PENJADWALAN PROSES

Abas Ali Pangera, Dony Ariyus, *Jurusan Teknik Informatika, STMIK AMIKOM Yogyakarta, Jl. Ring Road Utara, Condong Catur, Sleman, Yogyakarta - Indonesia*

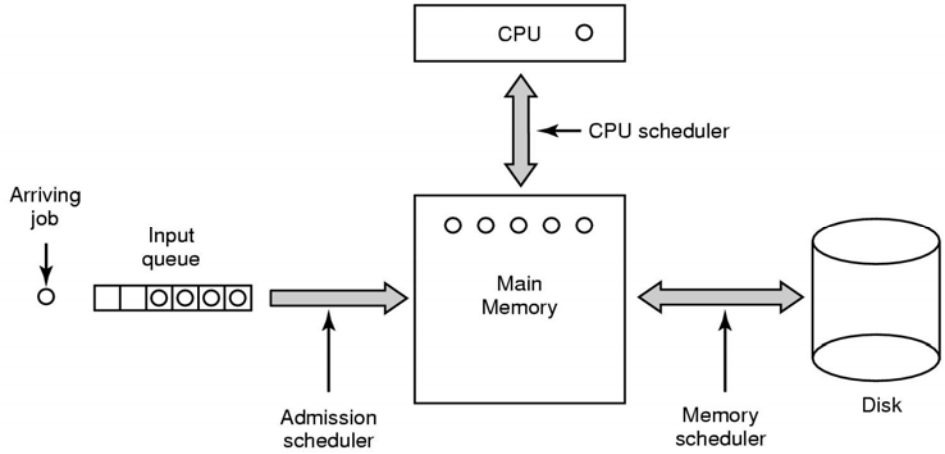
Abstract

Konsep penjadwalan proses seperti, konsep dasar penjadwalan proses, kriteria penjadwalan dan algoritma penjadwalan, tujuan Setelah mempelajari materi dalam bab ini, diharapkan mampu:

- Memahami tentang konsep dasar penjadwalan CPU
- Memahami kriteria yang diperlukan untuk penjadwalan CPU
- Memahami beberapa algoritma penjadwalan CPU yang terdiri dari algoritma First Come First Serve, Shortest Job First, Priority dan Round Robin

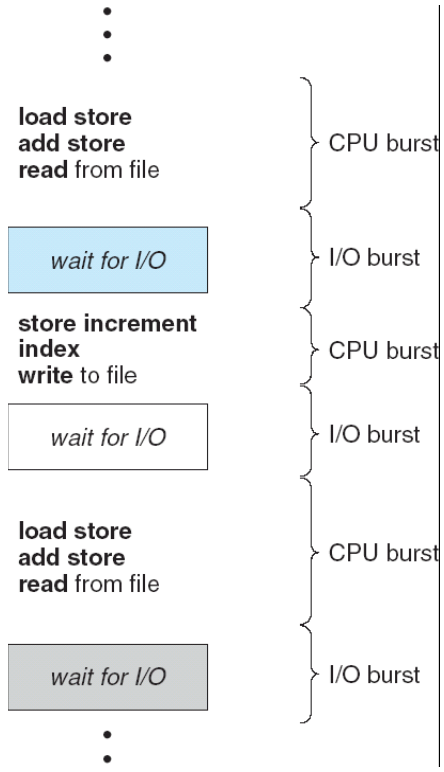
Konsep Dasar Penjadwalan Proses

Pada sistem komputer terdapat beberapa bentuk dari penjadwalan, admission (pintu masuk ke system), memory, dan CPU scheduler coba lihat gambar scheduler dari bacth system berikut:

