International Journal Software Engineering and Computer Science (IJSECS)
6 (2), 2026, 696-704
Published Online August 2026 in IJSECS (http://www.journal.lembagakita.com/index.php/ijsecs)
P-ISSN: 2776-4869, E-ISSN: 2776-3242. DOI: https://doi.org/10.35870/ijsecs.v6i2.7560.

RESEARCH ARTICLE **Open Access**

Performance Trade-Off Between Windows 10 and Windows 11: A Comparative Analysis of CPU and Memory Behavior Using Performance Monitor and LatencyMon

Ekie Revsie Akbar [1], **Muhammad Fadhiil Rachman** [2], **Tiara Fitriana** [3], **Try Mulyoto** [4*], **Jan Everhard Riwurohi** [5]

[1,2,3,4*,5] Master of Computer Science Program, Faculty of Information Technology, Universitas Budi Luhur South Jakarta City, Special Capital Region of Jakarta, Indonesia.

**Corresponding author*: trymulyoto@gmail.com.

Received: June 15, 2026; Accepted: June 28, 2026; Published: August 1, 2026.

Abstract: The operating system consumes processor and memory resources and affects kernel-level responsiveness before user applications run, and these resource demands may differ across operating system versions. This study compared Windows 10 and Windows 11 under a realistic and reproducible video transcoding workload performed with HandBrake, measuring CPU utilization, hard page faults, and interrupt-to-process latency. A quantitative experimental method was used. Performance Monitor recorded CPU and memory behavior, while LatencyMon measured kernel timer, interrupt service routine, deferred procedure call, and interrupt-to-process latency. An important constraint is that the two operating systems were tested on physically different devices with processors from different generations: a 6th-generation dual-core Intel Core i7-6600U and an 11th-generation quad-core Intel Core i7-1165G7. Therefore, the measured differences cannot be attributed to the operating system version alone. Windows 11 showed more sustained CPU utilization and recorded fewer hard page faults than Windows 10 (22,480 versus 50,539), although the difference is also subject to the unequal monitoring durations. Windows 10 recorded a lower highest interrupt-to-process latency (1,147.60 µs versus 1,430.70 µs) and a lower average interrupt latency (10.10 µs versus 22.44 µs), while both systems remained within the conventional real-time suitability threshold. The findings indicate that the observed performance differences are entangled with processor generation and other hardware-related effects. Therefore, a dual-boot configuration on identical hardware is recommended in future studies to isolate the effect of the operating system version.

Keywords: CPU Utilization; HandBrake Transcoding; Hard Page Fault; Interrupt Latency; LatencyMon; Windows 10; Windows 11.

1. Introduction

The operating system is the central software layer that manages hardware resources and provides essential services to applications, mediating between the physical machine and the programs that run on it. Among its responsibilities, the management of the central processing unit (CPU) and main memory is an important factor in system performance and stability. Modern computing increasingly relies on multicore processors, making the way an operating system distributes workloads across processor cores and manages memory relevant to system responsiveness (Fauzia Fredella & Ulya Rahman, 2025a; Bagas Sulistio *et al.*, 2025). The resource behavior of an operating system therefore affects the amount of hardware capacity available for applications and other user activities.

Every operating system imposes a baseline cost on the machine on which it runs. Background services, the kernel scheduler, memory management processes, and security subsystems consume CPU resources and occupy memory before a user application is started (Ali Ridho, 2025). This baseline cost, often described as operating system overhead, affects the amount of system capacity available for productive workloads. In addition to aggregate CPU and memory utilization, kernel-level metrics such as interrupt-to-process latency, deferred procedure call (DPC) execution time, and hard page fault counts can provide information about system responsiveness and memory activity (Resplendence Software, 2023). These metrics are therefore useful as complementary indicators when evaluating operating system behavior under workload conditions.

The main problem addressed in this study is the difficulty of evaluating operating system behavior independently of the hardware on which the operating system runs. Successive Windows versions are often installed on devices with different processor generations, making it difficult to determine whether an observed performance difference results from the operating system or from changes in hardware capability. This issue is particularly relevant when comparing processors with different core counts, microarchitectures, and memory capabilities. A comparison between a dual-core processor from an older generation and a quad-core processor from a newer generation, for example, introduces a hardware-related performance bias that cannot be separated from the operating system effect without a controlled hardware configuration (Rahmika *et al.*, 2025). Recognizing this limitation is necessary when interpreting performance differences between Windows versions.

