

Penjadwalan Tugas

“Multiple Feedback Queues”

Teknik Informatika 1

Ahmad Fathan Syakir

Atrisa Endya Nur Hidayah

Muhammad Adil Nashrul Haq

Muhammad Azhar Rasyad

Sekolah Tinggi Teknologi Terpadu Nurul Fikri
2018

A. Definisi

Salah satu penjadwalan yang digunakan untuk menyelesaikan process-process adalah algoritma penjadwalan Multilevel Feedback Queue(MLFQ). MLFQ menerapkan beberapa algoritma-algoritma yaitu algoritma First Come First Serve (FCFS), Round Robin(RR), Priority Scheduler(PS) dan Multilevel Queue(MLQ) dan aturan perpindahan kebawah level antrian persatu antrian.

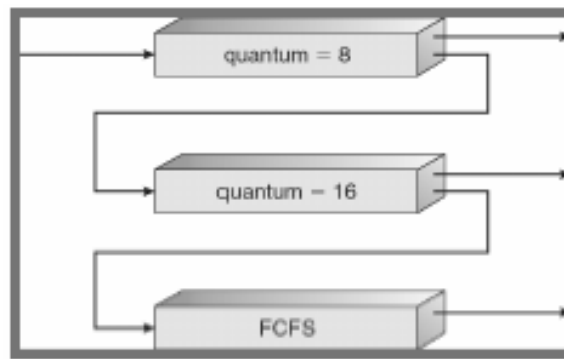
Algoritma Penjadwalan Multilevel Feedback Queue (MLFQ) adalah algoritma yang memiliki urutan level antrian dan proses dapat berpindah dari level antrian awal ke level-level antrian berikutnya. Algoritma MLFQ memiliki beberapa antrian secara urutan jumlah nilai waktu jatah (QuantumTime) berbeda. Secara urutan waktu jatah antrian level terkecil memiliki waktu jatah lebih kecil dari waktu jatah antrian lainnya.

Penyelesaian secara multi level antrian membuat proses yang mempunyai waktu pengerjaan ($\text{BurstTime} / \text{ServiceTime}$) besar akan memiliki pecahan-pecahan BurstTime dari QuantumTime di setiap level antrian. Sehingga mencari waktu tunggu dan waktu keseluruhan sulit diterapkan pada algoritma MLFQ. pendefinisian pecahan BurstTime dalam bentuk variabel-variabel Pengerjaan (Running Process).

Algoritma ini mirip sekali dengan algoritma Multilevel Queue. Perbedaannya ialah algoritma ini mengizinkan proses untuk pindah antrian. Jika suatu proses menyita CPU terlalu lama, maka proses itu akan dipindahkan ke antrian yang lebih rendah. Ini menguntungkan proses interaksi, karena proses ini hanya memakai waktu CPU yang sedikit. Demikian pula dengan proses yang menunggu terlalu lama.

Proses ini akan dinaikkan tingkatannya. Biasanya prioritas tertinggi diberikan kepada proses dengan CPU burst terkecil, dengan begitu CPU akan dimanfaatkan penuh dan I/O dapat terus sibuk. Semakin rendah tingkatannya, panjang CPU burst proses juga semakin besar. Algoritma ini didefinisikan melalui beberapa parameter, antara lain:

1. Jumlah antrian
2. Algoritma penjadwalan tiap antrian
3. Kapan menaikkan proses ke antrian yang lebih tinggi
4. Kapan menurunkan proses ke antrian yang lebih rendah
5. Antrian mana yang akan dimasuki proses yang membutuhkan



Dengan pendefinisian seperti tadi membuat algoritma ini sering dipakai. Karena algoritma ini mudah dikonfigurasi ulang supaya cocok dengan sistem. Tapi untuk mengetahui mana penjadwal terbaik, diharuskan mengetahui nilai parameter tersebut. Multilevel feedback queue adalah salah satu algoritma yang berdasar pada algoritma multilevel queue.

Perbedaan mendasar yang membedakan multilevel feedback queue dengan multilevel queue biasa adalah terletak pada adanya kemungkinan suatu proses berpindah dari satu antrian ke antrian lainnya, entah dengan prioritas yang lebih rendah ataupun lebih tinggi, misalnya pada contoh berikut.

