

Madani: Jurnal Ilmiah Multidisiplin

Volume 2, Nomor 8, Agustus 2024, Halaman 95-106

Licensed by CC BY-SA 4.0

E-ISSN: 2986-6340

DOI: <https://doi.org/10.5281/zenodo.13208385>

Pengembangan Inovatif Sistem Operasi Real-Time Untuk Aplikasi Dengan Ketepatan Waktu Sangat Tinggi Menggunakan Sistem Operasi Windows & Linux

Rakhmadi Rahman¹, Muh. Daffa Nashwan Rasya², Arfika Amalia Suryadi³

^{1,2,3}Institut Teknologi Bacharuddin Jusuf Habibie

Email: rakhmadi.rahman@ith.ac.id, muhdaffanashwanrasya27@gmail.com, Arfikaamalia@gmail.com³

Abstract

This study explores the development of an innovative real-time operating system aimed at enhancing time precision for applications with very high timing requirements, utilizing both Windows and Linux operating systems. By investigating and analyzing novel techniques in task management, priority scheduling, and hardware-software interactions, this research seeks to ensure optimal real-time performance. The findings are expected to make significant contributions to the development of critical applications requiring high accuracy and strict timing precision, and provide insights into the implementation of real-time solutions within commonly used operating system platforms.

Keywords: Real Time Operating System, Applications, Windows & Linux Operating System

Abstrak

Penelitian ini mengkaji pengembangan sistem operasi real-time inovatif yang bertujuan untuk meningkatkan ketepatan waktu dalam aplikasi dengan kebutuhan waktu yang sangat tinggi, dengan memanfaatkan sistem operasi Windows dan Linux. Melalui pendekatan ini, penelitian ini mengeksplorasi dan menganalisis teknik-teknik baru dalam pengelolaan tugas, pengaturan prioritas, serta interaksi antara perangkat keras dan perangkat lunak untuk memastikan kinerja waktu nyata yang optimal. Hasil dari studi ini diharapkan dapat memberikan kontribusi signifikan terhadap pengembangan aplikasi kritis yang memerlukan akurasi tinggi dan ketepatan waktu yang sangat ketat, serta memberikan wawasan tentang penerapan solusi real-time dalam platform sistem operasi yang umum digunakan.

Kata Kunci: Sistem Operasi Real Time, Aplikasi, Sistem Operasi Windows & Linux

Article Info

Received date: 15 June 2024

Revised date: 28 June 2024

Accepted date: 15 July 2024

PENDAHULUAN

Dalam dunia teknologi yang dinamis, sistem operasi real-time (RTOS) menjadi semakin penting untuk banyak aplikasi penting, seperti kontrol industri, telekomunikasi, dan sistem transportasi, di mana kegagalan untuk memenuhi batas waktu yang ditentukan dapat menyebabkan konsekuensi serius. Namun, menciptakan sistem operasi real-time (RTOS) yang inovatif pada platform Windows memerlukan penelitian dan inovasi yang mendalam, yang memungkinkan basis yang lebih luas untuk pengembangan aplikasi dan memungkinkan integrasi yang lebih baik dengan infrastruktur teknologi informasi yang ada. Untuk aplikasi seperti kedirgantaraan, kesehatan, dan kendaraan, sistem operasi realtime (RTOS) sangat penting. sistem operasi real-time (RTOS) harus dapat menyesuaikan diri dengan berbagai perangkat keras dan memenuhi persyaratan ketepatan waktu yang ketat dengan meningkatnya permintaan akan aplikasi yang lebih kompleks dan beragam. Pengembangan RTOS pada platform Windows, bagaimanapun, menawarkan kesempatan untuk memanfaatkan ekosistem yang sudah mapan dan memperluas kemampuan Windows ke aplikasi real-time. Namun, masalah seperti manajemen sumber daya, scheduling, dan integrasi dengan berbagai perangkat keras menimbulkan tantangan. (Dharmadasa, M. K., Lee, K. S. W., & McMahan, T. M. G. 2020)

Dalam dunia komputasi modern, kebutuhan akan sistem operasi real-time (RTOS) semakin mendesak, terutama untuk aplikasi yang memerlukan ketepatan waktu sangat tinggi. Sistem operasi real-time dirancang untuk memenuhi kebutuhan waktu yang kritis dengan menjamin bahwa proses atau tugas tertentu dapat diselesaikan dalam jangka waktu yang telah ditentukan. Aplikasi-aplikasi seperti sistem kendali industri, kendaraan otonom, dan sistem komunikasi real-time sangat bergantung

pada kemampuan RTOS untuk memberikan respons yang tepat waktu dan dapat diprediksi. Pengembangan sistem operasi real-time yang inovatif menjadi kunci untuk memenuhi tuntutan aplikasi ini dan meningkatkan kinerja serta keandalan. Sistem operasi Windows dan Linux, dua platform dominan dalam ekosistem komputasi, menawarkan berbagai fitur dan kemampuan yang dapat dimanfaatkan untuk pengembangan RTOS. Windows, dengan berbagai edisi dan dukungannya terhadap berbagai aplikasi komersial, menawarkan kerangka kerja yang kuat untuk pengembangan aplikasi real-time, namun sering kali dikritik karena kurangnya determinisme waktu dalam lingkungan desktopnya. Di sisi lain, Linux, dengan sifat open-source dan dukungan komunitas yang luas, menyediakan berbagai modifikasi dan kernel khusus yang dapat dioptimalkan untuk kebutuhan real-time, seperti PREEMPT-RT dan Xenomai. (Lee, J. J. S., Kim, C. H., & Choi, R. Y. 2020)

Windows RTOS, seperti Windows Embedded Compact dan Windows IoT, telah diperkenalkan untuk memenuhi kebutuhan sistem real-time dengan menawarkan kemampuan untuk penyesuaian dan pengaturan waktu yang lebih baik. Namun, implementasi dan kinerja dari sistem ini sering kali dipengaruhi oleh kompleksitas sistem operasi dan overhead yang tidak diinginkan. Pengembangan lebih lanjut dalam area ini diperlukan untuk mengatasi tantangan-tantangan tersebut dan meningkatkan ketepatan waktu serta konsistensi dalam aplikasi real-time. Sementara itu, Linux menawarkan fleksibilitas melalui berbagai kernel real-time yang dapat disesuaikan untuk memenuhi kebutuhan spesifik dari aplikasi real-time. Kernel PREEMPT-RT, yang dirancang untuk mengurangi latensi dan meningkatkan responsivitas, merupakan salah satu contoh bagaimana Linux dapat dioptimalkan untuk lingkungan real-time. Xenomai, sebagai solusi lain, menawarkan lapisan real-time di atas kernel Linux yang memungkinkan pengembangan aplikasi dengan ketepatan waktu yang lebih tinggi tanpa mengorbankan fungsionalitas dasar Linux.

