

Penjadwalan Proses

Deskripsi Penjadwalan Proses

Kumpulan kebijaksanaan dan mekanisme di sistem operasi yang berkaitan dengan urutan kerja yang dilakukan sistem komputer.

Penjadwalan bertugas memutuskan hal-hal berikut :

- Proses yang harus berjalan
- Kapan dan selama berapa lama proses berjalan

Sasaran utama penjadwalan proses adalah Optimasi kinerja sistem komputer menurut kriteria tertentu.

Kriteria untuk mengukur dan optimasi kinerja penjadwalan adalah sbb:

1. Adil (*fairness*)
2. Efisiensi
3. Waktu Tanggap (*response time*)
4. *Turn around Time*
5. *Troughput*

Adil (*fairness*)

Proses-proses diperlakukan sama yaitu mendapat jatah waktu layanan pemroses yang sama dan tidak ada proses yang tidak kebagian layanan pemroses sehingga mengalami *startvation*.

Startvation adalah kondisi bahwa proses tidak pernah berjalan karena tidak dijadwalkan untuk berjalan. Sasaran penjadwalan seharusnya menjamin setiap proses mendapat pelayanan dari pemroses secara adil.

Efisiensi

Efisiensi atau utilisasi pemroses dihitung dengan perbandingan (rasio) waktu sibuk pemroses dengan total waktu operasi sistem komputer secara keseluruhan. Sasaran penjadwalan adalah menjaga agar pemroses tetap dalam keadaan sibuk sehingga efisiensi sistem komputer mencapai nilai maksimum. Keadaan sibuk berarti pemroses tidak menganggur. Layanan pemroses termasuk waktu yang dihabiskan untuk mengeksekusi program pemakai dan layanan sistem operasi secara efektif, bukan untuk melakukan penjadwalan itu sendiri.

Waktu Tanggap (*response time*)

Waktu tanggap berbeda untuk :

- Sistem interaktif

Waktu yang dihabiskan dari saat karakter terakhir dari perintah dimasukkan oleh program atau transaksi sampai hasil pertama muncul di jperangkat masukan keluaran seperti layar (terminal). Waktu tanggap untuk sistem interaktif biasa disebut *terminal response time*.

- Sistem waktu nyata (*real time*)

Pada sistem waktu nyata, waktu tanggap didefinisikan sebagai waktu dari saat kemunculan suatu kejadian (internal/eksternal) sampai instruksi pertama rutin layanan terhadap kejadian dieksekusi. Waktu untuk sistem waktu nyata biasa disebut *event response time*.

Sasaran penjadwalan adalah meminimalkan waktu tanggap sehingga menghasilkan sistem yang responsif.

Turn *around* *Time*

Waktu yang dihabiskan dari saat proses atau job mulai masuk ke sistem sampai proses itu diselesaikan sistem. Waktu yang dimaksud adalah waktu yang dihabiskan proses berada di sistem, diekspresikan sebagai penjumlahan waktu eksekusi (waktu layanan proses/job) dan waktu menunggu dari proses itu, yaitu :

$\text{Turn around time} = \text{waktu eksekusi} + \text{waktu menunggu}.$

Sasaran penjadwalan adalah meminimalkan *turn around time*.

Troughput

Troughput adalah jumlah kerja yang dapat diselsesaikan selama satu selang/ unit waktu. Cara untuk mengekspresikan *throughput* adalah dengan jumlah proses/job pemakai yang dapat dieksekusi dalam satu unit/ interval waktu tertentu. Sasaran penjadwalan adalah memaksimalkan jumlah job/ proses yang dilayani per satu interval waktu. Lebih tinggi angka *throughput* maka lebih banya kerja yang dilakukan sistem.

Kriteria tsb saling bergantung dan dapat saling bertentangan sehingga tidak dimungkinkan optimasi semua kriteria secara simultan.

Tipe-Tipe Penjadwalan

Dapat terdapat 3 tipe penjadwal berada secara bersama-sama pada sistem operasi yang kompleks, yaitu :

1. Penjadwal jangka pendek (*short-term scheduler*). Penjadwalan jangka pendek bertugas menjadwalkan alokasi pemroses di antara proses-proses Ready yang berada di memori utama. sasaran utama penjadwal jangka pendek adalah memaksimumkan kinerja sistem untuk memenuhi satu kumpulan kriteria yang diharapkan. Penjadwal ini dijalankan setiap terjadi pengalihan proses untuk