1. Semua proses yang baru datang akan diletakkan pada antrian 0 (quantum = 8 ms)
2. Jika suatu proses tidak dapat diselesaikan dalam 8 ms, maka proses tersebut akan dihentikan dan dipindahkan ke antrian pertama (quantum = 16 ms)
3. Antrian pertama hanya akan dikerjakan jika tidak ada lagi proses di antrian 0, dan jika suatu proses di antrian pertama 1 tidak selesai dalam 16 ms, maka proses tersebut akan dipindahkan ke antrian kedua
4. Antrian kedua akan dikerjakan bila antrian 0 dan 1 kosong, dan akan berjalan dengan algoritma Round Robin.

B. Analisa

Langkah-langkah yang dilakukan dalam analisa menentukan waktu tunggu dan waktu keseluruhan dalam penjadwalan MLFQ yaitu :

1. Mendefenisikan rumus running process untuk ketentuan dalam penghitungan waktu tunggu.
2. Waktu tunggu dijumlahkan dengan waktu layanan untuk mendapatkan waktu keseluruhan.

1. Analisa Waktu Tunggu (WaitingTime)

Setiap proses akan memiliki waktu tunggu yang didapatkan dari waktu layanan proses yang mempengaruhi ditambah dengan waktu kedatangan proses yang dicari.

Ketentuan proses yang mempengaruhi yaitu:

- a. Proses yang dikerjakan waktu layanannya digantikan oleh proses lain. Maka proses yang menggantikan disebut sebagai proses yang mempengaruhi, jika proses yang mempengaruhi juga terpengaruhi atau menunggu karena proses lain maka terdapat lebih dari satu proses yang mempengaruhi proses yang berjalan tersebut.
- b. Proses belum dikerjakan pada waktu kedatangannya.

Contoh problema dalam waktu tunggu :

1. P0 dikerjakan pertama kali dan tidak ada proses yang mempengaruhi, maka kondisi proses yang mempengaruhi Null (0).
2. P1 dikerjakan dan digantikan P2, maka P1 dipengaruhi P2.
3. P1 dikerjakan dan digantikan P2 dan juga digantikan P3, maka P1 dipengaruhi P2 dan P3.
4. P2 belum dikerjakan pada waktu kedatangannya dan pada waktu itu P1 sedang dikerjakan, maka P2 dipengaruhi P1.
5. P2 belum dikerjakan pada waktu kedatangannya dan pada saat itu P1 sedang dikerjakan, sementara sebelumnya P1 juga dipengaruhi P0, maka P2 dipengaruhi P1 dan P0.
6. P1 digantikan P2. Setelah P2 selesai P1 dikerjakan kembali dan pada waktu pengerjaan selanjutnya berjalan P1 digantikan P3, maka P1 dipengaruhi P2 dan P3.

Penyederhanaan problema waktu tunggu. Dalam menghitung waktu tunggu proses diketahui dari proses-proses yang mempengaruhinya.

1. Dari problema no. 1
 $WTP_0 = 0$
 Karena tidak ada yang mempengaruhi P0
2. Dari problema no. 2
 $WTP_1 = BTP_2 - ATP_1$
3. Dari problema no. 3
 $WTP_1 = (BTP_2 + BTP_3) - ATP_1$
4. Dari problema no. 4
 $WTP_2 = BTP_1 - ATP_2$
5. Dari problema no. 5
 $WTP_2 = (BTP_0 + BTP_1) - ATP_2$
6. Dari problema no. 6
 $WTP_1 = (BTP_2 + BTP_3) - ATP_1$

2. Running Process (RP)

Running Process mempermudah penghitungan waktu tunggu karena pada MLFQ diketahui process dikerjakan berdasarkan QuantumTime (QT). Ketentuan pengerjaan Running Process yaitu dengan

pembagian BurstTime dibagi QuantumTime. Hasil Bagi untuk perulangan QuantumTime sedangkan Sisa Bagi untuk pengerjaan selanjutnya yang kurang dari QuantumTime:

Adapun Running Process sebagai berikut:

1. Hasil bagi → perulangan QuantumTime
2. Sisa bagi → Running process selanjutnya
3. Jika BurstTime < QuantumTime maka Running Process adalah BurstTime process tersebut.