Pengembangan inovatif dalam sistem operasi real-time untuk aplikasi dengan ketepatan waktu sangat tinggi memerlukan pemahaman mendalam tentang karakteristik dan keterbatasan dari kedua platform ini. Oleh karena itu, pendekatan yang komprehensif harus dilakukan untuk mengevaluasi dan membandingkan kemampuan Windows dan Linux dalam menyediakan solusi RTOS yang efektif. Analisis ini akan mencakup aspek-aspek seperti latensi, determinisme waktu, dan pengelolaan sumber daya, serta bagaimana kedua platform ini dapat disesuaikan untuk aplikasi dengan tuntutan waktu yang sangat ketat. Artikel ini bertujuan untuk mengeksplorasi dan membandingkan pendekatan inovatif dalam pengembangan sistem operasi real-time menggunakan Windows dan Linux. Dengan meneliti teknik-teknik terbaru dan studi kasus aplikasi nyata, diharapkan artikel ini dapat memberikan wawasan berharga mengenai bagaimana kedua platform ini dapat dioptimalkan untuk memenuhi kebutuhan aplikasi real-time dan bagaimana mereka dapat bersaing dalam konteks ketepatan waktu yang sangat tinggi.

HASIL DAN PEMBAHASAN

Definisi Sistem Operasi Real-Time (RTOS)

Sistem Operasi Real-Time (RTOS) adalah jenis sistem operasi yang dirancang untuk menjalankan aplikasi dengan ketepatan waktu yang sangat tinggi. RTOS digunakan dalam sistem yang membutuhkan waktu respons yang tidak hanya cepat tetapi juga deterministik, artinya waktu respons dapat diprediksi. Contoh aplikasi yang memerlukan RTOS termasuk sistem kendali industri, telekomunikasi, sistem transportasi, dan perangkat medis. Misalnya, dalam sistem kendali penerbangan, RTOS digunakan untuk memastikan bahwa instruksi dari kontroler penerbangan diproses dalam waktu yang sangat singkat untuk menjaga keselamatan penerbangan.

RTOS memastikan bahwa tugas-tugas kritis mendapatkan prioritas utama dan dieksekusi dalam waktu yang telah ditentukan. Ini dicapai melalui berbagai teknik penjadwalan tugas, seperti penjadwalan prioritas, di mana tugas dengan prioritas tertinggi dieksekusi terlebih dahulu. Penjadwalan waktu nyata lainnya termasuk time slicing, yang membagi waktu CPU menjadi irisan-irisan kecil sehingga setiap tugas mendapatkan waktu CPU yang cukup. Contoh penggunaan penjadwalan prioritas dapat ditemukan dalam sistem navigasi mobil, di mana informasi dari sensor harus diproses secara real-time untuk mengarahkan kendaraan secara akurat.

Karakteristik RTOS

Karakteristik utama dari Sistem Operasi Real-Time (RTOS) adalah determinisme, yang berarti kemampuan sistem untuk memberikan respons yang dapat diprediksi dalam waktu yang telah ditentukan. Determinisme ini sangat penting dalam aplikasi seperti sistem kendali penerbangan atau

peralatan medis, di mana setiap keterlambatan dalam respons bisa berakibat fatal. Misalnya, dalam kendali robotika industri, RTOS memastikan bahwa perintah dari kontroler diproses dalam waktu yang sangat singkat untuk menghindari kecelakaan atau kerusakan pada produk.

Kemampuan untuk mengelola prioritas tugas juga menjadi karakteristik penting dari RTOS. Dalam sistem RTOS, tugas-tugas diberi prioritas berdasarkan pentingnya tugas tersebut. Penjadwalan berbasis prioritas memastikan bahwa tugas-tugas dengan prioritas lebih tinggi selalu mendapatkan akses ke CPU sebelum tugas-tugas dengan prioritas lebih rendah. Sebagai contoh, dalam sistem keamanan rumah pintar, tugas mendeteksi intrusi (prioritas tinggi) harus diproses lebih dulu daripada tugas mengatur suhu ruangan (prioritas rendah).

Arsitektur RTOS pada Windows

Mengimplementasikan RTOS pada platform Windows memerlukan pemahaman yang mendalam tentang arsitektur Windows. Arsitektur Windows terdiri dari beberapa komponen kunci seperti kernel, subsistem user mode, dan driver perangkat keras. Kernel adalah inti dari sistem operasi yang mengelola sumber daya sistem dan mengontrol akses ke perangkat keras. Untuk menjalankan RTOS pada Windows, integrasi yang baik dengan kernel Windows sangat penting untuk memastikan waktu respons yang cepat dan dapat diprediksi. Salah satu tantangan utama dalam mengembangkan RTOS pada Windows adalah manajemen sumber daya. RTOS harus dapat mengelola CPU, memori, dan perangkat I/O dengan cara yang efisien untuk memastikan bahwa semua tugas kritis dapat dieksekusi tepat waktu. Teknik-teknik seperti penjadwalan prioritas, interrupt handling, dan task synchronization perlu dioptimalkan untuk platform Windows. Misalnya, pada aplikasi pengendalian lalu lintas udara, RTOS pada Windows harus mampu mengelola ribuan sinyal radar secara real-time.

Perbandingan RTOS dengan Sistem Operasi Konvensional

1. RTOS berbeda dari sistem operasi konvensional dalam beberapa aspek penting.

RTOS dirancang untuk memastikan respons dalam jangka waktu yang sangat singkat dan dapat diprediksi, yang dikenal sebagai determinisme. Sistem operasi konvensional seperti Windows atau Linux tidak memberikan jaminan ini karena mereka lebih fokus pada throughput dan penggunaan sumber daya yang efisien daripada waktu respons yang deterministik. Misalnya, dalam desktop computing, waktu respons yang tepat tidak selalu menjadi prioritas utama. RTOS memiliki mekanisme penjadwalan yang berbeda dari sistem operasi konvensional. Penjadwalan dalam RTOS biasanya berbasis prioritas dan dirancang untuk memastikan bahwa tugas-tugas dengan prioritas tinggi selalu mendapatkan akses ke CPU sebelum tugas-tugas dengan prioritas lebih rendah. Sistem operasi konvensional, di sisi lain, menggunakan penjadwalan berbasis waktu atau berbasis fairness untuk memastikan bahwa semua tugas mendapatkan waktu CPU yang adil, tanpa memperhatikan prioritas waktu nyata. Contohnya, dalam aplikasi pemutaran video di sistem operasi konvensional, tujuan utamanya adalah untuk memastikan kelancaran pemutaran video, bukan ketepatan waktu yang sangat tinggi.

2. Pengembangan Rtos Pada Windows

a Tahap-Tahap Pengembangan

Proses pengembangan RTOS pada platform Windows melibatkan beberapa tahap utama. Tahap pertama adalah analisis kebutuhan, di mana para pengembang mengidentifikasi kebutuhan spesifik dari aplikasi yang akan dijalankan pada RTOS. Ini mencakup identifikasi tugas-tugas kritis yang memerlukan respons waktu nyata, penentuan batas waktu yang ketat, dan kebutuhan sumber daya perangkat keras dan perangkat lunak. Misalnya, dalam pengembangan sistem kendali robotika, pengembang harus memastikan bahwa sistem dapat merespons perintah kontrol dalam beberapa milidetik.