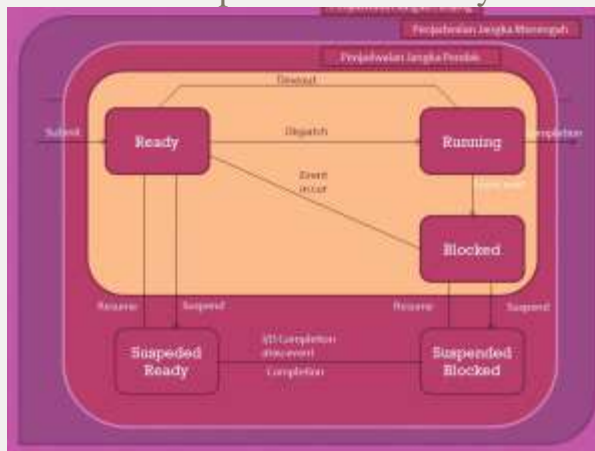
memilih proses berikutnya yang harus



Gambar. Postel Tipe-Tipe Penjadwalan yang dapat terdapat di satu sistem operasi

dijalankan.

2. Penjadwal jangka menengah (*medium-term scheduler*). Setelah eksekusi selama suatu waktu, proses mungkin ditunda karena permintaan layanan masukan/keluaran atau memanggil suatu system call. Proses-proses yang tertunda tidak dapat membuat suatu kemajuan untuk menuju selesai sampai kondisi yang menyebabkannya hilang. Agar ruang memori dapat bermanfaat maka proses dipindah dari memori utama ke memori sekunder sehingga tersedia ruang yang lebih besar untuk proses yang lain. Kapasitas memori utama terbatas untuk sejumlah proses yang aktif. Aktivitas pemindahan proses yang tertunda dari memori utama ke memori sekunder disebut *swapping*. Penjadwal jangka menengah bertugas menangani proses swapping. Proses yang mempunyai kepentingan kecil saat itu adalah proses yang tertunda. Tetapi begitu kondisi yang membuat proses tertunda hilang dan proses dimasukkan kembali ke memori utama dan Ready. Penjadwal jangka menengah mengendalikan transisi dari suspended ke ready (dari state suspend ke Ready dari proses yang mengalami



swapping).

3. Penjadwal jangka panjang (*long-term scheduler*). Penjadwal jangka panjang bekerja terhadap antrian *batch* dan memilih batch berikutnya yang harus dieksekusi sistem. Batch biasanya berupa proses-proses dengan penggunaan sumber daya yang intensif (yaitu waktu pemroses, memori, perangkat masukan/keluaran), program ini mempunyai prioritas yang rendah, dan biasa digunakan sebagai pengisi (agar pemroses sibuk) selama periode aktivitas proses-proses interaktif rendah. Sasaran utama penjadwal jangka panjang adalah memberi keseimbangan proses-proses campuran. Tipe-tipe penjadwal dapat dikaitkan

1. RR (*Round-Robin*)
2. MFQ (*Multiple Feedback Queues*)
3. SRF (*Shortest-Remaining-First*)

4. HRN (*Highest-Remaining-Next*)
5. PS (*Priority Schedulling*)
6. GS (*Guaranteed Schedulling*)

Klasifikasi lain selain berdasarkan dapat/tidaknnya suatu proses diambil alih secara paksa adalah klasifikasi yang berdasarkan adanya prioritas diproses-proses, yaitu :

1. Algoritma penjadwalan tanpa berprioritas
2. Algoritma penjadwalan berprioritas, terdiri dari :
 - Algoritma penjadwalan berprioritas statis
 - Algoritma penjadwalan berprioritas dinamis

Algoritma-Algoritma Penjadwalan Proses

1. Penjadwalan Round-Robin (RR)
2. Penjadwalan FIFO (FIFO)
3. Penjadwalan Berprioritas (PS)
4. Penjadwalan yang Terpendek yang Lebih Dahulu (SJF)
5. Penjadwalan dengan Banyak Antrian (MFQ)
6. Penjadwalan dengan Sisa Waktu Terpendek, Lebih Dahulu (SRF)
7. Penjadwalan Rasio Tanggapan Tertinggi, Lebih Dahulu(HRN)
8. Penjadwalan Terjamin (GS)

Penjadwalan Round Robin

Penjadwalan Round Robin merupakan

- Penjadwalan Preemptive, namun proses tidak di-preempt secara langsung oleh proses lain, namun oleh penjadwal berdasarkan lama waktu berjalannya suatu proses. Maka penjadwalan ini disebut *preempt-by-time*
- Penjadwalan tanpa prioritas

Semua proses dianggap penting dan diberi jumlah waktu pemroses yang disebut kwanta (*quantum*) atau *time-slice* tempat proses tsb berjalan. Proses berjalan selama 1 kwanta, kemudian penjadwal akan mengalihkan kepada proses berikutnya juga untuk berjalan satu kwanta, begitu seterusnya sampai kembali pada proses pertama dan berulang.