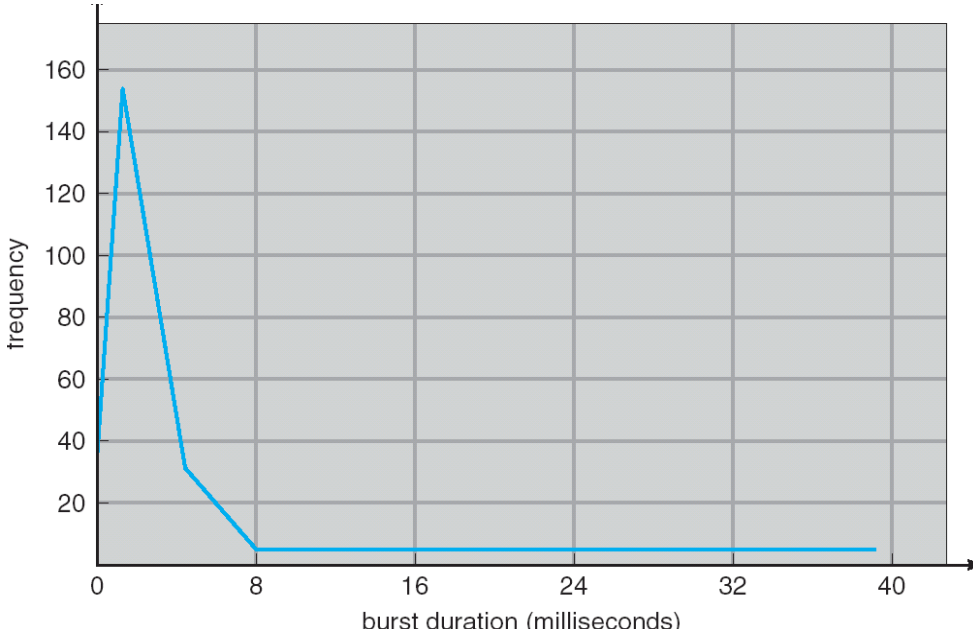


Tiga Level Penjadwalan

Pada sistem multiprogramming, selalu akan terjadi beberapa proses berjalan dalam suatu waktu. Sedangkan pada uniprogramming hal ini tidak akan terjadi, karena hanya ada satu proses yang berjalan pada saat tertentu. Sistem multiprogramming diperlukan untuk memaksimalkan utilitas CPU. Pada saat proses dijalankan terjadi siklus eksekusi CPU dan menunggu I/O yang disebut dengan siklus CPU-I/O burst. Eksekusi proses dimulai dengan CPU burst dan dilanjutkan dengan I/O burst, diikuti CPU burst lain, kemudian I/O burst lain dan seterusnya seperti pada Gambar 6-2.



Siklus CPU-I/O Burst



Histogram Waktu CPU Burst

Pada saat suatu proses dieksekusi, terdapat banyak CPU burst yang pendek dan terdapat sedikit CPU burst yang panjang. Program yang I/O bound biasanya sangat pendek CPU burst nya, sedangkan program yang CPU bound kemungkinan CPU burst nya sangan lama. Hal ini dapat digambarkan dengan grafik yang eksponensial atau hiper eksponensial seperti pada Gambar 6-3. Oleh karena itu sangat penting pemilihan algoritma penjadwalan CPU.

CPU Scheduler

Pada saat CPU menganggur, maka sistem operasi harus menyeleksi prosesproses yang ada di memori utama (ready queue) untuk dieksekusi dan mengalokasikan CPU untuk salah satu dari proses tersebut. Seleksi semacam ini disebut dengan shortterm scheduler (CPU scheduler). Keputusan untuk menjadwalkan CPU mengikuti empat keadaan di bawah ini :

1) Apabila proses berpindah dari keadaan running ke waiting;
2) Apabila proses berpindah dari keadaan running ke ready;
3) Apabila proses berpindah dari keadaan waiting ke ready;
4) Apabila proses berhenti.

Apabila model penjadwalan yang dipilih menggunakan keadaan 1 dan 4, maka penjadwakan semacam ini disebut *non-peemptive*. Sebaliknya, apabila yang digunakan adalah keadaan 2 dan 3, maka disebut dengan *preemptive*. Pada *non-preemptive*, jika suatu proses sedang menggunakan CPU, maka proses tersebut akan tetap membawa CPU sampai proses tersebut melepaskannya (berhenti atau dalam keadaan waiting). *Preemptive* scheduling memiliki kelemahan, yaitu biaya yang dibutuhkan sangat tinggi. Antara lain, harus selalu dilakukan perbaikan data. Hal ini terjadi jika suatu proses ditinggalkan dan akan segera dikerjakan proses yang lain.