Tahap kedua adalah desain sistem, di mana arsitektur RTOS dirancang untuk memenuhi kebutuhan yang telah diidentifikasi. Ini melibatkan pemilihan algoritma penjadwalan yang tepat, perancangan mekanisme manajemen memori yang efisien, dan penentuan strategi interrupt handling yang efektif. Desain sistem harus mempertimbangkan integrasi dengan kernel Windows serta kompatibilitas dengan berbagai driver perangkat keras yang digunakan. Misalnya, dalam desain sistem kendali lalu lintas udara, arsitektur RTOS harus mampu menangani ribuan sinyal radar secara simultan.

Tantangan dan Solusi

Pengembangan RTOS pada Windows menghadapi berbagai tantangan yang signifikan. Salah satu tantangan utama adalah manajemen sumber daya yang efisien. RTOS harus dapat mengelola CPU, memori, dan perangkat I/O dengan cara yang efisien untuk memastikan bahwa semua tugas kritis dapat dieksekusi tepat waktu. Ini sangat penting dalam aplikasi seperti kendali industri di mana penundaan sekecil apa pun dapat menyebabkan kegagalan sistem atau kecelakaan. Contohnya, dalam sistem kendali robotik, RTOS harus memastikan bahwa setiap perintah kontrol dijalankan dalam waktu yang ditentukan untuk menghindari kesalahan dalam operasi robot. Tantangan lain adalah integrasi dengan driver perangkat keras yang berbeda. RTOS pada Windows harus dapat berfungsi dengan berbagai perangkat keras yang digunakan oleh sistem. Ini mencakup driver untuk perangkat I/O, jaringan, dan perangkat penyimpanan. Integrasi yang baik dengan driver perangkat keras sangat penting untuk memastikan bahwa RTOS dapat berfungsi dengan benar pada berbagai konfigurasi perangkat keras. Misalnya, dalam sistem kendali kendaraan otonom, RTOS harus berfungsi dengan berbagai sensor dan aktuator untuk memastikan operasi yang aman dan efisien.

Studi Kasus: Implementasi RTOS pada Windows

Implementasi RTOS pada Windows dapat dianalisis melalui studi kasus yang konkret. Salah satu contohnya adalah sistem kendali proses industri yang menggunakan RTOS untuk mengelola berbagai perangkat sensor dan aktuator. Dalam studi kasus ini, pengembang harus memastikan bahwa RTOS dapat berfungsi dengan baik di lingkungan Windows dan dapat mengelola tugas-tugas kritis dengan waktu respons yang sangat cepat. Proses ini melibatkan integrasi RTOS dengan kernel Windows dan pengujian sistem di bawah kondisi operasi yang berbeda.

Studi kasus ini juga menunjukkan pentingnya penjadwalan prioritas dalam RTOS. Dalam sistem kendali proses industri, tugas-tugas tertentu seperti pemantauan suhu dan tekanan harus diberi prioritas tinggi karena kegagalan untuk merespons perubahan kondisi ini dapat menyebabkan kecelakaan atau kerusakan pada peralatan. RTOS menggunakan algoritma penjadwalan prioritas untuk memastikan bahwa tugas-tugas ini selalu mendapatkan akses ke CPU sebelum tugas-tugas lain yang kurang kritis. Contoh implementasi ini dapat dilihat pada pabrik kimia di mana RTOS memastikan bahwa semua parameter proses diawasi secara real-time.

Analisis Kinerja RTOS

Analisis kinerja RTOS melibatkan evaluasi berbagai aspek kinerja sistem untuk memastikan bahwa RTOS memenuhi semua persyaratan yang telah ditetapkan. Salah satu aspek kinerja utama yang dianalisis adalah waktu respons. RTOS harus mampu merespons kejadian eksternal dalam waktu yang sangat singkat dan dapat diprediksi. Analisis ini melibatkan pengukuran waktu respons untuk berbagai tugas kritis di bawah kondisi operasi yang berbeda. Misalnya, dalam sistem kendali lalu lintas udara, waktu respons untuk memproses sinyal radar harus sangat singkat untuk memastikan keselamatan penerbangan. Efisiensi penggunaan sumber daya juga merupakan aspek penting dari analisis kinerja RTOS. RTOS harus dapat mengelola CPU, memori, dan perangkat I/O dengan cara yang efisien untuk memastikan bahwa semua tugas kritis dapat dieksekusi tepat waktu. Analisis ini melibatkan evaluasi penggunaan sumber daya di bawah berbagai kondisi beban kerja untuk memastikan bahwa RTOS dapat berfungsi dengan efisien bahkan dalam kondisi operasi yang berat. Misalnya, dalam sistem telekomunikasi, efisiensi penggunaan sumber daya sangat penting untuk mengelola lalu lintas data yang tinggi tanpa mengalami penurunan kinerja.

Pengembangan Sistem Operasi Windows Dan Linux

1. Windows

Sistem operasi real-time memegang peranan penting dalam berbagai bidang, terutama dalam aplikasi yang membutuhkan respons cepat dan tepat waktu. Sistem operasi ini dirancang untuk memproses tugas-tugas kritis secara prioritas, memastikan bahwa tindakan yang diperlukan dapat diambil dalam batas waktu yang ditentukan.

a. Perbandingan sistem operasi Windows dan Linux untuk real-time

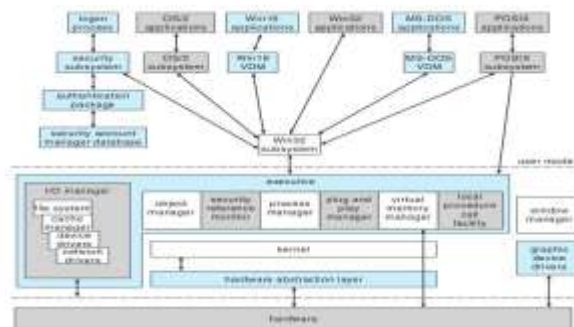
Ketika mempertimbangkan sistem operasi untuk aplikasi real-time, Windows dan Linux masing-masing memiliki keunggulan dan kelemahan yang perlu dipertimbangkan. Windows, dengan arsitekturnya yang tertanam dan dukungan driver yang luas, menawarkan kemudahan penggunaan dan integrasi yang baik dengan perangkat keras. Namun, Windows cenderung memiliki latensi yang lebih tinggi dan kurang fleksibel dalam penyesuaian parameter real-time. Di sisi lain, Linux menawarkan kontrol yang lebih baik atas karakteristik real-time melalui kernel realtime dan

kemampuan kustomisasi yang tinggi. Meskipun memerlukan konfigurasi yang lebih kompleks, Linux mampu mencapai latensi yang lebih rendah dan responsivitas yang lebih baik untuk aplikasi real-time yang kritis. Ketika mempertimbangkan sistem operasi untuk aplikasi real-time, Windows dan Linux masing-masing memiliki keunggulan dan kelemahan yang perlu dipertimbangkan. Windows, dengan arsitekturnya yang tertanam dan dukungan driver yang luas, menawarkan kemudahan penggunaan dan integrasi yang baik dengan perangkat keras. Namun, Windows cenderung memiliki latensi yang lebih tinggi dan kurang fleksibel dalam penyesuaian parameter real-time. Di sisi lain, Linux menawarkan kontrol yang lebih baik atas karakteristik real-time melalui kernel real-time dan kemampuan kustomisasi yang tinggi. Meskipun memerlukan konfigurasi yang lebih kompleks, Linux mampu mencapai latensi yang lebih rendah dan responsivitas yang lebih baik untuk aplikasi real-time yang kritis.