Ketentuan algoritma round robin adalah sbb :

1. Jika kwanta habis dan proses belum selesai maka proses Running menjadi Ready dan pemroses dialihkan ke proses lain
2. Jika kwanta belum habis dan proses menunggu suatu kejadian (misalnya menunggu selesainya suatu operasi I/O), maka proses Running menjadi Blocked dan pemroses dialihkan ke proses lain.
3. Jika kwanta belum habis tapi proses telah selesai maka proses Running itu diakhiri dan pemroses dialihkan ke proses lain

Algoritma penjadwalan ini dapat diimplementasi sbb :

- Sistem mengelola senarai proses Ready sesuai urutan kedatangannya
- Sistem mengambil proses yang berada di ujung depan antrian Ready menjadi Running
- Bila kwantra belum habis dan proses selesai maka sistem mengambil proses di ujung depan antrian proses Ready
- Jika kwantra habis dan proses belum selesai maka ditempatkan proses Running ke ekor antrian proses Ready dan sistem mengambil proses di ujung depan antrian proses Ready

Masalah penjadwalan ini adalah dalam hal menentukan besar kwantra, yaitu :

- Kwantra terlalu besar menyebabkan waktu tanggap besar dan turn around time rendah
- Kwantra terlalu kecil mengakibatkan peralihan proses terlalu banyak sehingga menurunkan efisiensi pemroses

Harus diterapkan besar kwantra waktu yang optimal berdasarkan kebutuhan sistem, terutama dari hasil percobaan atau data historis dari sistem. Besar kwantra waktu beragam yang bergantung beban sistem. Berdasarkan kriteria penilaian penjadwalan

- Fairness, penjadwalan RR adil bila dipandang dari persamaan pelayanan oleh pemroses
- Efisiensi, penjadwalan ini cenderung efisien pada sistem interaktif
- *Respons Time* (waktu tanggap), penjadwalan ini memuaskan untuk sistem interaktif, tidak memadai untuk sistem waktu nyata. *Turn around Time*, penjadwalan RR cukup bagus
- *Throughput*, penjadwalan RR cukup bagus

Penjadwalan FIFO

Penjadwalan FIFO merupakan :

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

Penjadwal FIFO adalah penjadwalan dengan ketentuan-ketentuan paling sederhana, yaitu :

- Proses-proses diberi jatah waktu pemroses diurutkan berdasarkan waktu kedatangan proses-proses itu ke sistem.
- Begitu proses mendapat jatah waktu pemroses, proses dijalankan sampai selesai

Penjadwalan ini dikatakan adil dalam arti resmi, tapi dikatakan tidak adil karena proses yang memerlukan waktu lama membuat proses pendek menunggu. Proses tidak

penting dapat membuat proses penting menjadi menunggu. FIFO jarang digunakan secara mandiri tapi dikombinasikan dengan skema lain, misalnya keputusan berdasarkan prioritas proses, sedangkan untuk proses berprioritas sama diputuskan berdasarkan FIFO.

Berdasarkan kriteria penilaian penjadwalan :

- *Fairness*, penjadwalan FIFO adil dalam arti resmi
- Efisiensi, FIFO sangat efisien dalam penggunaan pemroses
- Waktu tanggap, penjadwalan sangat tidak memuaskan karena proses dapat menunggu lama. Tidak cocok untuk sistem interaktif *Turn around time*, penjadwalan FIFO tidak bagus
- *Throughput*, penjadwalan FIFO tidak bagus.

Penjadwalan Berprioritas

Gagasan penjadwalan adalah masing-masing proses diberi prioritas dan proses berprioritas tertinggi menjadi Running (yaitu mendapat jatah waktu pemroses).

Prioritas dapat diberikan secara

- Prioritas statis (static priorities), prioritas tak berubah.

keunggulan : Mudah diimplementasikan dan mempunyai overhead relatif kecil

kelemahan : penjadwalan prioritas statis tidak tanggap perubahan lingkungan yang mungkin menghendaki penyesuaian prioritas

- Prioritas dinamis (dynamic priorities), mekanisme menanggapi perubahan lingkungan sistem saat beroperasi di lingkungan nyata. Prioritas awal yang diberikan ke proses mungkin hanya berumur pendek. Dalam hal ini sistem dapat menyesuaikan nilai prioritasnya ke nilai yang lebih tepat sesuai lingkungan.

keunggulan : waktu tanggap sistem yang bagus

kelemahan : implementasi mekanisme prioritas dinamis lebih kompleks dan mempunyai *overhead* yang lebih besar dibanding mekanisme prioritas statik.