Dispatcher

Dispatcher adalah suatu modul yang akan memberikan kontrol pada CPU terhadap penyeleksian proses yang dilakukan selama *short-term scheduling*. Fungsifungsi yang terkandung di dalam-nya meliputi:

1. *Switching context;*
2. Switching ke user-mode;
3. Melompat ke lokasi tertentu pada user program untuk memulai program. Waktu yang diperlukan oleh dispatcher untuk menghentikan suatu proses dan memulai untuk menjalankan proses yang lainnya disebut *dispatch latency*.

Kriteria Penjadwalan

Algoritma penjadwalan CPU yang berbeda akan memiliki perbedaan properti. Sehingga untuk memilih algoritma ini harus dipertimbangkan dulu properti-properti algoritma tersebut. Ada beberapa kriteria yang digunakan untuk melakukan pembandingan algoritma penjadwalan CPU, antara lain:

1. *CPU utilization*: Diharapkan agar CPU selalu dalam keadaan sibuk. Utilitas CPU dinyatakan dalam bentuk prosen yaitu 0-100%. Namun dalam kenyataannya hanya berkisar antara 40-90%.
2. *Throughput*: throughput Adalah banyaknya proses yang selesai dikerjakan dalam satu satuan waktu. Cara untuk mengekspresikan throughput adalah dengan jumlah proses user yang dapat dieksekusi dalam interval waktu tertentu. Sasaran penjadwalan adalah memaksimalkan jumlah proses yang dilayani per satu interval waktu. Lebih tinggi waktu throughput maka lebih banyak kerja yang dilakukan sistem
   Ktiteria-kriteria tersebut saling tergantung dan dapat saling bertentangan sehingga tidak dimungkinkan optimasi semua criteria secara simultan
   Contoh:

Untuk memberikan waktu tanggap kecil umumnya memerlukan penjadwalan yang sering beralih-alih di antara proses-proses yang ada. Cara ini meningkatkan overhead sistem sehingga menurunkan efisiensi.
Kebijaksanaan perancangan penjadwalan melibatkan kompromi di antara kebutuhan-kebutuhan yang saling bertentangan. Kompromi ini tergantung sifat dan penggunaan sistem komputer

3. *Turnaround time*: Banyaknya waktu yang diperlukan untuk mengeksekusi proses, dari mulai menunggu untuk meminta tempat di memori utama, menunggu di ready queue, eksekusi oleh CPU, dan mengerjakan I/O sampai semua proses-proses tersebut diselesaikan. Waktu yang dimaksud adalah waktu yang dihabiskan proses berada di sistem, atau juga bisa diekspresikan sebagai penjumlahan waktu eksekusi (waktu pelayanan proses) dan waktu menunggu dari porses tersebut atau bisa dirumuskan seperti:
   Trun arround time = waktu eksekusi + waktu tunggu
   Sasaran dari penjdawalan adalah meminimalkan turn arround time
4. *Waiting time*: Waktu yang diperlukan oleh suatu proses untuk menunggu di *ready queue*. Waiting time ini tidak mempengaruhi eksekusi proses dan penggunaan I/O.
5. *Response time*: Waktu yang dibutuhkan oleh suatu proses dari minta dilayani hingga ada respon pertama yang menanggapi permintaan tersebut. Respon time berbeda untuk sistem interaktif dan sistem waktu nyata
   a. Respon time sistem interaktif: respon time dalam sistem interkatif didefinisikan sebagai waktu yang dihabiskan dari saat karakter terakhir dari perintah dimasukan oleh program atau transaksi sampai hasil pertama muncul di perangkat output seperti layar (terminal)
   b. Respon time sistem waktu nyata: pada sistem waktu nyata (real time), respon didefinisikan sebagai waktu dari saat kemunculan suatu kejadian (internal atau eksternal) sampai instruksi pertama rutin layanan terhadap kejadian dieksekusi
6. *Fairness*: adil adalah menyakinkan proses-proses diperlakukan sama yaitu mendapat jatah waktu layanan CPU yang sama dan tidak ada proses yang tidak kebagian layanan CPU sehingga mengalai starvation. Starvation merupakan suatu kondisi bahwa proses tidak pernah berjalan tidak dijadwalkan untuk berjalan.