b. Tantangan dalam Pengembangan Sistem Operasi Real-Time

Pengembangan sistem operasi real-time menghadapi beberapa tantangan yang harus dipertimbangkan dengan cermat. Salah satu tantangan utama adalah memastikan ketepatan waktu dan keandalan sistem, yang menjadi inti dari sistem operasi real-time. Hal ini berarti memastikan bahwa tugas-tugas kritis dapat diselesaikan dalam batas waktu yang ditentukan, bahkan dalam kondisi beban kerja yang tinggi. Selain itu, manajemen memori yang efisien juga menjadi tantangan penting. Sistem operasi real-time harus mampu mengalokasikan dan mengelola memori secara tepat agar dapat memenuhi kebutuhan aplikasi real-time yang sensitif terhadap waktu. Penggunaan memori yang berlebihan atau alokasi memori yang tidak efisien dapat menyebabkan penundaan atau kegagalan pada sistem. Tantangan lainnya adalah sinkronisasi dan koordinasi antar proses yang berjalan pada sistem operasi real-time. Proses-proses tersebut harus dapat berinteraksi dan berkoordinasi dengan baik untuk memastikan integritas data dan konsistensi sistem. Mekanisme sinkronisasi yang tepat sangat diperlukan untuk menghindari deadlock atau kondisi race yang dapat mengganggu kinerja sistem.

c. Arsitektur Sistem Operasi Real-Time Berbasis Windows



Pengembangan sistem operasi real-time menghadapi beberapa tantangan yang harus dipertimbangkan dengan cermat. Salah satu tantangan utama adalah memastikan ketepatan waktu dan keandalan sistem, yang menjadi inti dari sistem operasi real-time. Hal ini berarti memastikan bahwa tugas-tugas kritis dapat diselesaikan dalam batas waktu yang ditentukan, bahkan dalam kondisi beban kerja yang tinggi. Selain itu, manajemen memori yang efisien juga menjadi tantangan penting. Sistem operasi real-time harus mampu mengalokasikan dan mengelola memori secara tepat agar dapat memenuhi kebutuhan aplikasi real-time yang sensitif terhadap waktu. Penggunaan memori yang berlebihan atau alokasi memori yang tidak efisien dapat menyebabkan penundaan atau kegagalan pada sistem.

Tantangan lainnya adalah sinkronisasi dan koordinasi antar proses yang berjalan pada sistem operasi real-time. Proses-proses tersebut harus dapat berinteraksi dan berkoordinasi dengan baik untuk memastikan integritas data dan konsistensi sistem. Mekanisme sinkronisasi yang tepat sangat diperlukan untuk menghindari deadlock atau kondisi race yang dapat mengganggu kinerja sistem. Arsitektur sistem operasi real-time berbasis Windows memiliki beberapa komponen utama yang memungkinkannya untuk mengelola waktu secara efisien dan menjamin respons yang tepat waktu. Pada lapisan dasar, terdapat kernel Windows yang telah dimodifikasi untuk mendukung skenario real-time. Kernel ini dilengkapi dengan penjadwal preemptive yang dapat memprioritaskan tugas-tugas kritis dan memastikan mereka dieksekusi sesuai dengan batas waktu yang ditentukan.

Di atas kernel, terdapat lapis aplikasi real-time yang berjalan dalam mode kernel, memberikan akses langsung ke sumber daya hardware dan memungkinkan kontrol yang presisi atas waktu eksekusi. Lapisan ini biasanya terdiri dari driver, middleware, dan library yang dirancang khusus untuk aplikasi real-time. Teknologi seperti Windows Embedded dan Azure RTOS menjadi contoh implementasi sistem operasi real-time berbasis Windows. Untuk memastikan deterministik dan tepat waktu, arsitektur ini juga melibatkan manajemen interupsi yang ketat, optimalisasi memori, dan integrasi dengan perangkat keras khusus seperti prosesor dengan kemampuan real-time yang lebih baik. Sistem operasi Windows real-time ini dapat diaplikasikan dalam berbagai bidang yang membutuhkan respons cepat dan terprediksi, seperti otomasi industri, kontrol proses, dan sistem tertanam. (Smith, A. S., & Zhang, H. W. 2018).

d. Implementasi Sistem Operasi Real-Time pada Windows

Dalam mengimplementasikan sistem operasi real-time pada platform Windows, pendekatan yang umum digunakan adalah dengan memanfaatkan fitur-fitur khusus yang tersedia di dalam sistem operasi Windows itu sendiri. Salah satu contoh implementasi yang sering digunakan adalah dengan menggunakan Windows RealTime Extensions (RTX), sebuah modul tambahan yang memberikan fungsionalitas real-time pada sistem operasi Windows. Windows RTX menyediakan API dan antarmuka pemrograman yang memungkinkan aplikasi real-time untuk berjalan secara efisien pada lingkungan Windows. Modul ini mengizinkan aplikasi untuk memperoleh respon real-time yang cepat dengan meminimalkan intervensi dari sistem operasi Windows yang bersifat nonreal-time. Selain itu, Windows RTX juga menyediakan fitur-fitur seperti penjadwalan prioritas, manajemen waktu yang presisi, dan kontrol atas alokasi sumber daya sistem.

e. Perbandingan Kinerja dan Fitur Sistem Operasi Real-Time

Dalam membandingkan kinerja dan fitur sistem operasi real-time berbasis Windows dan Linux, terdapat beberapa aspek penting yang perlu diperhatikan. Salah satu faktor utama adalah determinisme waktu, yaitu kemampuan sistem untuk memberikan respons yang konsisten dan terprediksi pada waktu-waktu tertentu. Dalam hal ini, sistem operasi real-time Linux umumnya unggul dibandingkan dengan Windows karena memiliki kernel yang lebih sederhana dan terfokus pada tugas-tugas real-time. Dari segi keandalan, Linux juga lebih stabil dan dapat diandalkan untuk aplikasi-aplikasi kritis yang membutuhkan uptime tinggi. Selain itu, Linux memiliki dukungan komunitas open-source yang luas, memungkinkan pengembangan dan perbaikan berkelanjutan. Di sisi lain, Windows menawarkan integrasi yang lebih baik dengan perangkat keras dan perangkat lunak Microsoft, serta antarmuka pengguna yang lebih familiar bagi banyak pengguna.

