JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Recent Advancements in Real-Time Operating Systems (RTOS): Kernels, Scheduling Algorithms, and Middleware for Real Time Embedded Systems

Rabiyabasari R Kagadagar,
Dept. Electronics and Communication Engineering,
RV College of Engineering,
Bengaluru, India
rabiyabrk.lvs24@rvce.edu.in

Prajwal Pataki,
Dept. Electronics and Communication Engineering,
RV College of Engineering,
Bengaluru, India
pprajwalpramod.lvs24@rvce.edu.in

Rahul N,
Dept. Electronics and Communication Engineering,
RV College of Engineering,
Bengaluru, India
rahuln.lvs24@rvce.edu.in

Pranav Sharma N,
Dept. Electronics and Communication Engineering,
RV College of Engineering,
Bengaluru, India
pranavsharman.lvs24@rvce.edu.in

Dr. Govinda Raju M,
Associate Professor,
Dept. Electronics and Communication Engineering,
RV College of Engineering,
Bengaluru, India
govindarajum.lvs24@rvce.edu.in

***Abstract*: Real-Time Operating Systems (RTOS) are essential for applications that require precise timing and reliability, particularly in embedded systems. This review paper examines recent advancements in RTOS kernels, highlighting innovations in kernel architectures, scheduling algorithms, and middleware solutions. We explore the development of robust kernels such as SylixOS and IROS, which enhance security, performance, and real-time capabilities. Additionally, the integration of General-Purpose Operating Systems (GPOS) with traditional RTOS through compounded RTOS (cRTOS) is discussed, showcasing how this approach allows for richer application features while maintaining stringent timing guarantees. Despite these advancements, challenges such as scalability, security, and energy efficiency persist, necessitating further research to address these issues. This paper underscores the importance of ongoing innovation in RTOS to meet the evolving demands of modern applications and ensure their effectiveness in critical environments.**

Keywords: Real-Time Operating Systems, Scheduling Algorithms, Middleware, Embedded Systems.

I. Introduction

Real-Time Operating Systems (RTOS) are critical for embedded systems that require timely and deterministic responses. This paper aims to provide an overview of recent advancements in RTOS kernels, scheduling algorithms, and middleware solutions. The structure of the paper is as follows: Section II discusses advancements in RTOS kernels, Section III covers scheduling algorithms, Section IV explores middleware solutions, Section V Case and Applications, Section VI Challenges and future research directions and Section VII concludes the paper. Embedded systems have become integral to various industries, including aerospace, automotive, industrial automation, healthcare, and telecommunications. The demand for systems that can process tasks within stringent time constraints has driven research in RTOS. RTOS is an essential component in systems that require predictability and low latency. With the advent of more complex embedded applications, researchers have been striving to improve various aspects of RTOS, including determinism, power efficiency, memory management, and security. This paper aims to provide a structured and detailed review of the most recent developments in RTOS technology and their implications for future applications.

II. Recent Advancements in RTOS Kernels

A. *SylixOS: A Secure and Compatible RTOS with Constant Scheduling on SMP*

Kernel Architecture: SylixOS features a robust kernel designed specifically for real-time applications, emphasizing constant scheduling times across Symmetric Multi-Processing (SMP) systems. This architecture allows the RTOS to maintain predictable performance, which is critical for safety-critical applications in sectors like aerospace and military.

Performance Optimization: The kernel employs advanced scheduling algorithms that ensure tasks are executed within their deadlines, regardless of system load. This is achieved through efficient resource management and dynamic task prioritization.

Security Enhancements: The kernel integrates comprehensive security measures, including network security stacks and isolation mechanisms, to protect against potential threats. This focus on security is increasingly important as RTOS are deployed in connected environments.

B. *Compounded Real-Time Operating Systems for Rich Real-Time Applications*

Integration of GPOS and RTOS: The introduction of compounded RTOS (cRTOS) represents a significant advancement in kernel design. By allowing a General-Purpose Operating System (GPOS) and a traditional RTOS to run concurrently, cRTOS enables applications to leverage the rich features of GPOS while maintaining real-time performance.

Remote System Calls: The cRTOS architecture includes a mechanism for remote system calls, which allows real-time applications to access GPOS functionalities without compromising their timing guarantees. This advancement enhances the flexibility and capability of RTOS kernels, making them suitable for complex applications.

Scheduling Innovations: The paper discusses the development of adaptive scheduling algorithms within the cRTOS framework, which can dynamically adjust to varying workloads and resource availability, further optimizing kernel performance.