Algoritma Penjadwalan

Penjadwalan CPU menyangkut penentuan proses-proses yang ada dalam ready queue yang akan dialokasikan pada CPU. Terdapat beberapa algoritma penjadwalan CPU seperti dijelaskan pada sub bab di bawah ini.

First Come First Server (FCFS)

Pertama datang, pertama dilayani (First In, First Out atau FIFO) tidak peduli apakah Burst timenya panjang atau pendek, sebuah proses yang sedang dikerjakan, diselesaikan dulu barulah proses berikutnya dilayani.

Penjadwalan FCFS merupakan penjadwalan:

- Penjadwalan non-preemptive (run-to-completion)
- Penjadwalan tidak berprioritas

Ketentuan dari penjadwalan FCFS adalah:

- Proses-proses dberi jatah waktu pemroses, diurut dengan waktu kedatangannya

- Begitu proses mendapat jatah waktu pemroses, proses dijalankan sampai proses tersebut selesai, walaupun ada proses lain yang datang, proses tersebut berada dalam antrian sistem atau disebut dengan ready queue

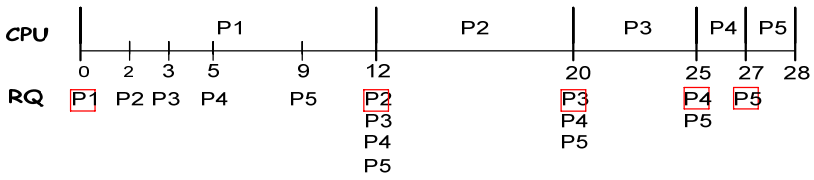
Pada dasarnya algoritma penjadwalan ini cukup adil dalam hal bahasa, karena proses yang datang lebih dulu dikerjakan terlebih dahulu, pada secara konsep dari sistem operasi, penjadwalan model ini tidak adil karena proses-proses yang membutuhkan waktu yang lama, membuat proses-proses yang memiliki waktu proses yang lebih pendek menunggu sampai proses tersebut selesai, sedangkan proses-proses yang tidak penting membuat proses penting menunggu.

Penjadwalan FCFS cocok digunakan untuk sistem batch yang sangat jarang melakukan interaksi dengan user secara langsung, tapi tidak cocok digunakan untuk sistem interaktif karena tidak memberi waktu tanggap yang bagus, dan begitu juga dengan sistem waktu nyata.

Contoh dari penjadwalan FCFS adalah:

Misalnya proses-proses yang akan dikerjakan oleh CPU adalah sebagai berikut :

Proses	Arrival Time (AT)	Burst Time (BT)
P1	0	12
P2	2	8
P3	3	5
P4	5	2
P5	9	1



Pada saat posisi P1 masuk pada waktu 0, maka P1 dijalankan sebanyak 12, dan P2 masuk pada saat 2, P3 pada saat 3, P4 pada saat 5 dan P5 pada waktu 9. Pada posisi 12 P1 selesai dikerjakan, dan P2 akan dieksekusi karena P2 lebih dulu berasa didalam Ready Queue(RQ) karena pada contoh di atas menggunakan algoritma FCFS (First Come First Server) dan begitulah seterusnya sampai semua proses selesai dikerjakan.setelah semua proses dikerjakan maka dihitung waktu tunggu setiap proses maka akan ditemukan jumlah waktu tunggu rata-rata dari setiap proses, seperti contoh di bawah ini:

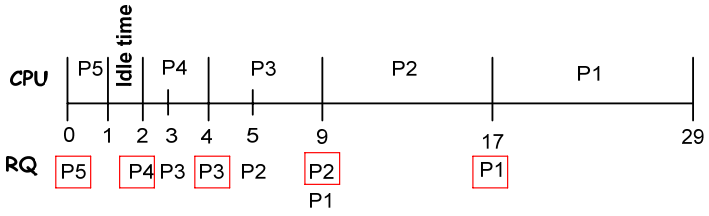
Jadi waktu tunggu setiap proses adalah :

P1 = 0	= 0
P2 = 12-2	=10
P3 = 20-3	= 17
P4 = 25-5	= 20
P5 = 27-9	= 18
Jumlah	= 65

Rata-rata waktu tunggu untuk setiap proses adalah : Avg 65/5 = 13 satuan waktu

Jika urutan proses pada contoh di atas di balikkan misal: P5, P4, P3, P2, P1 dengan waktu kedatangan dan burst tetap sama, maka akan terdapat waktu ganggur CPU (idle time) antara waktu 1 dan 2 sehingga panjangnya waktu yang dibutuhkan untuk menyelesaikan lima proses tersebut menjadi 29 atau bertambah satu dari waktu yang dibutuhkan sebelumnya.