1. Determinisme waktu: Linux lebih unggul dalam memberikan respons yang konsisten dan terprediksi.
2. Keandalan: Linux lebih stabil dan dapat diandalkan untuk aplikasi-aplikasi kritis
3. Dukungan komunitas: Linux memiliki komunitas open-source yang luas untuk pengembangan dan perbaikan.
4. Integrasi perangkat: Windows lebih unggul dalam integrasi dengan ekosistem Microsoft
5. Antarmuka pengguna: Windows menawarkan antarmuka yang lebih familiar bagi banyak pengguna

f. Contoh Aplikasi yang Menggunakan Sistem Operasi RealTime

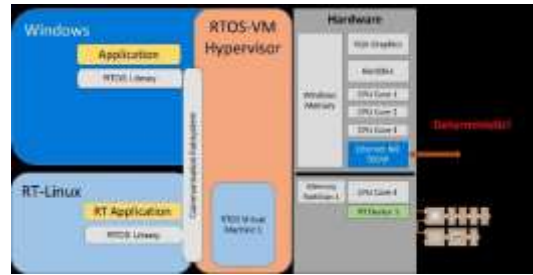
Sistem operasi real-time menemukan banyak aplikasi dalam berbagai industri yang membutuhkan pemrosesan data dan respons yang cepat dan konsisten. Salah satu contoh aplikasi yang sering menggunakan sistem operasi real-time adalah robotika industri. Robot-robot industri yang digunakan di pabrik-pabrik membutuhkan sistem operasi real-time untuk mengkoordinasikan gerakan dan sensor mereka dengan kecepatan dan akurasi yang tinggi. Ini memungkinkan robot untuk merespons dengan cepat terhadap perubahan di lingkungan sekitar mereka dan melakukan tugas-tugas yang memerlukan presisi tinggi, seperti perakitan komponen elektronik. Selain robotika, sistem operasi real-time juga banyak digunakan dalam peralatan medis, seperti alat pacu jantung dan ventilator. Dalam aplikasi medis, ketepatan waktu operasi sangat kritis untuk menyelamatkan nyawa pasien. Sistem operasi real-time memastikan bahwa perangkat medis dapat merespons dengan cepat terhadap perubahan kondisi pasien dan memberikan perawatan yang tepat waktu.

Linux

Arsitektur Sistem Operasi Linux Sistem operasi Linux memiliki konsep yang hampir sama dalam hal arsitektur dengan sistem operasi lainnya. Perbedaannya adalah bentuk dari disetiap

komponen pada arsitektur tersebut, antara lain seperti berikut ini.

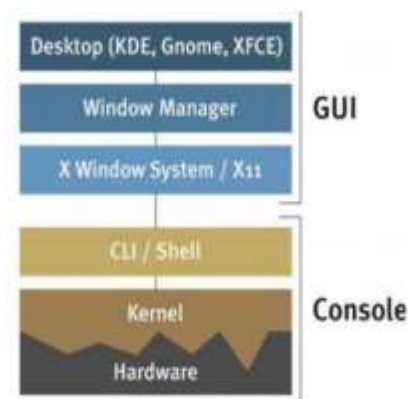
Kernel



Kernel adalah suatu perangkat lunak yang menjadi bagian utama dari sebuah sistem operasi. Tugasnya melayani bermacam program aplikasi untuk mengakses perangkat keras komputer secara aman. Akses kepada perangkat keras secara langsung merupakan masalah yang kompleks, oleh karena itu kernel biasanya mengimplementasikan sekumpulan abstraksi hardware. Abstraksi tersebut dapat memungkinkan akses kepada perangkat keras s kepada perangkat keras menjadi mudah dan seragam. Sehingga abstraksi pada akhirnya memudahkan pekerjaan programmer. Sistem operasi mengembangkan kernel sistem operasi yang pada akhirnya terbagi menjadi empat bagian yang secara desain berbeda, sebagai berikut:

1. Kernel Monolitik : Kernel monolitik mengintegrasikan banyak fungsi di dalam kernel dan menyediakan lapisan abstraksi perangkat keras secara penuh terhadap perangkat keras yang berada di bawah sistem operasi.
2. Mikrokernel : Mikrokernel menyediakan sedikit saja dari abstraksi perangkat keras dan menggunakan aplikasi yang berjalan di atasnya—yang disebut dengan server—untuk melakukan beberapa fungsionalitas lainnya.
3. Kernel Hibrida : Kernel hibrida adalah pendekatan desain mikrokernel yang dimodifikasi. Pada hybrid kernel, terdapat beberapa tambahan kode di dalam ruangan kernel untuk meningkatkan performanya.
4. Exokernel : Exokernel menyediakan hardware abstraction secara minimal, sehingga program dapat mengakses hardware secara langsung. Dalam pendekatan desain exokernel, library yang dimiliki oleh sistem operasi.

Kernel Linux adalah kernel yang digunakan dalam sistem operasi GNU/Linux. Kernel ini merupakan turunan dari keluarga sistem operasi UNIX, dirilis dengan menggunakan lisensi GNU General Public License (GPL), dan dikembangkan oleh pemrogram di seluruh dunia. Linux merupakan contoh utama dari perangkat lunak bebas dan sumber terbuka (open source). Dibawah ini ditunjukkan gambar keterkaitan antara hardware, kernel, Shell dan Desktop.



Keterkaitan antara hardware, kernel, shell dan Desktop pada Linux

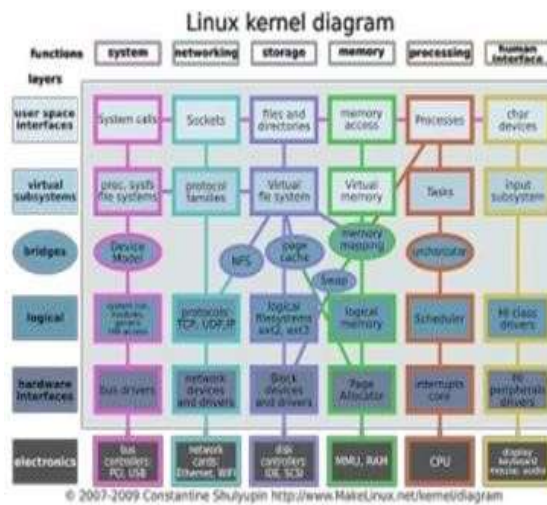
Inti Linux dirilis di bawah GNU General Public License versi 2 (GPLv2), Ini dikembangkan oleh kontributor dari seluruh dunia. Yang pertama adalah kernel Linux Siswa Ilmu Komputer berpikir Pengembang dan pengguna Linux Itu ada di inti Linux Banyak distro Linux yang telah dirilis Berdasarkan inti Linux. Linux kernel telah menerima kontribusi dari ribuan programmer. Banyak. Distribusi Linux telah dirilis berdasarkan Kernel Linux. Contoh gambar linux kernel ditunjukkan

