JETI
Jurnal Elektronika dan Teknologi Informasi, **Vol. I No. 2** õ **September 2020** õ **e-ISSN: 2721-9380**

OPTIMASI ALOKASI WAKTU PROSES CPU PADA LINUX DENGAN ALGORITMA PRIORITY SCHEDULING

Mutasar[1],
[1] Fakultas Komputer dan Multimedia, Universitas Islam Kebangsaan Indonesia (UNIKI) Aceh

ABSTRAK

Penjadwalan berkaitan dengan permasalahan memutuskan proses mana yang akan dilaksanakan dalam suatu sistem. Proses yang belum mendapat jatah alokasi dari CPU akan mengantri di *ready queue*. Algoritma penjadwalan berfungsi untuk menentukan proses manakah yang ada di *ready queue* yang akan dieksekusi oleh CPU.

Kata Kunci: Optimasi Alokasi Waktu, Algoritma Priority Scheduling

ABSTRACT

Scheduling is related to the problem of deciding which processes will be carried out in a system. Processes that have not received the allocation from the CPU will queue in the ready queue. The scheduling algorithm is used to determine which processes are in the ready queue to be executed by the CPU.

Keywords: *Time Allocation Optimization, Priority Scheduling Algorithm*

1. PENDAHULUAN

Linux memiliki dua algoritma proses penjadwalan. Yang pertama adalah *time-sharing* algoritma untuk penjadwalan yang adil dan preemptive diantara beberapa proses, sedangkan yang lain di desain untuk tugas-tugas realtime, dimana absolut prioritas adalah hal yang paling penting.

Algortima penjadwalan Linux adalah preemptive, berdasarkan prioritas yang ada dengan dua range prioritas yang terpisah (range real time dari 0-99, dan range lainnya dari 100-140). Dua range ini memetakan skema prioritas global dimana semakin kecil angka prioritasnya semakin tinggi prioritasnya.

Linux memberikan quantum waktu yang lebih panjang pada proses dengan prioritas tinggi dan sebaliknya. Sebuah proses dapat berjalan pada CPU jika proses tersebut memiliki waktu sisa pada slot waktu. Saat slot waktunya habis, proses tersebut dianggap kadaluarsa dan tidak akan dieksekusi sampai seluruh proses slot waktunya habis juga. Saat ini terjadi, list proses aktif akan menjadi kosong, maka list proses kadaluarsa akan menjadi aktif dan eksekusi akan dimulai kembali.

Penjadwalan real-time Linux bersifat soft real-time. Pada algoritma soft real-time, fitur yang paling penting adalah merespon dengan segera sebuah proses real-time secepat proses yang dibutuhkan oleh CPU. Algoritma penjadwalan berdasarkan prioritas memberikan prioritas kepada masing-masing proses berdasarkan tingkat kepentingannya, proses yang lebih penting di berikan prioritas lebih tinggi daripada proses lain yang dianggap kurang penting. Apabila penjadwalan yang digunakan juga mendukung preemption dan tersedia sebuah proses berprioritas tinggi, maka proses ini akan dijalankan lebih dahulu mengingat prioritasnya yang tinggi

2. METODE PENELITIAN

Optimasi adalah suatu proses untuk mencapai hasil yang ideal atau optimasi (nilai efektif yang dapat dicapai). Optimasi dapat diartikan sebagai suatu bentuk mengoptimalkan sesuatu hal yang sudah ada, ataupun merancang dan membuat sesusatu secara optimal. Penyelesaian permasalahan dalam teknik optimasi diarahkan untuk mendapatkan titik maksimum atau titik minimimum dari fungsi yang dioptimumkan.

Tujuan dari optimasi adalah untuk meminimumkan usaha yang diperlukan atau biaya operasional dan memaksimumkan hasil yang diinginkan.

Mutasar
Fakultas Komputer dan Multimedia, UNIKI
Email: mutasarstmik@gmail.com

Jika usaha yang diperlukan atau hasil yang diharapkan dapat dinyatakan sebagai fungsi dari peubah keputusan, maka optimasi dapat didefinisikan sebagai proses pencapaian kondisi maksimum dan minimum dari fungsi tersebut. Unsur penting dalam masalah optimasi adalah fungsi tujuan, yang sangat bergantung pada sejumlah peubah masukan. Peubah-peubah ini dapat tidak saling bergantung atau saling bergantung melalui satu atau lebih kendala.

3. HASIL DAN PEMBAHASAN

3.1. Algoritma Priority Scheduling

Priority Scheduling merupakan algoritma penjadwalan yang mendahulukan proses yang memiliki prioritas tertinggi. Setiap proses memiliki prioritasnya masing-masing. *Priority scheduling* juga dapat dijalankan secara *preemptive* maupun *non-preemptive*. [1]

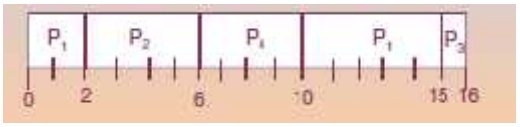
Algoritma ini dapat dibagi menjadi dua bagian yaitu :

1. ***Preemptive*** Jika ada proses yang sedang dieksekusi oleh CPU dan terdapat proses di ready queue dengan burst time yang lebih kecil daripada proses yang sedang dieksekusi tersebut, maka proses yang sedang dieksekusi oleh CPU akan digantikan oleh proses yang berada di *ready queue* tersebut. *Preemptive SJF* sering disebut juga Shortest-Remaining- Time-First scheduling.

