BIIKMA : Buletin Ilmiah Ilmu Komputer dan Multimedia
Volume 2, No. 1, Juni Tahun 2024
ISSN 3024-8248 (media online)
Hal 60-64

Penanganan Deadlock Yang Optimal Dalam Sistem Operasi Windows: Pencegahan, Identifikasi Penyebab, Dan Konsekuensi Deadlock

Sofyan Mufti Prasetiyo[1*], Rehan Gustiawan[2], Farhat[3], Fabian Rizzel Albani[4]

[1,2,3,4]Fakultas Ilmu Komputer, Teknik Informatika, Kota Tanggerang Selatan, Indonesia
Email: [1*]dosen01809@unpam.ac.id, [2]rehangustiawan50@gmail.com, [3]farhatsyawie01@gmail.com, [4]fabianrizzel00@gmail.com
(*: coresponden author: dosen01809@unpam.ac.id)

Abstrak - Deadlock adalah kondisi yang sering terjadi pada sistem operasi, dimana beberapa proses berhenti karena tidak dapat melanjutkan eksekusi karena sumber daya yang dibutuhkan oleh proses lain. Dalam artikel ini, kami akan membahas strategi dan implementasi simulasi pencegahan deadlock pada sistem operasi. Kami akan membahas konsep dasar deadlock, jenis-jenis deadlock, dan strategi pencegahan deadlock yang efektif. Selain itu, kami juga akan membahas implementasi simulasi pencegahan deadlock menggunakan algoritma dan teknologi terkini. Simulasi pencegahan deadlock dilakukan dengan menggunakan model sistem operasi yang kompleks, yang terdiri dari beberapa proses yang berinteraksi dengan sumber daya sistem operasi. Kami menggunakan algoritma pencegahan deadlock yang efektif, seperti algoritma banker dan algoritma *wddy*, untuk mengurangi frekuensi terjadinya deadlock.

Kata Kunci : Deadlock, Sistem Operasi, Windows.

Abstract - *Deadlock is a condition that often occurs in operating systems, where several processes stop because they cannot continue execution because of the resources required by other processes. In this article, we will discuss strategies and implementation of deadlock prevention simulations in operating systems. We will discuss the basic concept of deadlocks, types of deadlocks, and effective deadlock prevention strategies. Apart from that, we will also discuss the implementation of deadlock prevention simulations using the latest algorithms and technology. Deadlock prevention simulations are carried out using a complex operating system model, which consists of several processes that interact with operating system resources. We use effective deadlock prevention algorithms, such as the banker algorithm and the wddy algorithm, to reduce the frequency of deadlocks*

Keywords: *Deadlock, Operating System, Windows.*

1. PENDAHULUAN

Dalam lingkup sistem operasi, salah satu tantangan utama yang sering dihadapi adalah fenomena deadlock. Deadlock adalah keadaan sistem di mana setiap proses dalam suatu grup meminta sumber daya dari proses lain dalam grup dan menunggu tanpa batas waktu hingga permintaan tersebut dipenuhi (Saifulloh & Mumtahana, 2017). Situasi ini dapat mengganggu keseimbangan sumber daya sistem, menurunkan efisiensi operasional, dan menghambat produktivitas pengguna sistem komputer, khususnya dalam konteks Windows.

Untuk mengatasi permasalahan ini, strategi yang efektif dalam pencegahan, identifikasi penyebab, dan penanganan deadlock menjadi sangat penting. Pencegahan deadlock melibatkan penggunaan teknik dan algoritma yang dirancang untuk mencegah kemungkinan terjadinya deadlock. Identifikasi penyebab deadlock membantu dalam memahami kondisi-kondisi spesifik yang memicu terjadinya deadlock dalam sistem operasi Windows. Sementara itu, memahami konsekuensi deadlock menggambarkan dampak negatif yang dapat timbul akibat ketidakmampuan sistem untuk menangani deadlock dengan baik.