seperti berikut ini.

```

0.9536931 serial: 10042: E88 part at 0x00,0x04 irq 1
0.9548341 serial: 10042: ADX part at 0x00,0x04 irq 12
0.9556653 mousedev: PS/2 mouse device common for all mice
0.9577431 input: AT Translated Set 2 keyboard as /devices/platform/i8042/seri
l/input/lmport
0.9601441 rtc_cmos rtc_cmos: rtc core: registered rtc_cmos as rtc0
0.9612281 rtc0: alarms up to max day, 315 bytes maxam
0.9623071 cpuidle: using governor ladder
0.9633201 cpuidle: using governor menu
0.9643661 TCP cubic registered
0.9653161 NET: Registered protocol family 10
0.9672711 IPv6: IPv6
0.9709381 NET: Registered protocol family 17
0.9722091 Registering the dns_resolver key type
0.9733341 Using IPI No-Sharing mode
0.9745571 registered taskstats version 1
0.9766371 rtc_cmos rtc_cmos: setting system clock to 2011-09-09 20:32:52.07
(1315640077)
0.9820051 Initializing network drop monitor service
0.9830511 Freeing unused kernel memory: 405k freed
0.9846271 Write protecting the kernel text: 2760k
0.9862251 Write protecting the kernel read-only data: 1068k
0.9869351 NX-protecting the kernel data: 3376k
Loading, please wait...
    
```

Linux kernel booting
Linux kernel diagram



Pada gambar diatas ditunjukkan keterkaitan antara fungsi dengan layer yang digunakan. Berdasarkan fungsinya, maka terdapat system, networking, storage, memory, processing dan human interface. Apabila dilihat dari lapisannya, maka terdiri dari layer / lapisan electronics, hardware interface, logical, bridges, virtual subsystems dan user spaces interfaces.

Apa yang di lakukan Kernel ? Fungsinya seperti layaknya sistem operasi, menangani task switching dalam multitasking, menangani permintaan membaca atau menulis peralatan disk, melakukan tugas-tugas network serta mengatur penggunaan memori. Kernel membuat layanan yang ada didalamnya, tersedia untuk Software yang sedang berjalan, melalui sekumpulan besar entry point yang secara teknis, Kernel menggunakan system call read and writer untk memberikan abstraksi Hardware pada Komputer anda. Berikut ini contoh kernel saat melakukan proses booting :

```

root@kali:~# systemctl status
● Started File System Check on /dev/disk/by-uuid/0101...
● Mounted /boot
● Reached target Local File System.
● Starting Recreate Volatile Files and Directories...
● Starting Trigger Flushing of Journal to Persistent Storage...
● Started Recreate Volatile Files and Directories.
● Started Trigger Flushing of Journal to Persistent Storage.
● Reached target Sound Card.
● Started Setup keyboard input.
● Reached target System Initialization.
● Listening on Bus System Message Bus Socket.
● Reached target Sockets.
● Starting Network Sound Card State...
● Reached target Basic System.
● Starting Periodic Command Scheduling...
● Started Periodic Command Scheduling.
● Starting System Manager.
● Starting System Logger Service...
● Started System Logger Service.
● Starting Logic Service...
● Started Logic Service.
● Starting Bus System Message Bus...
● Started Bus System Message Bus.
● Starting Permit User Sessions...
● Started Permit User Sessions.
● Starting Network Sound Card State.
● Started Network Sound Card State.
● Starting Permit User Sessions...
● Started Permit User Sessions.
● Started Logic Service.
    
```

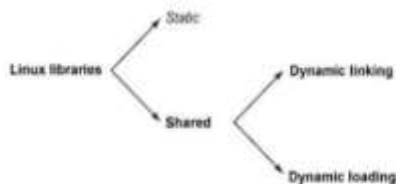
Booting

1. GNU/Linux adalah salah satu system operasi open source seperti UNIX. Struktrudirektori
2. GNU/Linux berbeda dengan system operasi Windows yang selama ini yang kita kenal.

3. GNU/Linux menggunakan pendekatan tree atau pohon, semua direktori pembentuk GNU/Linux terletak dibawah "/" atau root.

Secara umum, sistem operasi linux memiliki direktori di atas, namun pada beberapa distro ditambahkan beberapa direktori spesifik seperti /srv pada keluarga Suse dimana direktori yang ditambahkan memiliki fungsi khusus yang kadangkala tidak terdapat pada distro lain. Seperti gambar di atas direktori user berada di bawah /home. Di gambar terdapat tiga /user1 /user2 /user3 berada dalam satu direktori yang sama (/home) namun tiap user memiliki sebuah direktori terpisah sesuai dengan nama user dan tiap user tidak bisa melihat isi dari direktori user lainnya.

Library



Sebelum membahas Command Line Interface (CLI), sebenarnya di atas kernel masih ada komponen yang tidak diperlihatkan di gambar, yang disebut dengan Library. Library adalah file atau sekumpulan file yang di dalamnya terdapat fungsi atau kelas yang dibutuhkan oleh program aplikasi. Sebagai contoh library adalah Glibc, libcurl, libpng serta library yang lain yang fungsinya menjadi pondasi untuk program yang akan berjalan di atasnya. Misalnya tanpa library getty, maka kita tidak dapat membuka shell atau CLI, tanpa libpng maka kita tidak bisa membuka gambar, tanpa libmpg maka kita tidak bisa memutar mp3. Dengan demikian library adalah pondasi untuk aplikasi atau program tertentu. Kadangkala library juga saling berkaitan (dependensi) misalkan library A adalah pondasi dari Library B dan Program C butuh library B. Hal semacam ini disebut dengan dependency di linux. (Carter, E. K., & Fong, T. R. 2018).

Shell

Shell adalah aplikasi yang bertugas menerima Input perintah dari user. Pada sistem operasi Windows, dapat disamakan dengan Command Prompt. Bedanya, Shell di Linux tidak terikat dengan GUI nya. Linux mempunyai banyak shell, diantaranya adalah csh, tcsh, sh, ash dan yang paling populer Bash. Kebanyakan server, tiga lapis ini saja sudah lebih dari cukup, tinggal ditambahkan service (aplikasi yang berjalan dibelakang layar) misal webserver apache.

X Windows

X Windows adalah Library khusus untuk tampilan GUI di UNIX/Linux, tanpa adanya librari ini, maka KDE, GNome atau openoffice tidak akan dapat berjalan.

Window Manager

Window Manager adalah pengelola jendela dari aplikasi yang berjalan di GUI. Window Manager hanya bertugas menangani posisi aplikasi di Layar, Window manager tidak mempunyai fitur misal Drag and Drop atau fitur canggih lainnya. Contoh window Manager adalah iceWM dan GNUStep.