3. Analisa Running Process Algoritma

MLFQ Algoritma MLFQ memiliki berapa antrian. Setiap process akan dikerjakan pada antrian pertama jika BurstTime proses lebih besar dari QuantumTime pertama maka akan dikerjakan satu kali dan selebihnya dikerjakan pada antrian selanjutnya. Jika terdapat tiga antrian maka ketentuan tersebut berlaku pada antrian pertama dan kedua. Pada antrian ketiga yang merupakan antrian terakhir berlaku ketentuan Running Process Pembagian.

C. Pembahasan

Penyelesaian proses-proses dalam penjadwalan MLFQ.

1. Process

Tabel 1. Urutan proses-proses

Process	ArrivalTime	BurstTime
P1	0	6
P2	2	3
P3	5	4
P4	7	11

2. Level Antrian dan Quantum Time

Tabel 2. Nilai Quantum Time setiap Queue

Queue	QuantumTime
Q0	2
Q1	3
Q2	5

3. Running Process

P1 di Q1:

RP1Q1 \rightarrow BTP1 > QT1
 $\rightarrow 6 > 2$
 \rightarrow ya, kerjakan sebanyak 2
RP1Q1 = 2
Sisa = 4, dipindahkan ke Q2

P1 di Q2 :

RP1Q1 \rightarrow SBTP1 > QT2
 $\rightarrow 4 > 3$
 \rightarrow ya, kerjakan sebanyak 3
RP1Q2 = 3
Sisa = 1, dipindahkan ke Q3

P1 di Q3 :

RP1Q3 \rightarrow SBTP1 < Q3
 $\rightarrow 1 < 5$
 \rightarrow ya
RP1Q3(1) = 1

Jadi : RP1Q1 = 2, RP1Q2 = 3, RP1Q3(1) = 1

P2 di Q1:

RP2Q1 \rightarrow BTP2 > QT1
 $\rightarrow 3 > 2$
 \rightarrow ya, kerjakan sebanyak 2
RP2Q1 = 2
Sisa = 1, dipindahkan ke Q2

P2 di Q2 :

RP1Q1 \rightarrow SBTP2 > QT2
 $\rightarrow 1 > 3$
 \rightarrow tidak, kerjakan hanya sebanyak 1
RP1Q2 = 1

Jadi : RP2Q1 = 2, RP2Q2 = 2

P3 di Q1:

RP3Q1 \rightarrow BTP3 > QT1
 $\rightarrow 4 > 2$
 \rightarrow ya, kerjakan sebanyak 2
RP1Q1 = 2
Sisa = 2, dipindahkan ke Q2

P1 di Q2 :

RP3Q1 \rightarrow SBTP3 > QT2
 $\rightarrow 2 > 3$
 \rightarrow tidak, kerjakan hanya sebanyak 2
RP3Q2 = 2

Jadi : RP3Q1 = 2, RP3Q2 = 2

P4 di Q1:

RP4Q1 → BTP4 > QT1
 → 11 > 2
 → ya, kerjakan sebanyak 2

RP4Q1 = 2

Sisa = 9, dipindahkan ke Q2

P4 di Q2 :

RP4Q1 → SBTP4 > QT2
 → 9 > 3
 → ya, kerjakan sebanyak 3

RP4Q2 = 3

Sisa = 6, dipindahkan ke QT3

P1 di Q3 :

RP4Q3 → SBTP4 < Q3
 → 1 < 5
 → tidak

Maka

RP4Q3 = SBTP4/QT3
 = 6 / 5
 = 1, sisa 1

Hasil 1 perulangan sebanyak QT3

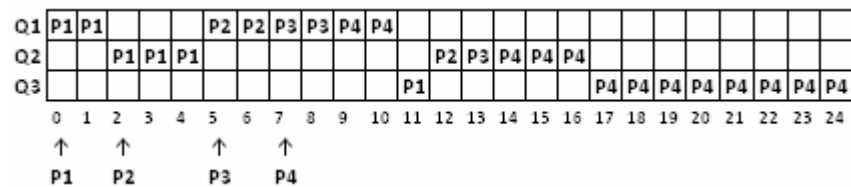
→ RP4Q3(1) = 5

→ RP4Q3(2) = 1

Jadi :