C. *IROS: A Hard Real-time Intelligent Robot Operating System*

Real-Time Kernel Development: IROS introduces a hard real-time kernel specifically designed for robotic applications, addressing the limitations of existing systems like ROS. This kernel is optimized for low-latency task execution, ensuring that critical operations meet stringent timing requirements.

Middleware Integration: The kernel architecture of IROS includes a middleware layer that facilitates communication between various robot components. This integration allows for efficient data exchange and coordination among tasks, enhancing the overall performance of the RTOS.

Enhanced Security Features: IROS incorporates advanced security protocols within its kernel to safeguard against cyber threats, ensuring reliable operation in safety-critical environments. This focus on security is essential as robots become more interconnected and integrated into larger systems.

III. Innovations in Scheduling Algorithms

A. Entropy-Based Scheduling in Real-Time Multiprocessor Systems

The research examines the behavior of an entropy-based scheduling layer when used with Global Earliest Deadline First (EDF), Least Laxity First (LLF), and PFair-based (PD2) algorithms. Synthetic task sets with 20 tasks with implicit deadlines were produced with the help of the SimSo tool, and the experiments were repeated 100 times for different numbers of CPUs (2, 4, 6, 8) and utilization levels (0.5, 0.75, 1.0). A command-line interface program called "Bench" was created to undertake scenario generation, algorithm execution, and result display through a web interface and SQLite database. Performance gains were quantified as the difference between the original and entropy-enabled versions of the same implementation using the following formula: Improvement = (OriginalResult – EntropyResult) / OriginalResult. The findings show that although the entropy layer drastically curtails task migrations—and, at times, job migrations—it might add extra overhead in situations where there are more frequent scheduling interrupts.

B. Resource Scheduling and Orchestration to Improve Distributed Reinforcement Learning

This work seeks to enhance the performance of distributed reinforcement learning (DRL) tasks in a Kubernetes setting through the introduction of two new scheduling algorithms: LeaderFirst and Batch Tasks Centralized. The LeaderFirst algorithm maximizes resource utilization by separating the scheduling of Learners and Actors, employing a weighted scoring system that favors CPU for Learners and GPU for Actors. Conversely, the Batch Tasks Centralized approach reduces communication overheads through co-location of tasks belonging to the same training batch on one node to avoid resource fragmentation. The experimental framework was executed on a Kubernetes deployment environment with 20 emulated DRL tasks, and performance evaluation was measured in terms of pending task quantities, resource utilization effectiveness, and task completion times. Comparative analysis with local Kubernetes scheduling algorithms, including LeastRequestedPriority and BalancedResourceAllocation, showed that both of the proposed algorithms provide considerable reductions in resource waste and task performance improvement.

C. Smart Energy Management System (SEMS) Based on IoT

The study designs a Smart Energy Management System (SEMS) that utilizes a new decisive algorithm for real-time optimal scheduling of loads with the objective of improving energy efficiency and averting shortages. The architecture of the system includes a central SEM Unit which serves as a gateway for users and utilities, smart sockets for local monitoring and control of electrical loads, and a communication module utilizing XBee Series-2 devices for wireless connection. The control algorithm first gathers power consumption information from enrolled appliances, identifies violations of a pre-set maximum demand limit (MDL), and subsequently energizes high-priority appliances in response, performing peak load analysis with user notification. Real-time user input is provided through an LCD interface for displaying energy measurements and priority modification. Laboratory experiments with different loads (lights, fans, and laptops) under different conditions and Time of Use tariff environments showed that the combined cost-optimization technique and IoT-based monitoring efficiently controls energy usage without violating MDL limits.

D. Real-Time Cyber/Physical Interaction for Peak Load Optimization in CPES

A real-time scheduling methodology, called SweetSleep (SwS), to minimize peak power usage in Cyber-Physical Energy Systems (CPES) is introduced in this paper. CPES is presented as a multiple-input multiple-output (MIMO) control system, wherein electrical loads are handled by a cyber scheduler utilizing differential equations for characterizing system behavior. The suggested scheduling method integrates a two-dimensional bin-packing algorithm and a multi-processor real-time scheduling algorithm in a coordinated approach to minimize concurrent executions resulting in peak loads. Furthermore, the approach establishes a policy for non-schedulable load management where controllable loads are activated during low non-controllable consumption periods. The method also covers the impact of time quantization in practical applications by considering error bounds and statistically comparing performance with other techniques, including Adaptive Real-Time Scheduling and Pulse Width Modulation. In general, the SwS approach proves its capability to significantly decrease peak loads without compromising system scalability and performance.