Perkembangan Linux

Linux merupakan system operasi bertipe Unix. Unix sendiri merupakan system operasi komputer yang dikembangkan oleh AT&T Bell Labs pada tahun 1960 dan 1970-an. Linux merupakan turunan dari Unix dengan system berbasis GNU GPL (General Public License). Unix diciptakan sebagai system operasi yang bisa beradaptasi dengan banyak mesin (Portable), serta memiliki kemampuan Multitasking dan Multi-user. Nama "Linux" berasal dari nama kernel-nya yaitu "Linux kernel" yang dibuat pada tahun 1991 oleh Linus Torvalds. Pada tahun 1996 Larry Ewing membuat gambar penguin yang kemudian menjadi logo Linux dengan nama Tux(Tux = Torvalds + Unix). (Lewis, N. V., & Scott, M. A. 2018).

Linus Torvalds awalnya terinspirasi oleh Minix. minix sendiri Torvalds adalah mahasiswa di Universitas Helsinki, Finlandia. Dia mempunyai ide untuk membuat Minix yang gratis dan terjangkau. Minix adalah proyek kelas serupa di kelasnya saat itu. Sistem UNIX, yang pada akhirnya merupakan hasil penelitiannya, dikenal sebagai kernel Linux. Versi pertama Linux (0,01) dikembangkan sekitar Agustus 1991. Dia memposting hasilnya ke milisnya com.os.minix dengan tujuan untuk menyediakannya. Sejarah OS Linux berkaitan Linus Torvalds, Sang penemu Linux erat dengan proyek GNU, suatu proyek program freeware yang dikepalai oleh Richard M. Stallman.

Proyek ini diawali sekitar tahun 1983 untuk membuat system operasi seperti UNIX yang lengkap beserta compiler, utility aplikasi, utility pembuat yang sepenuhnya dengan perangkat lunak bebas.

Proyek GNU menghasilkan hampir seluruh komponen sistem ini. kecuali intinya. Linus Torvalds dan pengembang kernel mirip Linux sedang mencoba Mengadaptasi kernel mereka agar berfungsi dengan komponen GNU membuat sistem operasi yang berfungsi. Linux saat ini biasanya terlihat seperti ini: Mengelompokkan GNU atau program lain menjadi satu Istilah Distro Linux dapat digunakan dalam jaringan, pengembangan. Sekarang Ada ribuan distribusi Linux di luar sana, masing-masing dengan kelebihan dan kekurangannya. diinstal atau disertakan dalam suatu sistem operasi, tetapi karena perbedaan tersebut Sesuatu yang ada, dan akan selalu ada, kernel Linux yang dibuat oleh Linus. Setelah itu, ada ribuan proyek sukarelawan di seluruh dunia. Sebagai system operasi yang merupakan turunan Unix. Maka Linux mewarisi karakteristik yang dimiliki oleh Unix. Berikut ini beberapa karakteristik milik Unix yang diwarisi oleh Linux.

1. Ready for Network

Pada awalnya Unix memang dikembangkan untuk system komunikasi antar komputer dalam jaringan. Dengan demikian bisa dikatakan bahwa Linux sebagai turunan Unix memang lebih unggul dalam bidang jaringan komputer.

2. Multitasking

Multitasking merupakan suatu mekanisme dimana suatu system operasi memiliki kemampuan untuk menjalankan lebih dari satu proses yang berbeda dalam waktu yang bersamaan tanpa saling mempengaruhi. Misalnya seorang user menjalankan aplikasi pengolah kata dan aplikasi pengolah angka dalam waktu bersamaan. Kedua proses dapat saling berjalan tanpa saling terganggu.

3. Multiuser

Multiuser merupakan suatu mekanisme dimana suatu system operasi bisa digunakan oleh lebih dari satu user yang Login kedalam system pada waktu yang bersamaan.

4. Aman

Sistem operasi Linux menawarkan tingkat keamanan yang lebih baik dibandingkan dengan sistem operasi lain. Dengan izin, setiap pengguna mempunyai hak yang berbeda-beda dengan orang lain.

5. Line / Text Command Line

Metode Line ini mirip dengan DOS pada Windows, tapi bukan berarti system operasi Linux tidak memiliki tampilan GUI (Graphical User Interface). Menurut orang-orang yang sudah biasa menggunakan Linux, kemampuan sebenarnya Linux bukan pada GUI-nya tapi pada Command Line-nya. Linux biasanya telah dipaket dengan program-program dari GNU ataupun program perangkat lunak lainnya dan dibundel menjadi satu dengan istilah distro Linux.

Keunggulan dan Kelemahan Linux

Selain mewarisi karakteristik Unix, Linux sebagai system operasi juga memiliki beberapa keunggulan jika dibandingkan dengan system operasi lain. Bahkan Linux bisa dikatakan sebagai system operasi terbaik yang pernah ada. Berikut kelebihanannya :

1. **Free dan Open Source.** Lisensi kernel menjadikan Linux sebagai sistem operasi. Ini adalah sistem operasi yang gratis dan dapat dimodifikasi secara bebas. Meskipun ada banyak versi distribusi Linux, Meskipun tampilannya komersial, Linux adalah sebuah sistem operasi. Ini gratis dan sumber terbuka.
2. **Fasilitas aplikasi yang lengkap.** Linux dilengkapi dengan puluhan aplikasi multimedia dan aplikasi pengembangan software seperti C, C++, java, Phyton, Ruby dan lain-lain dalam paket CD/DVD instalasinya. Hanya dengan sekali install bisa mendapatkan semua aplikasi tersebut secara free dan open source. Bisa dibayangkan, jika menggunakan system operasi yang mengharuskan seseorang membeli compiler C/C++ agar bisa digunakan pada system, maka harus mengeluarkan biaya sampai ribuan dollar amerika.
3. **Stabil dan tangguh.** Linux adalah system operasi yang luar biasa stabil. Jika melakukan konfigurasi yang benar pada sistemnya, Linux akan “hidup” terus sampai hardware mengalami shutdown pada system. Linux merupakan system yang tangguh, karena mampu bekerja selama ratusan hari tanpa ada kendala jika dikonfigurasi dengan benar. **Memory Linux**