RP4Q1 = 2, RP4Q2 = 3, RP4Q3(1) = 5, RP4Q3(1)
 = 1

4. Diagram Gantt



5. Indeks proses sesuai ArrivalTime

Tabel 3. Indeks process

No.	Process yang dilayani			
1	P1			
2	P1	P2	P3	P4
3	P1	P4		

6. Waktu Tunggu

$$\begin{aligned} \text{WTP1} &= (\text{RP2Q1} + \text{RP3Q1} + \text{RP4Q1}) - \text{ATP1} \\ &= (2 + 2 + 2) - 0 \\ &= 6 - 0 \\ &= 6 \text{ ms} \end{aligned}$$

$$\begin{aligned} \text{WTP2} &= (\text{RP1Q1} + \text{RP1Q2} + \text{RP3Q1} + \text{RP4Q1} + \text{RP1Q3}_{(1)}) - \text{ATP1} \\ &= (2 + 3 + 2 + 2 + 1) - 2 \\ &= 10 - 2 \\ &= 8 \text{ ms} \end{aligned}$$

$$\begin{aligned} \text{WTP3} &= (\text{RP1Q1} + \text{RP1Q2} + \text{RP2Q1} + \text{RP4Q1} + \text{RP1Q3}_{(1)} + \text{RP2Q2}) - \text{ATP1} \\ &= (2 + 3 + 2 + 2 + 1 + 1) - 5 \\ &= 11 - 5 \\ &= 6 \text{ ms} \end{aligned}$$

$$\begin{aligned} \text{WTP4} &= (\text{RP1Q1} + \text{RP1Q2} + \text{RP2Q1} + \text{RP3Q1} + \text{RP1Q3}_{(1)} + \text{RP2Q2}) + \text{RP3Q2} + \text{RP1Q3}_{(1)}) - \text{ATP1} \\ &= (2 + 3 + 2 + 2 + 1 + 1 + 1) - 7 \\ &= 12 - 7 \\ &= 5 \text{ ms} \end{aligned}$$

7. Waktu Keseluruhan

$$\begin{aligned} \text{TAP1} &= \text{BTP1} + \text{WTP1} \\ &= 6 + 6 \\ &= 12 \text{ ms} \end{aligned}$$

$$\begin{aligned} \text{TAP2} &= \text{BTP2} + \text{WTP2} \\ &= 3 + 8 \\ &= 11 \text{ ms} \end{aligned}$$

$$\begin{aligned} \text{TAP3} &= \text{BTP3} + \text{WTP3} \\ &= 4 + 6 \\ &= 10 \text{ ms} \end{aligned}$$

$$\begin{aligned} \text{TAP4} &= \text{BTP4} + \text{WTP4} \\ &= 11 + 5 \\ &= 15 \text{ ms} \end{aligned}$$

D. Kesimpulan

Berdasarkan studi dan penelitian yang dilakukan maka dapat disimpulkan yaitu :

1. Algoritma Multi Level Feedback Queue (MLFQ) merupakan algoritma penjadwalan yang baik dalam mengoptimalkan proses-proses yang memiliki prioritas, karena dapat memindahkan proses yang memiliki waktu layanan (BurstTime) yang besar.
2. Running Process yaitu waktu layanan dibagi waktu jatah pengerjaan atau QuantumTime (QT) sehingga hasil bagi digunakan untuk banyak perulangan QT dan sisa bagi untuk banyak pengerjaan berikutnya tapi kurang dari QT.
3. Penerapan Running Process (RP) memudahkan penentuan waktu tunggu, karena dapat menggunakan RP dalam mengetahui proses- proses yang mempengaruhi dalam diagram gantt penjadwalan MLFQ.

E. Saran

Beberapa saran yang dapat dipertimbangkan untuk yaitu.

1. Mengimplementasikan analisa algoritma MLFQ ke dalam algoritma pembuatan program pembelajaran dan penyelesaian problema yang menyelesaikan antrian.
2. Analisa algoritma MLFQ ini dapat dikembangkan lagi dalam penentuan waktu respon tiap proses.
3. Process dan Queue diharapkan dapat lebih banyak lagi agar mekanisme penjadwalan Algoritma MLFQ lebih jelas.

Referensi

- https://www.researchgate.net/publication/317564529_IMPLEMENTASI_ALGORITMA_MULTILEVEL_FEEDBACK_QUEUE_DALAM_MENENTUKAN_WAKTU_TUNGGU_DAN_WAKTU_KESELURUHAN_PROSES/download