Artikel ini akan membahas secara mendalam tentang konsep dasar deadlock, berbagai jenis deadlock yang mungkin terjadi dalam lingkungan Windows, serta strategi dan implementasi simulasi untuk mencegah deadlock. Selain itu, akan dibahas juga mengenai algoritma-algoritma modern seperti algoritma banker dan algoritma WDY yang digunakan untuk mengurangi frekuensi dan dampak dari deadlock dalam sistem operasi Windows.

BIIKMA : Buletin Ilmiah Ilmu Komputer dan Multimedia
Volume 2, No. 1, Juni Tahun 2024
ISSN 3024-8248 (media online)
Hal 60-64

Dengan memperdalam pemahaman tentang deadlock dan menerapkan strategi yang tepat dalam penanganannya, diharapkan artikel ini dapat memberikan panduan yang bermanfaat bagi pengguna sistem komputer, khususnya mereka yang mengandalkan sistem operasi Windows, untuk mengoptimalkan kinerja dan stabilitas sistem mereka.

Sistem operasi adalah salah satu dari empat komponen utama komputer (buku teks sistem operasi), dan sistem operasi adalah penghubung antara pengguna komputer (brainware) dan perangkat keras yang digunakan untuk membuat pengguna (perangkat keras).Pengalaman komputer Anda akan lebih menyenangkan dan sumber daya kinerja sistem komputer akan digunakan dengan lebih efisien (Yunianto & Adhiyarta, 2020).

2. METODE PENELITIAN

Dalam penyajian artikel ini, penulis menggunakan metode penelitian deskriptif kualitatif dengan pendekatan tinjauan pustaka. Penelitian kepustakaan atau penelitian kepustakaan merupakan suatu metode pengumpulan data yang melibatkan kajian buku secara mendalam, penelitian ini berfokus pada aspek pencegahan dan penanganan deadlock, mengidentifikasi faktor-faktor penyebab deadlock serta dampak yang ditimbulkannya pada sistem operasi Windows. Studi ini memberikan gambaran rinci dan komprehensif tentang strategi penanganan kebuntuan yang efektif pada sistem operasi berbasis Windows (Fajar et al., 2022).

3. HASIL DAN PEMBAHASAN

3.1. Pengertian Windows

Windows adalah sistem operasi yang dikembangkan oleh Microsoft, pertama kali diperkenalkan pada 20 November 1985 sebagai antarmuka grafis tambahan untuk MS-DOS. Sejak itu, Windows telah berkembang menjadi sistem operasi yang dominan di pasar komputer pribadi (PC), digunakan di berbagai perangkat termasuk desktop, laptop, tablet, dan server. Windows dikenal dengan antarmuka pengguna grafisnya yang intuitif, ekosistem perangkat lunak yang luas, dan kompatibilitas yang tinggi dengan berbagai jenis perangkat keras.

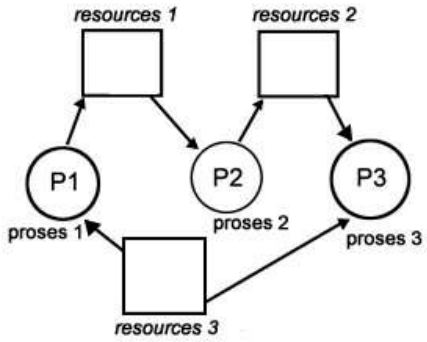
Sistem operasi Windows (atau sering disebut dengan Windows saja) merupakan sistem operasi yang dikembangkan oleh perusahaan besar bernama Microsoft Corporation. Windows hanyalah pilihan sistem operasi. Windows menggunakan antarmuka pengguna grafis (GUI) atau memiliki antarmuka berbasis grafis sebagai sistem operasinya (Muhammad, 2020).