Contoh penjadwalan berprioritas

Proses-proses yang sangat banyak operasi masukan/keluaran dan menghabiskan kebanyakan waktu proses untuk menunggu selesainya operasi masukan/ keluaran. Proses demikian disebut I/O bound process. Proses-proses ini dapat diberi prioritas sangat tinggi sehingga begitu proses-proses memerlukan pemroses, segera saja diberikan dan proses akan segera memulai permintaan masukan/keluaran berikutnya sehingga menyebabkan proses Blocked menunggu selesainya operasi masukan/keluaran. Dengan demikian pemroses segera dialihkan, dapat dipergunakan oleh proses lain tanpa mengganggu proses I/O bound. Proses I/O bound berjalan paralel bersama proses lain yang benar-benar memerlukan pemroses.

Proses-proses yang sangat banyak operasi masukan/keluaran jika harus menunggu lama untuk memakai pemroses(karena diberi prioritas rendah) hanya akan membebani memori, karena sistem harus menyimpan tanpa perlu proses-proses itu di memori karena tidak selesai-selesai menunggu operasi masukan/keluaran dan menunggu jatah pemroses.

Algoritma Prioritas Dinamis
Algoritma ini dituntun oleh keputusan untuk memenuhi kebijaksanaan tertentu yang menjadi tujuan sistem komputer. Algoritma sederhana yang memberi layanan yang baik adalah dengan menge-set proses dengan prioritas berdasarkan rumus nilai $1/f$ bahwa f adalah rasio kwanta terakhir yang digunakan proses.

- Proses yang menggunakan 2 milidetik, kwanta 100 ms maka prioritasnya 50
- Proses yang berjalan selama 50 milidetik sebelum Blocked berprioritas 2
- Proses yang menggunakan seluruh kwanta berprioritas 1

Kebijaksanaan yang diterapkan adalah jaminan proses-proses mendapat layanan yang adil dari pemroses dalam arti jumlah waktu pemroses yang sama untuk masing-masing pemroses pada satu waktu. Biasanya memenuhi kebijaksanaan yang ingin mencapai level maksimal berdasarkan suatu kriteria tertentu di sistem.

Algoritma penjadwalan berprioritas dapat dikombinasikan yaitu dengan mengelompokkan proses-proses menjadi kelas-kelas prioritas. Penjadwalan berprioritas diterapkan antar kelas-kelas proses itu. Penjadwalan round-robin atau penjadwalan FIFO diterapkan pada proses-proses di dalam satu kelas.

Penjadwalan yang Terpendek yang Lebih Dahulu (SJF)

Penjadwalan SJF ini merupakan

- Penjadwalan non preemptive
- Penjadwalan dapat dikatakan sebagai berprioritas. Di SJF, prioritas diasosiasikan dengan masing-masing proses dan pemroses dialokasikan ke proses dengan prioritas tertinggi. Proses-proses dengan prioritas yang sama akan dijadwalkan secara FIFO.

Penjadwalan ini mengasumsikan waktu jalan proses (sampai selesai) atau waktu lamanya proses diketahui sebelumnya. Mekanisme penjadwalan SJF adalah lebih dulu menjadwalkan proses dengan waktu jalan terpendek sampai selesai. Setelah proses itu selesai, maka proses dengan waktu jalan terpendek berikutnya dijadwalkan. Demikian seterusnya.

Keunggulan : penjadwalan SJF mempunyai efisiensi tinggi dan turn around time rendah.

Contoh : Terdapat 4 proses A,B,C,D dengan waktu jalan selama 8,7,6,5 kwanta. Gambar (a) menunjukkan penjadwalan cara I, dengan proses-proses dijadwalkan berurutan sebagai A,B,C,D. Gambar (b) menunjukkan bila proses dijadwalkan secara SJF yaitu berurutan D,C,B,A.