Proses	Arrival Time (AT)	Burst Time (BT)
P5	0	1
P4	2	2
P3	3	5
P2	5	8
P1	9	12



Daftar Pustaka

Ariyus,Dony,2006, "*Computer Security*", Andi Offset, Yogyakarta

Ariyus, Dony,2005," kamus hacker", Andi offset, Yogyakarta

Bob DuCharme, 2001," *The Operating System Handbook or, Fake Your Way Through Minis and Mainframes*" Singapore: McGraw-Hill Book Co

Bill Venners.1998. "*Inside the Java Virtual Machin"e* . McGraw-Hill.

Deitel, Harvey M, 2004 " *operating systems" 3th Edition*, Massachusetts: Addison-Wesley Publshing Company

Gary B. Shelly, 2007**,** "*Discovering Computers: Fundamentals*" Thomson

Gollmann, Dieter,1999 "*Computer Security*" Jhon Willey & Son Inc, Canada

Grosshans,D. 1986," *File system: design and implementation*", Englewwod Cliffs, New Jersey : Prentice-Hall Inc.

Harvey M Deitel dan Paul J Deitel. 2005. *Java How To Program*. Sixth Edition. Prentice Hall.

Hoare, C.A.R. 1985" *Communication sequential processes*"Englewood Cliffs, New Jersey, Prentice Hall Inc

Jean Bacon, Tim Harris, 2003 "*Operating Systems: Concurrent and Distributed Software Design*" Massachhussets. Addison Wesley

Kenneth H Rosen. 1999. "*Discrete Mathematics and Its Application"*. McGraw Hill.

Madnick,Stuart E dan John J. Donovan, 1974 " *operating system*", Singapore: McGraw-Hill Book Co

Michael Kifer and Scott A. Smolka, 2007 *Introduction to Operating System Design and Implementation The OSP 2 Approach*, Springer-Verlag London

Microsoft 1999. *Microsoft Windows User Experience*. Microsoft Press.

Milenkovie, Milan. 1992. "*Operationg system: Concepts and Design*", Singapore: McGraw-Hill Book Co

Randall Hyde. 2003. *The Art of Assembly Language*. First Edition. No Strach Press

Robert betz, 2001 "*Intoduction to Real-time operation system*", Department of Electrical and Computer Engineering University of Newcastle, Australia

Robert Love. 2005. *Linux Kernel Development* . Second Edition. Novell Press

Ron White,1998, *How Computers Work, Fourth Edition*, Que corporation, A Division of Macmillan Computer Publishing, USA

Shay, William A. 1993, " *Introduction to Operationg System*" New York: HarperCollins College Publishers

Silberschatz, Peter Galvin, dan Grag Gagne. 2000. *"Applied Operating System, 1$^{s"}$t"* John Wile & Hiil Book Co

Silberschatz, A., dan Galvin, P.2003, "Operating Sistem Concept. Sixth Edition". Massachhussets. Addisson- Wasley

Silberschatz, Peter Galvin, dan Grag Gagne. 2005. "*Operating Systems Concepts"*. Seventh Edition. John Wiley & Sons.

Stalling, William, 1995, "*Operating Sistems"*. New Jersey. Prentice – Hall

Stalling, William, 1996" *Computer Organization and Architecture*". New Jersey. Prentice – Hall

Stalling William, 1995, "*Network and Internetwork Security"* Prentice-Hall, USA

Tanenbaum, Andrew S, 1992 *"Modern Operating Sistems*". New Jersey. Prentice – Hall

Taenbaum, Andrew S, 2006, *"Operating Systems Design and Implementation, Third Edition"* Massachusetts