throughput dan meminimalkan turnaround time, waiting time dan response time

Algoritma penjadwalan

- Penjadwalan CPU adalah permasalahan menentukan proses mana pada ready queue yang dialokasikan ke CPU
- Terdapat beberapa algoritma penjadwalan CPU, pada kuliah ini akan dibahas :
  - First Come First Served (FCFS)
  - Shortest Job First Scheduler (SJF)
  - Priority Scheduling
  - Round Robin
- Setiap algoritma diukur "turnaround time dan "waiting time" untuk membandingkan performansi dengan algoritma lain
- Untuk mengukur turnaround dan waiting time, digunakan "Gant chart". CPU time (Burst time) membutuhkan semua proses diasumsikan diketahui. Arrival time untuk setiap proses pada ready queue diasumsikan diketahui

First Come First Served (FCFS)

- Proses yang pertama kali meminta jatah waktu untuk menggunakan CPU akan dilayani terlebih dahulu
- Rata-rata waktu tunggu (Average Waiting Time = AWT) cukup tinggi
- Non-preemptive karena sekali CPU dialokasikan pada suatu proses, maka proses tersebut akan tetap memakai CPU sampai proses tersebut melepaskannya, yaitu jika proses berhenti atau meminta I/O
- Kelemahan : adanya convoy effect

Contoh FCFS

Proses	Burst Time
P_1	24
P_2	3
P_3	3

Shortest Job First Scheduler (SJF)

- Proses yang memiliki CPU burst paling kecil dilayani terlebih dahulu
- Optimal, tapi sulit diimplementasikan karena sulit mengetahui CPU burst berikutnya
- Adanya prediksi CPU burst dengan exponential average
- Termasuk preemptive atau non preemptive
- Non preemptive bila proses pertama diselesaikan sampai habis CPU burst-nya sebelum proses kedua dijalankan
- Preemptive bila sisa waktu proses pertama lebih besar dari proses kedua, maka proses pertama dihentikan dan diganti proses kedua (dikenal dg shortest-remaining-time-first)

Contoh SJF

Proses	Burst Time
P_1	6
P_2	8
P_3	7
P_4	3

Shortest Job First

Proses	Arrival Time	Burst Time
P_1	0	8
P_2	1	4
P_3	2	9
P_4	3	5

Shortest remaining time first

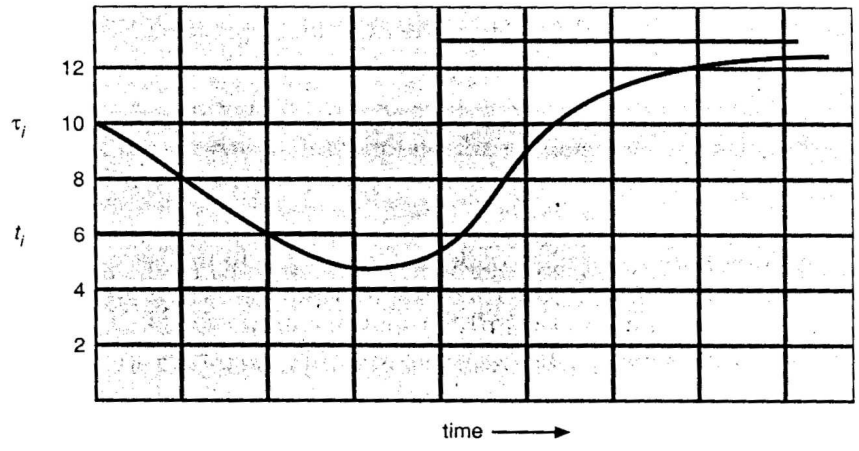
Prediksi Panjang CPU Burst Berikutnya

- CPU burst berikutnya biasanya diprediksi sebagai rata-rata exponensial dari ukuran panjang CPU burst berikutnya dg formulasi :

$$\tau_{n+1} = \alpha t_n + (1-\alpha)\tau_n$$

Bila α=0, maka $\tau_{n+1} = \tau_n$

Bila α=1, maka $\tau_{n+1} = t_n$



CPU burst (t_i)	6	4	6	4	13	13	13	...
"guess" (τ_i)	10	8	6	6	5	9	11	12	...