Cara I (a)				Cara II (b)			
8	7	6	5	5	6	7	8
Proses	Turn Around Cara I			Turn Around Cara II			
A	8			26			
B	15			18			
C	21			11			
D	26			5			
Rata-rata	17,5			15			

Kedua cara menghasilkan *turn around time* yang ditunjukkan pada gambar (c). Cara I *turn around time* rata-rata adalah 17,5 kwanta, sedangkan cara II adalah 15 kwanta

Walaupun mempunyai *turn around* yang bagus, SJF mempunyai masalah, yaitu

- Tidak dapat mengetahui ukuran proses saat proses masuk
- Proses tidak datang bersamaan sehingga penetapannya harus dinamis

Untuk mengetahui ukuran lama proses agar dapat ditetapkan yang terpendek, biasanya dilakukan dengan cara pendekatan. Pendekatan yang biasa dilakukan adalah dengan membuat estimasi berdasarkan perilaku historis sistem. Merupakan kajian teoritis untuk perbandingan dalam perbandingan *turn around time*.

Penjadwal dengan Banyak Antrian (MFQ)

Penjadwalan MFQ ini merupakan

- Penjadwalan preemptive
- Penjadwalan berprioritas dinamis

Sasaran penjadwalan ini adalah untuk mencegah banyaknya aktivitas swapping. Cara yang dilakukan adalah dengan

1. Proses-proses yang sangat banyak menggunakan pemroses (karena menyelesaikan tugasnya memakan waktu yang lama) diberi jatah waktu (jumlah kwanta) lebih banyak dalam satu waktu.
2. Penjadwalan ini menghendaki kelas prioritas bagi proses-proses yang ada. Kelas tertinggi berjalan selama satu kwanta, kelas berikutnya berjalan selama dua

kwanta, kelas berikutnya lagi berjalan empat kwanta, kelas berikutnya-berikutnya lagi berjalan delapan kwanta dan seterusnya.

Ketentuan yang berlaku adalah sebagai berikut :

- Jalankan proses-proses yang berada pada kelas prioritas tertinggi
- Jika proses telah menggunakan seluruh kwanta yang dialokasikan maka proses itu diturunkan kelas prioritasnya
- Proses yang masuk untuk pertama kali ke sistem langsung diberi kelas tertinggi

Penjadwalan dengan Sisa Waktu Terpendek, Lebih Dahulu (SRF)

Penjadwalan ini merupakan

- Penjadwalan preemptive
- Penjadwalan berprioritas dinamis

Penjadwalan SRF merupakan perbaikan dari SJF, SJF merupakan penjadwalan *nonpreemptive* sedang SRF adalah preemptive yang dapat digunakan untuk sistem timesharing.

Pada SRF, proses dengan sisa waktu jalan diestimasi terendah dijalankan, termasuk proses-proses yang baru tiba.

Perbedaan SRF dengan SJF

- Pada SJF, begitu proses dieksekusi, proses dijalankan sampai selesai
- Pada SRF proses yang sedang berjalan (Running) dapat diambil alih oleh proses baru dengan sisa waktu jalan yang diestimasi lebih rendah

SRF mempunyai overhead yang lebih besar dibanding SJF. SRF memerlukan penyimpanan waktu layanan yang telah dihabiskan proses dan kadang-kadang harus menangani peralihan.

- Tibanya proses-proses kecil akan segera dijalankan
- Proses-proses lebih lama berarti dengan lama dan variasi waktu tunggu lebih lama dibanding dengan SJF

Secara teoretis, SRF memberi waktu tunggu minimum tapi karena adanya overhead peralihan, maka pada situasi tertentu SJF bisa memberi kinerja yang lebih baik dibanding SRF.

Penjadwalan Rasio Tanggapan Tertinggi, Lebih Dahulu (HRN)

Penjadwalan HRN ini merupakan :

- Penjadwalan *non preemptive*
- Penjadwalan berprioritas dinamis

Penjadwalan ini juga untuk mengoreksi kelemahan SJF. HRN adalah strategi penjadwalan *non preemptive* dengan prioritas proses tidak hanya merupakan fungsi

dari waktu layanan, tapi juga jumlah waktu tunggu proses. Prioritas dinamis HRN dihitung berdasarkan rumus berikut :

Prioritas = (waktu tunggu + waktu layanan) / waktu layanan

Karena waktu layanan muncul sebagai pembagi maka proses yang lebih pendek mempunyai prioritas yang lebih baik. Karena waktu tunggu sebagai pembilang maka proses yang telah menunggu lebih lama juga mempunyai kesempatan lebih bagus untuk memperoleh layanan pemroses. Disebut HRN (*High respons next*) karena waktu tanggap adalah (waktu tunggu + waktu layanan). Ketentuan HRN berarti agar memperoleh waktu tanggap tertinggi yang harus dilayani.