3.2. Pengertian Deadlock

Deadlock adalah suatu kondisi dalam sistem komputasi di mana dua atau lebih proses atau thread terjebak dalam keadaan menunggu satu sama lain untuk melepaskan sumber daya yang mereka butuhkan agar dapat melanjutkan eksekusi. Hal ini menyebabkan proses-proses tersebut tidak dapat bergerak maju, mengakibatkan kebuntuan dalam sistem. Kondisi deadlock adalah salah satu masalah yang signifikan dalam manajemen sumber daya di sistem operasi dan dapat menyebabkan penurunan kinerja atau bahkan kegagalan sistem.

Deadlock juga mengacu pada situasi di mana dua atau lebih proses menunggu proses lain melepaskan sumber daya yang digunakannya. Pekerjaan beberapa proses tidak mengalami kemajuan karena mereka menunggu satu sama lain. Deadlock sebenarnya adalah kebuntuan. Kebuntuan yang disebutkan dalam sistem operasi adalah kebuntuan proses. Deadlock adalah masalah umum yang terjadi ketika banyak proses berbagi sumber daya yang hanya dapat diubah oleh satu proses pada satu waktu. Deadlock merupakan kondisi dimana proses tidak lagi berjalan atau tidak ada komunikasi antar proses (Apriyani et al., n.d.).

BIIKMA : Buletin Ilmiah Ilmu Komputer dan Multimedia
Volume 2, No. 1, Juni Tahun 2024
ISSN 3024-8248 (media online)
Hal 60-64



Gambar 1. Ilustrasi Deadlock

a. Terdapat tiga proses: P1, P2, dan P3. Ada tiga sumber daya: Resource 1, Resource 2, dan Resource 3.

b. Setiap proses memerlukan beberapa sumber daya:

   1. P1 memerlukan Resource 1 dan Resource 3.
   2. P2 memerlukan Resource 2 dan Resource 1.
   3. P3 memerlukan Resource 3 dan Resource 2.

c. Kondisi deadlock terjadi karena:

   1. P1 menunggu Resource 3 yang dipegang oleh P3.
   2. P2 menunggu Resource 1 yang dipegang oleh P1.
   3. P3 menunggu Resource 2 yang dipegang oleh P2.

Akibatnya, tidak ada proses yang dapat melanjutkan, dan sistem terjebak dalam kondisi deadlock.

3.3. Identifikasi Penyebab Terjadinya Deadlock

a. Graf Alokasi Sumber Daya

Diagram alokasi sumber daya adalah model yang digunakan untuk mewakili hubungan antara proses dan sumber daya dalam suatu sistem. Dalam diagram ini, node mewakili proses dan sumber daya, dan tepi mewakili pemetaan sumber daya dari proses ke sumber daya atau dari sumber daya ke proses yang menunggu. Kebuntuan dapat dideteksi dengan memeriksa keberadaan siklus dalam grafik alokasi sumber daya. Ketika ada siklus, kebuntuan dapat terjadi karena proses-proses dalam siklus menunggu sumber daya masing-masing.

b. Jenis Sumber Daya yang Terbatas

Jenis sumber daya yang terbatas juga dapat menyebabkan kebuntuan. Sumber daya yang sering digunakan dan dalam jumlah terbatas, seperti printer, memori, dan perangkat input/output lainnya, lebih rentan terhadap kebuntuan. Ketika beberapa proses bersaing untuk mendapatkan sumber daya yang terbatas, kebuntuan kemungkinan besar akan terjadi, terutama jika permintaan sumber daya dari proses tersebut tidak dikelola dengan baik.

c. Kesalahan dalam Pengelolaan Sumber Daya

Kesalahan pengelolaan sumber daya, seperti kegagalan dalam menerapkan protokol alokasi sumber daya yang tepat, dapat meningkatkan risiko kebuntuan. Misalnya, kegagalan dalam memantau dan mengelola antrian permintaan sumber daya atau menggunakan kebijakan yang tidak