Previous studies have examined operating system performance from several perspectives. Studies comparing Windows 10 and Windows 11 have reported differences in areas such as boot performance, workload distribution across processor cores, and memory behavior, particularly on newer multicore hardware (Bagas Sulistio *et al.*, 2025). Other benchmarking studies have compared Windows 11 with Ubuntu on similar hardware and examined CPU utilization and memory-related behavior. Comparisons of different Windows and Linux versions have also indicated that newer operating system releases do not necessarily reduce resource consumption under every workload, because additional system features and background services may offset improvements in scheduling or resource management (Al Fajar *et al.*, 2025). Research on memory behavior under computational workloads further indicates that memory pressure, resident memory requirements, and memory bandwidth can influence system performance (Bagas Sulistio *et al.*, 2025). These studies indicate that operating system behavior can be assessed through measurable resource and responsiveness indicators, but differences in hardware, operating system families, and selected performance metrics make direct Windows 10 and Windows 11 comparisons difficult.

The gap addressed by this study is the limited availability of version-to-version comparisons between Windows 10 and Windows 11 that combine resource utilization measurements with kernel-level responsiveness indicators while explicitly acknowledging hardware differences. A dual-boot configuration on identical hardware would provide a more controlled basis for isolating the operating system variable. However, such a configuration was not feasible in the present study. Instead, two devices with comparable RAM and storage characteristics but different processor generations were used. The hardware difference is therefore treated as a major limitation rather than as a factor that can be disregarded. This study combines Performance Monitor resource logging with LatencyMon kernel-level measurements during a reproducible video transcoding workload. The analysis also considers the extent to which the observed differences may reflect hardware characteristics rather than operating system behavior.

The objectives of this research are: (1) to measure and compare CPU utilization and memory behavior between Windows 10 and Windows 11 under a reproducible video transcoding workload using Performance Monitor; (2) to measure and compare kernel responsiveness, including interrupt-to-process latency, DPC-related metrics, and hard page fault activity, using LatencyMon; (3) to examine the temporal patterns observed in the Performance Monitor traces of each system; and (4) to assess the extent to which the observed differences can be attributed to the operating system version given the hardware differences between the two devices.

2. Related Work

2.1 Virtual Memory, Paging, and Hard Page Faults

Memory management is an important determinant of system performance, particularly when an application's working set approaches the limits of available physical memory. Research using instrumentation of standard workload suites has shown substantial variation in resident set sizes and has indicated that workloads exceeding cache capacity can become increasingly constrained by the memory hierarchy rather than compute throughput (Fauzia Fredella & Ulya Rahman, 2025a). Other studies have examined the relationship between virtual memory footprint, page fault overhead, memory bandwidth, and hardware prefetching, indicating that hardware characteristics can influence paging-related performance [5]. Studies of page replacement and working-set behavior further indicate that hard page faults, which require data to be retrieved from storage, can contribute to increased system latency under memory pressure (Apriana *et al.*, 2024; Fauzia Fredella & Ulya Rahman, 2025a). These findings support the use of hard page fault activity as a complementary performance metric while indicating that differences in page fault behavior should not be attributed solely to operating system policy.

2.2 Cross-Platform and Cross-Version Scheduling Analyses

A second body of research examines how operating systems distribute workloads across processor cores. Previous studies comparing Windows 10 and Windows 11 on multicore hardware have reported differences in boot performance, core utilization, and memory behavior, with some improvements observed on Windows 11 (Bagas Sulistio *et al.*, 2025). Comparisons between Windows 11 and Ubuntu 22.04 LTS have also examined CPU utilization and context-switch behavior under workload conditions [10]. Other scheduling studies indicate that operating system version and scheduling strategies can produce measurable differences in resource utilization, although the magnitude and direction of these differences depend on workload characteristics and system configuration (Al Fajar *et al.*, 2025; Yusnanto *et al.*, 2022). Research on multicore scheduling further indicates that processor core count and threading capability can influence observed CPU utilization and task distribution (Ghufron Malik *et al.*, 2024). Collectively, these studies indicate that CPU utilization is a useful but insufficient indicator of operating system performance and that observed differences should be interpreted alongside hardware characteristics and workload conditions.