IV. Middleware Enhancements for Real-Time Systems

A. Adaptive Communication Middleware for Distributed Real-Time Systems

Scalable Architecture: The proposed middleware introduces a modular and scalable architecture that supports adaptive communication across distributed real-time systems .This design allows the middleware to adjust its communication patterns based on system load and network conditions, improving overall system responsiveness and reliability.

Low-Latency Communication: The middleware leverages lightweight communication protocols and optimized data serialization methods to minimize transmission delays [8]. These enhancements ensure that time-critical messages are delivered within strict deadlines, which is essential for high-performance real-time applications in sectors like industrial automation and autonomous systems.

Dynamic Resource Management: The middleware integrates intelligent resource allocation techniques that monitor system performance and dynamically adjust task scheduling and memory usage.This helps maintain consistent real-time performance even under varying workloads and environmental conditions, improving the predictability of task execution.

B. Security-Enhanced Middleware for Medical Real-Time Systems

End-to-End Encryption: The middleware incorporates robust encryption protocols to secure communication between medical devices and healthcare systems. This ensures that sensitive patient data remains protected from unauthorized access and cyber threats, which is critical for compliance with healthcare regulations like HIPAA and GDPR.

Authentication and Access Control: Advanced user authentication mechanisms and fine-grained access control policies are implemented within the middleware, ensuring that only authorized personnel can access critical system components. This prevents unauthorized access and ensures data integrity.

Real-Time Monitoring and Threat Detection: The middleware features real-time threat detection capabilitieusing machine learning algorithm. It continuously monitors system activity, identifying and mitigating security threats before they impact system performance. Such proactive security measures are particularly important in healthcare, where system failures can have life-critical consequences.

C. AI-Driven Middleware Optimization

Predictive Scheduling: AI-based predictive models are integrated into the middleware to anticipate task execution patterns and dynamically adjust scheduling strategie. This helps reduce task contention and improves real-time response times, particularly in complex, multi-threaded environment.

Self-Healing Mechanisms: The middleware employs machine learning algorithms to identify and recover from system anomalies automatically. This enhances the fault tolerance and reliability of real-time systems, reducing downtime and improving overall stability. AI-driven fault detection and correction help maintain system consistency even under high-stress conditions.

V. Case Studies and Applications

The practical applications of these advancements are evident in various domains:

Robotics and Automation: RTOS integration with microcontrollers like ESP32 enhances multitasking control in automation industries. These applications require real-time responsiveness for motor control and sensor fusion.

Smart Home Automation: RTOS implementation in smart home devices ensures precise timing and predictable responses. Real-time scheduling is critical for ensuring synchronization among various smart devices.

Automotive Systems: Advanced driver-assistance systems (ADAS) rely on real-time task execution for sensor processing, decision-making, and actuation. RTOS advancements in this domain have led to safer and more reliable vehicle automation.

Aerospace and Defense: RTOS is widely used in avionics systems to meet stringent safety and reliability standards. Recent research focuses on fault-tolerant RTOS designs that can handle mission-critical operations in extreme conditions.

Healthcare and Medical Devices: Real-time systems play a crucial role in medical monitoring devices and robotic surgery. RTOS advancements have improved the accuracy and reliability of these applications, ensuring patient safety.

VI. Challenges and Future Research Directions

Despite the significant advancements made in Real-Time Operating Systems (RTOS), several challenges persist that require ongoing research and innovative solutions. Below are some of the key challenges and potential future research directions in the field:

A. Scalability

As real-time applications become increasingly complex, ensuring that RTOS can scale effectively is a major challenge. Many modern applications, such as those in autonomous vehicles or smart cities, involve numerous tasks and processes that must be managed simultaneously. Future research should focus on developing RTOS architectures that can efficiently handle a growing number of tasks without compromising performance. This includes exploring dynamic task management techniques and adaptive scheduling algorithms that can respond to changing workloads in real-time.

B. Security and Reliability

The integration of Internet of Things (IoT) devices and cloud computing with RTOS introduces significant security and reliability concerns. As more devices connect to the internet, the risk of cyberattacks increases, making it essential to implement robust security mechanisms within RTOS. Future research should aim to develop advanced security protocols that can protect against unauthorized access and ensure data integrity. Additionally, enhancing the reliability of RTOS in critical applications, such as healthcare and industrial automation, is crucial. This may involve creating fault-tolerant systems that can maintain functionality even in the event of hardware or software failures.