BIIKMA : Buletin Ilmiah Ilmu Komputer dan Multimedia
Volume 2, No. 1, Juni Tahun 2024
ISSN 3024-8248 (media online)
Hal 60-64

memperhitungkan potensi kebuntuan dapat mengakibatkan proses menunggu tanpa batas waktu untuk mendapatkan sumber daya satu sama lain.

d. Kompleksitas Sistem

Deadlock lebih mungkin terjadi ketika kompleksitas sistem tinggi, dengan banyak proses berjalan secara paralel dan bersaing untuk mendapatkan sumber daya yang sama. Semakin kompleks suatu sistem, semakin sulit untuk memprediksi dan mengelola interaksi antara proses dan sumber daya, dan semakin besar kemungkinan terjadinya deadlock.

3.4. Pencegahan Terjadinya Deadlock

Pencegahan kebuntuan adalah serangkaian teknik yang digunakan untuk mencegah sistem operasi memasuki situasi kebuntuan. Kebuntuan terjadi ketika suatu proses menunggu sumber daya bersama, sehingga terjadi kebuntuan. Berikut beberapa cara untuk mencegah kebuntuan:

a. Hilangkan status pending dan wait: Suatu proses harus meminta semua sumber daya yang diperlukan pada awal eksekusi, atau melepaskan semua sumber daya sebelum meminta sumber daya tambahan. Pencegahan saling pengecualian: Mengonversi beberapa sumber daya menjadi sumber daya yang dapat dibagikan, sehingga memungkinkan beberapa proses mengaksesnya secara bersamaan.

b. Pengaturan sumber daya: Anda dapat memaksa sumber daya dari proses yang memuatnya dan mengalokasikannya ke proses yang diperlukan, atau menggunakan rollback untuk menghentikan proses menunggu.

c. Pencegahan menunggu melingkar: Tetapkan urutan linier untuk semua sumber daya dan pastikan pemrosesan sumber daya untuk permintaan mengikuti urutan ini.

d. Algoritma Bunker : Mengevaluasi setiap permintaan berdasarkan jumlah maksimum sumber daya yang diperlukan oleh setiap proses, sehingga menghindari alokasi sumber daya yang dapat menyebabkan kondisi tidak aman. Mencegah kebuntuan memerlukan kombinasi kebijakan dan mekanisme yang cermat untuk memastikan alokasi sumber daya yang efisien dan mencegah kebuntuan dalam sistem.

3.5. Konsekuensi Deadlock

Deadlock adalah situasi di mana dua atau lebih proses atau sumber daya menunggu proses atau sumber daya lain melepaskan sumber daya yang mereka gunakan, dan tidak ada yang dapat melanjutkan. Akibat dari kebuntuan adalah:

a. Kehilangan Produktivitas: Sumber daya yang terjebak dalam kebuntuan tidak lagi tersedia untuk proses lain, sehingga mengurangi produktivitas sistem secara keseluruhan.

b. Waktu tunggu tidak produktif: Proses yang terjebak dalam kebuntuan harus menunggu hingga sumber daya dibebaskan, yang dapat menyebabkan penundaan yang tidak produktif.

c. Hilangnya keandalan sistem: Kebuntuan dapat menyebabkan sistem berhenti merespons atau terhenti, yang dapat mempengaruhi keandalan dan kinerja sistem secara keseluruhan.

d. Kesulitan Identifikasi: Kebuntuan bisa jadi sulit dideteksi dan diselesaikan, terutama dalam sistem kompleks dengan banyak proses dan sumber daya.

e. Hilangnya efisiensi: Penggunaan sumber daya yang tidak efisien dapat mengurangi efisiensi sistem secara keseluruhan. Untuk mengurangi dampak kebuntuan, sistem komputer yang kompleks biasanya menerapkan strategi pencegahan dan penanganan kebuntuan, seperti deteksi kebuntuan, penghindaran, dan pemulihan.