2.3 Kernel Latency and Real-Time Responsiveness Measurement

A third group of studies concerns the measurement of kernel-level responsiveness. Interrupt-to-process latency measures the time associated with processing an interrupt and transferring execution toward the relevant process or thread, while related measurements include interrupt service routine (ISR) and deferred procedure call (DPC) execution times (Resplendence Software, 2023). LatencyMon provides measurements of ISR and DPC activity, interrupt-to-process latency, and hard page fault activity, making it useful for examining responsiveness characteristics that are not captured by aggregate CPU utilization alone. LatencyMon also provides a real-time audio suitability assessment based on observed latency values, including a commonly referenced threshold of approximately 2,000 µs in its reporting context (Resplendence Software, 2023). These measurements provide a basis for evaluating kernel-level responsiveness during workload execution and complement conventional CPU and memory utilization measurements.

2.4 Comparison with Previous Studies

Table 1 summarizes representative studies relevant to the present research and identifies their principal focus and limitations. Previous studies have examined memory management, Windows memory behavior, cross-platform multitasking, and CPU scheduling, but the scope of operating system comparisons and selected performance metrics varies across studies. The present study combines a standardized video transcoding workload with Performance Monitor and LatencyMon measurements to examine both resource utilization and kernel-level responsiveness. It also explicitly considers the hardware difference between the two test devices when interpreting the results.

Table 1. Comparison with Previous Studies

Study	Focus / Metrics	Limitation
Angella Harsono *et al.* (2022)	Memory management	No experimental validation
Ridho & Avorizano (2025)	Windows memory behavior	Single OS only
Ghufron Malik *et al.* (2024)	Windows vs. Linux multitasking	Not Windows 10 vs. Windows 11
Verano & Purnamasari (2025)	CPU and memory performance	No latency analysis

Fadillah Putra *et al.* (2024)	CPU scheduling and responsiveness	No Windows 10 comparison
This study	Windows 10 vs. Windows 11; CPU utilization, ISR, DPC, interrupt latency, and hard page faults	Different processor generations

The comparison indicates that the present study differs from previous work primarily in its combination of resource utilization and kernel-level responsiveness measurements under a standardized video transcoding workload. However, because the two operating systems were tested on different processor generations, the study does not provide a fully hardware-controlled comparison. This limitation is therefore considered when interpreting the observed performance differences.

3. Methodology

This study used a quantitative experimental method in which a standardized workload was applied and the outcomes were recorded as numerical data. The treatment of interest was the operating system version, while the measured outcomes included CPU utilization, memory behavior, hard page fault activity, and kernel-level latency. The experimental design, workload, measurement tools, measurement procedure, and hardware constraint are described below.

3.1 Device Configuration and the Hardware Constraint

Two laptops were used, with one operating system installed on each device. Device A ran Windows 10 on an Intel Core i7-6600U processor with 2 cores and 4 threads, while Device B ran Windows 11 on an Intel Core i7-1165G7 processor with 4 cores and 8 threads. The processors belong to the 6th and 11th generations, respectively. Both devices had 16 GB of RAM and SSD storage, but they differed substantially in processor generation, core count, microarchitecture, cache characteristics, and memory subsystem capabilities. Therefore, the observed performance differences cannot be attributed to the operating system version alone. A dual-boot configuration on identical hardware would have provided a more controlled design by isolating the operating system variable. However, such a configuration was not feasible within the available test environment. To reduce differences in the experimental conditions, both systems were tested using the same workload, source video, encoder configuration, and measurement tools. The hardware difference is treated as a major limitation throughout the analysis, and the results are interpreted as indicative rather than as direct evidence of operating system superiority. Both systems were configured using their respective Windows installations and were updated before testing. Default operating system settings were maintained throughout the experiment, and no additional optimization software or deliberate system modifications were applied. Table 2 summarizes the hardware configuration used in the study.