Priority Scheduling

- Tiap proses dilengkapi dengan prioritas
- CPU dialokasikan untuk proses dg prioritas paling tinggi, apabila prioritas sama, digunakan algoritma FCFS
- Prioritas menyangkut masalah waktu, memori, banyaknya file yang boleh dibuka dan perbandingan rata-rata I/O burst dg CPU burst
- Bersifat preemptive dan non preemptive
- Pada non preemptive, bila P1 datang saat proses P0, prioritas P1 > P0, maka P0 diselesaikan sampai habis CPU burst-nya
- Pada preemptive, P0 dihentikan dulu dan CPU digunakan untuk P1

Contoh Priority

Proses	Burst Time	Priority
P_1	10	3
P_2	1	1
P_3	2	3
P_4	1	4
P_5	5	2

Round Robin (1)

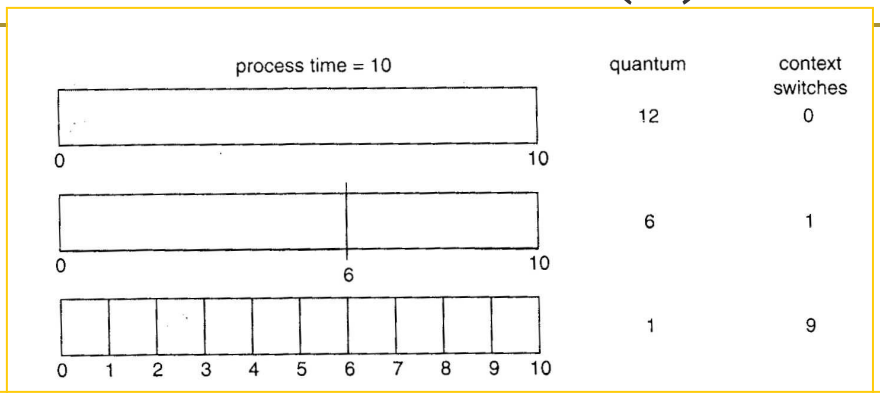
- Konsep dasar : time-sharing
- Sama dengan FCFS yang bersifat preemptive
- Quantum time untuk membatasi waktu proses
- Bila CPU burst < Quantum time, proses melepaskan CPU jika selesai dan CPU digunakan untuk proses selanjutnya
- Bila CPU burst > Quantum time, proses dihentikan sementara dan mengantri di ekor dari ready queue, CPU menjalankan proses berikutnya

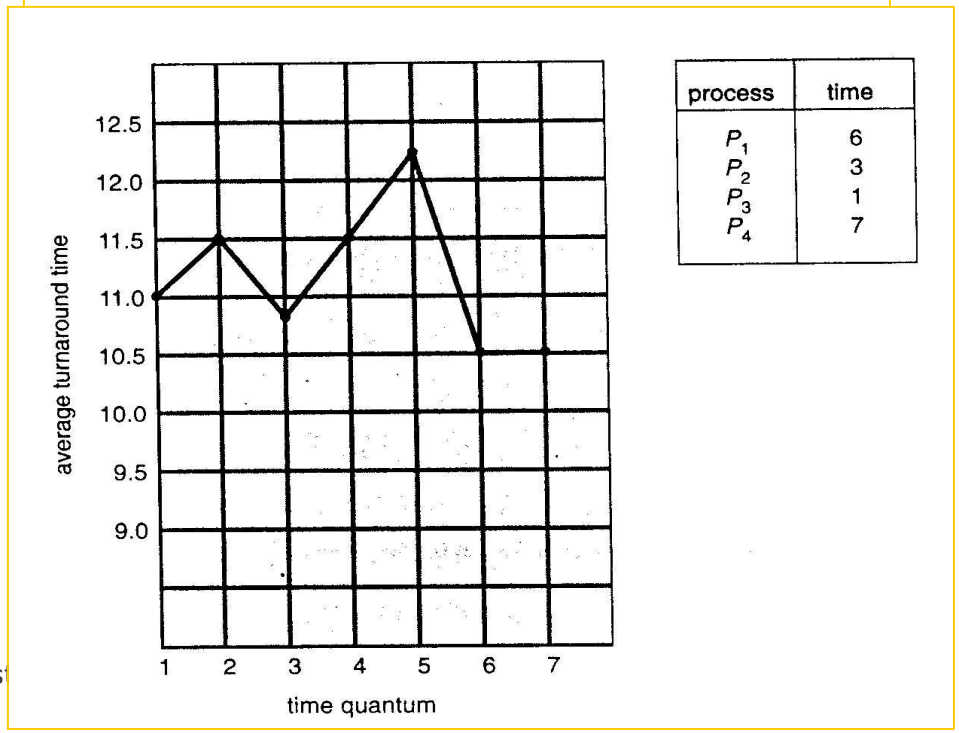
Contoh Round Robin

Proses	Burst Time
P_1	24
P_2	3
P_3	3

Quantum Time = 4

Round Robin (2)





process	time
P_1	6
P_2	3
P_3	1
P_4	7

Waktu quantum yang kecil meningkatkan context switch

Waktu turnaround bervariasi dengan waktu quantum