4. KESIMPULAN

Deadlock merupakan suatu kondisi dimana terdapat dua atau lebih proses yang menunggu untuk mendapatkan resource yang mereka butuhkan tanpa melepaskan resource yang mereka miliki sehingga terjadi kebuntuan proses. Kondisi deadlock memiliki empat faktor penyebab yakni graf

BIIKMA : Buletin Ilmiah Ilmu Komputer dan Multimedia
Volume 2, No. 1, Juni Tahun 2024
ISSN 3024-8248 (media online)
Hal 60-64

alokasi sumber daya, jenis sumber daya yang terbatas, kesalahan dalam pengelolaan sumber daya, kompleksitas sistem. Kita dapat melakukan pencegahan terhadap kondisi deadlock dengan cara hilangkan status pending dan wait, pengaturan sumber daya, pencegahan menunggu melingkar, dan algoritma bunker. Apabila deadlock sudah terjadi, kita dapat mengatasinya dengan melakukan preemption, melacak kembali, mematikan proses yang menyebabkan deadlock, dan menghindari deadlock. Deadlock mempunyai dampak yang cukup signifikan, dimana akan terjadi ketidakseimbangan sistem operasi, hilangnya produktivitas, keamanan tidak terjamin, serta sulit memecahkan masalah.

Mengatasi Deadlock pada sistem operasi Windows merupakan aspek penting yang mempengaruhi stabilitas, kinerja, dan keandalan sistem. Deadlock, suatu kondisi di mana dua proses atau lebih menunggu satu sama lain untuk sumber daya yang tidak pernah tersedia, dapat menyebabkan gangguan yang tidak diinginkan dalam eksekusi aplikasi dan menurunkan kinerja sistem secara keseluruhan. Oleh karena itu, penting untuk mengembangkan metode yang efektif untuk mencegah, mengidentifikasi penyebab, dan mengatasi kebuntuan.

REFERENCES

Apriyani, M. E., Hamdana, E. N., & Alhamri, R. Z. (n.d.). *Sistem operasi*.

Fajar, A., Dhika, D. S., & Febriansyah, R. P. (2022). Strategi Penanganan Deadlock Yang Efektif Dalam SistemOperasi Berbasis Windows: Pencegahan Deadlock,Mengidentifikasi Faktor Penyebab, Dan Dampak Dari Deadlock. *SINTESIA: Jurnal Sistem Dan Teknologi Informasi Indonesia*, *2*(1), 6–11.

Muhammad, B. H. P. H. (2020). *Penggunaan Windows Bajakan Di Kalangan Mahasiswa Iain Palangka Raya (Perspektif Hukum Ekonomi Syariah)*. 79–81.

Saifulloh, & Mumtahana, H. A. (2017). Implementasi Penanganan Deadlock Menggunakan Metode Taskkill. *Jurnal TRANSFORMASI*, *13*(2), 115–119.

Yunianto, I., & Adhiyarta, K. (2020). Jurnal Review: Perbandingan Sistem Operasi Linux Dengan Sistem Operasi Windows. *JUPITER : Journal of Computer & Information Technology*, *1*(1), 1–7. https://doi.org/10.53990/jupiter.v1i1.3

Bintara, W. S. (2023). “Pengertian Windows – Definisi, Fungsi, Sejarah, Kelebihan, Kekurangan.” Tersedia di: [https://dianisa.com/pengertian-windows/].

Muhajir, M. “Strategi Mengatasi Deadlock.” Tersedia di: [https://sites.google.com/a/student.unsika.ac.id /karaos/deadlock-pada-sistem-operasi/strategimengatasi-deadlock]. (Akses: 16 Juni 2023).

Universitas Kristen Maranatha. “Sistem Operasi Komputer - Deadlock.” Tersedia di: [https://repository.dinus.ac.id/docs/ajar/7- deadlock.pdf]. (Akses: 15 Juni 2023).