Table 2. System Configuration

Component	Windows 10 System	Windows 11 System
Processor	Intel Core i7-6600U	Intel Core i7-1165G7
CPU configuration	2 cores, 4 threads	4 cores, 8 threads
Processor generation	6th generation	11th generation
Memory (RAM)	16 GB	16 GB
Storage	SSD	SSD

3.2 Workload

The workload consisted of a video transcoding task performed using HandBrake. The same source video file was transcoded on both devices using the same preset and encoder configuration. The workload was selected because video transcoding provides a repeatable and computationally intensive task that places substantial demand on processor resources and can be executed consistently across both systems. Its multithreaded nature also allows differences in processor capability and workload distribution to be observed during execution (HandBrake Documentation, 2024; Mercat *et al.*, 2024).

3.3 Measurement Tools

Two primary tools were used to capture complementary aspects of system behavior. Performance Monitor (PerfMon) was used as the primary resource monitoring tool to record processor utilization, available memory, committed bytes, and hard page fault activity at a fixed sampling interval. The resulting time-series measurements were used to examine resource utilization during the workload. LatencyMon, developed by Resplendence Software, was used to evaluate kernel-level responsiveness. It measured interrupt-to-process latency, interrupt service routine (ISR) execution time, deferred procedure call (DPC) execution time, and hard page fault activity. LatencyMon also reports system suitability for real-time audio workloads based on observed latency values (Resplendence Software, 2023). The combination of Performance Monitor and LatencyMon enabled the study to examine both resource utilization and kernel-level responsiveness.

3.4 Measurement Procedure and Data Handling

Before measurement, each device was freshly restarted and allowed to reach a settled state. The same video transcoding workload was then executed while Performance Monitor and LatencyMon recorded system behavior. The quantitative results reported in Section 4 were obtained from a single recorded monitoring session for each device rather than from an average of repeated sessions. Consequently, the findings are treated as indicative rather than statistically conclusive. The monitoring durations differed between the devices, with approximately 15 minutes recorded on Device A and approximately 26 minutes on Device B. Where cumulative metrics were compared across unequal monitoring durations, the values were normalized by monitoring duration. The original values and the corresponding normalized values should be reported consistently in Section 4 so that the effect of the different recording durations can be evaluated.

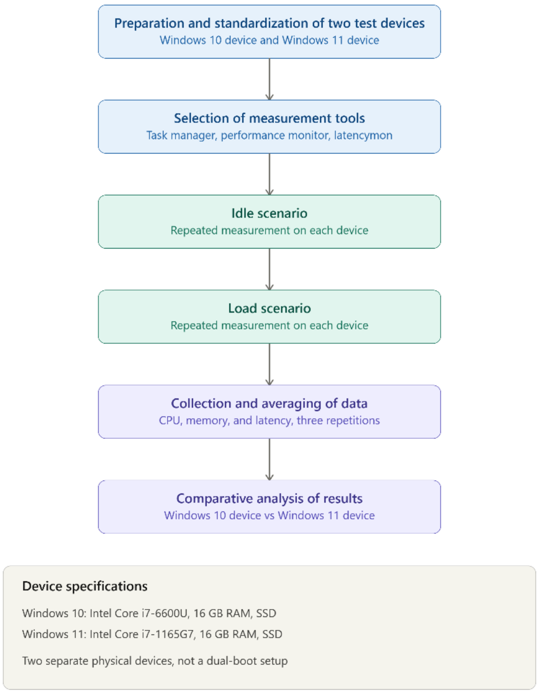

Figure 1. Experimental Flow of the Study

The experimental procedure consisted of the following stages: preparation and standardization of the two test devices; configuration of the measurement tools; execution of the standardized video transcoding workload; continuous recording using Performance Monitor and LatencyMon; extraction of CPU, memory, and latency measurements from the recorded sessions; normalization of cumulative metrics where necessary; and comparative analysis of the Windows 10 and Windows 11 results. Because only one monitoring session was used for each device, no averaging across repeated experimental sessions was performed.

4. Result and Discussion

4.1 Results

This section presents the measured CPU utilization and kernel responsiveness of the Windows 10 and Windows 11 systems under the HandBrake video transcoding workload. Performance Monitor was used to record CPU utilization, while LatencyMon was used to measure interrupt-to-process latency, ISR and DPC execution times, and hard page fault activity. Table 3 reports the CPU utilization recorded by Performance Monitor. Both systems reached a maximum CPU utilization of 100%, indicating that the workload was able to fully utilize the available processor capacity on both devices during the recorded interval. The Windows 10 system recorded a slightly higher average CPU utilization of 70.52% compared with 67.10% on the Windows 11 system. The minimum CPU utilization was 4.59% on Windows 10 and 1.85% on Windows 11.