Process	Arrival Time	Burst Time	Priority
P_1	0.0	7	3
P_2	2.0	4	1
P_3	4.0	1	3
P_4	5.0	4	2

Gambar 1: waktu eksekusi (premtive)

Priority (*preemptive*), asumsi apabila prioritas sama,diambil yang datang lebih dahulu



Gambar 2: waktu prioritas

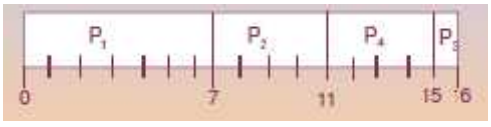
- Waktu tunggu untuk *P1* = 10; *P2* = 0; *P3* = 11; *P4* =1
- Rata-rata waktu tunggu = (10 + 0 + 11 + 1)/4 = 4.5
- Waktu turnaround untuk *P1* = 15; *P2* = 4; *P3* = 12; *P4* = 5
- Rata-rata waktu tunggu = (15 + 4 + 12 +5)/4 = 9

2. ***Non-preemptive*** CPU tidak memperbolehkan proses yang ada di *ready queue* untuk menggeser proses yang sedang dieksekusi oleh CPU meskipun proses yang baru tersebut mempunyai *burst time* yang lebih kecil.

Process	Arrival Time	Burst Time	Priority
P_1	0.0	7	3
P_2	2.0	4	1
P_3	4.0	1	3
P_4	5.0	4	2

Gambar 3: waktu eksekusi (*Nonpremtive*)

Priority (*non-preemptive*)



Gambar 4: waktu prioritas (*Nonpremtive*)

- Waktu tunggu untuk *P1* = 0; *P2* = 5; *P3* = 11; *P4* =6
- Rata-rata waktu tunggu = (0 + 5 + 11 + 6)/4 = 4.5
- Waktu turnaround untuk *P1* = 7; *P2* = 9; *P3* = 12; *P4* = 10
- Rata-rata waktu tunggu = (7 + 9 + 12 +10)/4 = 9.5

3.2. Kelemahan

Kelemahan pada *priority scheduling* adalah dapat terjadinya *indefinite blocking* (*starvation*). Suatu proses dengan prioritas yang rendah memiliki kemungkinan untuk tidak dieksekusi jika terdapat proses lain yang memiliki prioritas lebih tinggi darinya.

3.3. Solusi

Solusi dari permasalahan ini adalah *aging*, yaitu meningkatkan prioritas dari setiap proses yang menunggu dalam *queue* secara bertahap.dapat dilihat pada analisa dibawah ini:

Setiap 10 menit, prioritas dari masing-masing proses yang menunggu dalam *queue* dinaikkan satu tingkat. Maka, suatu proses yang memiliki prioritas 127, setidaknya dalam 21 jam 20 menit, proses tersebut akan memiliki prioritas 0, yaitu prioritas yang tertinggi (semakin kecil angka menunjukkan bahwa prioritasnya semakin tinggi).[2]

4. SIMPULAN

Dengan menggunakan Algoritma priority scheduling maka prioritas yang rendah memiliki kemungkinan untuk tidak dieksekusi jika terdapat proses lain yang memiliki prioritas lebih tinggi darinya yang digunakan dalam pemrosesan CPU.

REFERENCES

[1] Abdalla, H. I., & Amer, A. A. (2012, March). Dynamic horizontal fragmentation, replication and allocation model in DDBSs. In Information Technology and e-Services (ICITeS), 2012 *International Conference on* pp. 1-7, IEEE.

[2] Baker, Kenneth R. 1974. *Introduction to Sequencing and Schedulling*. John Wiley & Sons: USA.

[3] Bhura, S. A., Mahamune, A., & Alvi, A. S. (2015, February). Limited Preemptive Disk Scheduling for Real Time Database System. In Computing Communication Control and Automation (ICCUBEA), 2015 International Conference on (pp. 362-366). IEEE.

[4] D. A. Neamen, *Semiconductor Physics and Devices*, 3rd ed. New York, NY: McGraw-Hill, 2003.

[5] Firoozy-Najafabadi, H. R., & Navin, A. H. (2012, October). Rule scheduling methods in active database systems: A brief survey. In Application of Information and Communication Technologies (AICT), 2012 6th International Conference on (pp. 1-5). IEEE.

[6] Furkhat, T. (2009). Network Issues in Clock Synchronization on Distributed Database. Network, 2(2).J. Zhou, M. Abdel-Mottaleb, "A content-based system for human identification based on bitewing dental X-ray images," *Pattern Recognation*., vol. 38, pp. 2132–2142, 2005

[7] L.Burlion, *et al.*, "Keeping a ground point in the camera field of view of a landing UAV," *IEEE International Conference on Robotics and Automation, Vols 1*, pp. 5763-5768, 2013.