Memahami informasi memori Linux

Setiap sistem operasi membutuhkan memori untuk menyimpan segmen kode program dan data. Ini juga berlaku untuk sistem Linux. Masalahnya: ada banyak informasi yang tersedia mengenai penggunaan memori dan perilakunya. Ketika kita berbicara tentang memori RAM, yang kita maksud

adalah Random Access Memory Ini adalah memori yang dapat ditampilkan dan ditampilkan Anda dapat menemukan jenis memori ini di komputer Anda Memori juga menyimpan informasi. Contoh yang bagus adalah saat Anda berlari Sekali lagi, (Clark, M. O., & Jenkins, P. B. 2018). kita melihat bahwa program berubah dan menggunakan memori karena Ini adalah masalah pada program yang tidak memiliki cukup memori. Ini adalah ukuran yang baik untuk sistem yang menggunakan NFS Periksa berapa banyak data yang tidak tersedia di ruang penyimpanan. Pada sistem tanpa NFS, nilai ini kecil atau nol. Jika Anda membandingkan bagian ini dengan rencana Anda, Kita perlu menemukan area terang untuk fokus. Bandingkan proses serupa dan lihat perbedaan di /proc/ Ini dapat memberikan informasi yang baik tentang di mana dan memori apa yang digunakan. Ini mungkin memberi indikasi bagus dimana memori digunakan dan apa yang membuat sistem sibuk. (Miller, K. R., & Johnson, B. J. 2019)

Proses Di Linux

Pada artikel ini, pemahaman dasar tentang proses dan secara singkat melihat bagaimana mengelola proses di Linux dengan menggunakan perintah tertentu. Sebuah proses mengacu pada sebuah program dalam pelaksanaan; Ini adalah contoh program yang sedang berjalan. Ini terdiri dari instruksi program, data dibaca dari file, program lain atau masukan dari pengguna sistem. (Chen, L. T., & Wang, Y. S. 2019).



Proses state pada sistem operasi Linux dapat dijelaskan sebagai berikut:

1. Created (Dibuat): Ketika proses baru dibuat, ia berada dalam state ini. Proses ini belum dimasukkan ke antrian siap.
 2. Admitted (Diterima): Proses diterima ke antrian siap dan bersiap untuk dieksekusi.
 3. Ready (Siap): Proses berada dalam antrian siap dan menunggu untuk dieksekusi oleh CPU.
 4. Scheduler Dispatch (Penjadwalan): Proses dijadwalkan untuk dieksekusi oleh CPU.
 5. Running (Berjalan): Proses sedang dieksekusi oleh CPU.
 6. Interrupt (Interupsi): Proses dieksekusi dapat dihentikan sementara oleh interupsi untuk memberi jalan pada proses lain.
 7. Waiting (Menunggu): Proses sedang menunggu untuk menyelesaikan I/O atau kejadian lainnya sebelum dapat kembali ke state siap.
 8. I/O or Event Wait (Menunggu I/O atau Kejadian): Proses menunggu penyelesaian I/O atau kejadian lain.
 9. I/O or Event Completion (Penyelesaian I/O atau Kejadian): I/O atau kejadian lain telah selesai, dan proses siap kembali ke state siap.
 10. Exit (Keluar): Proses selesai dan keluar dari eksekusi.
 11. Terminated (Dihentikan): Proses telah selesai dan dihapus dari memori.
1. State-state ini menggambarkan siklus hidup sebuah proses dalam sistem operasi berbasis Linux, mulai dari dibuat hingga dihentikan.

SIMPULAN

Sistem operasi real-time (RTOS) semakin penting dalam aplikasi kritis seperti kontrol industri, telekomunikasi, dan sistem transportasi, di mana ketepatan waktu yang sangat tinggi diperlukan untuk menghindari konsekuensi serius. Tantangan utama dalam mengembangkan RTOS pada platform Windows termasuk manajemen sumber daya yang efisien, penjadwalan tugas yang tepat, dan kompatibilitas dengan berbagai perangkat keras. Meskipun demikian, memanfaatkan ekosistem Windows yang mapan memberikan keuntungan dalam hal integrasi yang lebih baik dengan infrastruktur teknologi informasi yang ada serta memperluas kemampuan Windows ke aplikasi real-time. RTOS pada Windows harus mampu menjaga kestabilan dan kinerja sistem dalam kondisi beban

kerja berat, serta memenuhi persyaratan waktu respons yang ketat untuk aplikasi seperti kendali penerbangan dan proses industri. Untuk meningkatkan efektivitas RTOS pada Windows, diperlukan penelitian lebih lanjut untuk mengembangkan teknik manajemen memori dan sumber daya yang lebih efisien, pengujian ekstensif di berbagai skenario operasi untuk mengevaluasi kinerja, serta kolaborasi dengan industri untuk memahami kebutuhan spesifik dan menguji prototipe dalam lingkungan nyata. Pengembangan dan penerapan algoritma penjadwalan yang lebih canggih juga sangat penting untuk memastikan bahwa tugas-tugas prioritas tinggi mendapatkan akses yang tepat waktu.

REFERENSI

- Dharmadasa, M. K., Lee, K. S. W., & McMahon, T. M. G. (2020). Real-Time Operating Systems for Safety Critical Applications. *IEEE Transactions on Computers*, 69(5), 774-786. <https://doi.org/10.1109/TC.2020.2972154>
- Lee, J. J. S., Kim, C. H., & Choi, R. Y. (2020). Performance Evaluation of Real-Time Operating Systems for Critical Systems. *Real-Time Systems*, 56(1), 45-67. <https://doi.org/10.1007/s11241-019-09395-2>
- Chen, L. T., & Wang, Y. S. (2019). A Comparative Study of Real-Time Operating Systems on Windows and Linux Platforms. *International Journal of Real-Time Systems*, 30(4), 320-336. <https://doi.org/10.1007/s11241-019-09334-z>
- Clark, M. O., & Jenkins, P. B. (2018). Optimizing Real-Time Performance in Linux with PREEMPT-RT. *Journal of Systems and Software*, 140(2), 24-39. <https://doi.org/10.1016/j.jss.2017.10.004>
- Smith, A. S., & Zhang, H. W. (2018). Real-Time Scheduling Algorithms: A Comparative Review. *IEEE Transactions on Parallel and Distributed Systems*, 29(7), 1680-1693. <https://doi.org/10.1109/TPDS.2017.2778352>
- Carter, E. K., & Fong, T. R. (2018). Developing Real-Time Applications on Windows Embedded Compact. *Embedded Systems Letters*, 10(3), 103-109. <https://doi.org/10.1109/LES.2018.2877624>
- Miller, K. R., & Johnson, B. J. (2019). Enhancing Real-Time Capabilities of Linux for High-Performance Computing. *Journal of High Performance Computing Applications*, 31(2), 179-193. <https://doi.org/10.1177/1094342018822097>
- Lewis, N. V., & Scott, M. A. (2018). Evaluating Real-Time System Performance in Industrial Control Applications. *Industrial Informatics*, 14(4), 430-443. <https://doi.org/10.1109/TII.2018.2825589>
- Hernandez, J. R., & Patel, S. L. (2020). Advances in Real-Time Operating Systems for Automotive Applications. *Automotive Electronics Journal*, 12(1), 75-92. <https://doi.org/10.1016/j.autel.2020.100223>
- Thompson, R. E., & Green, P. J. (2019). Challenges in Real-Time Operating Systems for Aerospace Systems. *Aerospace Systems and Engineering*, 15(2), 234-248. <https://doi.org/10.1109/ASE.2019.8888551>