C. Heterogeneous Architectures

Modern computing environments often consist of heterogeneous architectures, which include a mix of CPUs, GPUs, and FPGAs. Managing scheduling and resource allocation across these diverse platforms presents a unique challenge for RTOS. Future research should focus on developing scheduling algorithms that can effectively allocate resources based on the specific capabilities of each type of processor. This includes exploring techniques for load balancing and optimizing task execution across different hardware components to maximize performance and efficiency.

D. Energy Efficiency

With the growing emphasis on sustainability and energy conservation, optimizing power consumption in RTOS while maintaining real-time performance is becoming increasingly important. Future research should investigate methods to reduce energy usage in real-time systems, particularly in battery-operated devices such as drones and wearable technology. This may involve developing energy-aware scheduling algorithms that prioritize tasks based on their energy requirements and exploring hardware-software co-design approaches to enhance energy efficiency.

VII. Conclusion

In summary, Real-Time Operating Systems (RTOS) are essential for modern embedded systems, particularly in applications requiring timely and deterministic responses. This review highlights significant advancements in RTOS kernels, scheduling algorithms, and middleware solutions, showcasing innovations like SylixOS and IROS that enhance security, reliability, and performance.

Despite these advancements, challenges such as scalability, security, and the management of heterogeneous architectures remain. Future research must focus on developing scalable RTOS that can efficiently handle complex applications, robust security mechanisms to protect against cyber threats, and energy-efficient solutions that maintain real-time performance.

Overall, ongoing advancements in RTOS are crucial for meeting the demands of future applications, ensuring that these systems remain capable and versatile in an increasingly interconnected world.

References

[1] C. -F. Yang and Y. Shinjo, "Compounded Real-Time Operating Systems for Rich Real-Time Applications," in IEEE Access, vol. 13, pp. 26079-26104, 2025, doi:10.1109/ACCESS.2025.3538561.

[2] Y. Zhang, H. Han, J. Jiao, G. Xu, G. Chen and K. Huang, "Brief Industry Paper: SylixOS: A Secure and Compatible

RTOS with Constant Scheduling on SMP," 2021 IEEE 27th Real-Time and Embedded Technology and Applications Symposium (RTAS), Nashville, TN, USA, 2021, pp.449-452, doi: 10.1109/RTAS52030.2021.00049.

[3] C. Li, G. Nan and P. Liu, "IROS: A Hard Real-time Intelligent Robot Operating System," 2024China Automation Congress (CAC), Qingdao, China, 2024, pp. 3673-3678, doi: 10.1109/CAC63892.2024.10864617.

[4] C. A. Rincon, D. Rivas and A. M. K. Cheng, "Entropy-based scheduling performance in real-time multiprocessor systems," 2023 57th Annual Conference on Information Sciences and Systems (CISS), Baltimore, MD, USA, 2023, pp. 1-5, doi: 10.1109/CISS56502.2023.10089704.

[5] M. Li, F. Wu, Q. Kou, L. Qian, X. Chen and X. Lan, "Resource Orchestration and Scheduling Algorithms for Enhancing Distributed Reinforcement Learning," 2023 China Automation Congress (CAC), Chongqing, China, 2023,pp.8450-8455,
doi:10.1109/CAC59555.2023.10451295.

[6] De Martini, Daniele & Benetti, Guido & Facchinetti, Tullio. (2024). Real-time cyber/physical interplay in scheduling for peak load optimization in Cyber-Physical Energy Systems. Intelligent Systems with Applications. 22. 200380. doi: 10.1016/j.iswa.2024.200380.

[7] Challa Krishna Rao, Sarat Kumar Sahoo, Franco Fernando Yanine,"Design and deployment of a novel decisive algorithm to enable real-time optimal load scheduling within an intelligent smart energy management system based on IoT,"Energy Reports, Volume 12, 2024, Pages 579-592, ISSN 2352-4847,
DOI: 10.1016/j.egyr.2024.06.030

[8] C. Chen and J. Li, "An embedded software middleware that supports PCS," 2023 IEEE Sustainable Power and Energy Conference (iSPEC), Chongqing, China, 2023, pp. 1-4, doi: 10.1109/iSPEC58282.2023.10402926.

[9] Schöpping T, Kenneweg S, Hesse M, Rückert U. μRT: A lightweight real-time middleware with integrated validation of timing constraints. Front Robot AI. 2023 Mar 21;10:1081875. doi: 10.3389/frobt.2023.1081875. PMID: 37025423; PMCID: PMC10070970.