Table 3. Performance Monitor Results

Operating System	Average CPU Utilization	Maximum CPU Utilization	Minimum CPU Utilization	Monitoring Duration
Windows 10 (Intel Core i7-6600U)	70.52%	100%	4.59%	1m 40s
Windows 11 (Intel Core i7-1165G7)	67.10%	100%	1.85%	1m 40s

Table 4 presents the kernel responsiveness measurements obtained from LatencyMon. The Windows 11 system recorded a higher maximum interrupt-to-process latency of 1,430.70 µs and a higher average latency of 22.44 µs, compared with 1,147.60 µs and 10.10 µs, respectively, on the Windows 10 system. The Windows 10 system recorded a higher total hard page fault count, with 50,539 events compared with 22,480 events on Windows 11. The highest ISR execution time was 191.55 µs on Windows 10 and 206.55 µs on Windows 11, while the highest DPC execution time was 1,339.13 µs and 738.94 µs, respectively. The LatencyMon report for the Windows 10 system also identified a network-related DPC issue associated with ndis.sys, which was not reported for the Windows 11 system.

Table 4. LatencyMon Results

Operating System	Highest Interrupt-to-Process Latency	Average Interrupt Latency	Highest ISR	Highest DPC	Hard Page Faults	Monitoring Duration
Windows 10	1,147.60 µs	10.10 µs	191.55 µs	1,339.13 µs	50,539	0:15:19
Windows 11	1,430.70 µs	22.44 µs	206.55 µs	738.94 µs	22,480	0:26:54

Figures 2 and 3 show the temporal CPU utilization patterns recorded by Performance Monitor for the Windows 10 and Windows 11 systems, respectively. The Windows 10 trace shows substantial variation in CPU utilization during the recorded interval, including periods of lower utilization followed by a rise to 100%. Toward the end of the recording, CPU utilization decreases as the encoding workload approaches completion.

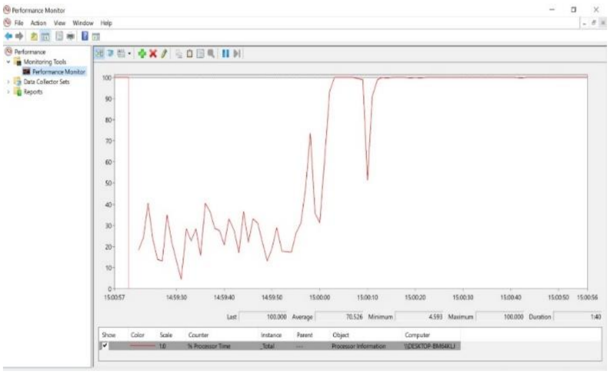

Figure 2. CPU utilization over time on the Windows 10 device (Performance Monitor)

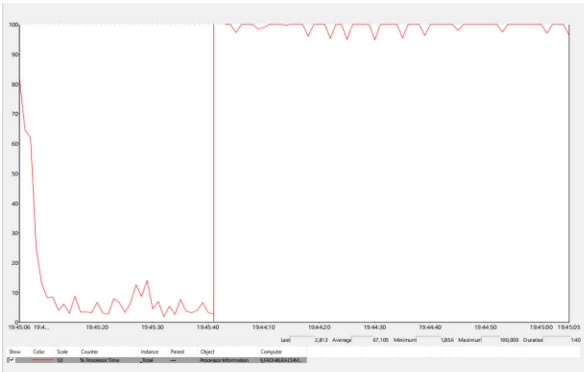

Figure 3. CPU utilization over time on the Windows 10 device (Performance Monitor)

The Windows 11 trace shows a more sustained high-utilization pattern, with CPU utilization remaining close to full utilization for most of the recorded interval and exhibiting only brief decreases. This pattern indicates that the workload maintained a high level of processor activity for a substantial portion of the recorded period. The temporal traces provide additional information beyond the average CPU utilization values in Table 3. Although the Windows 10 system recorded a slightly higher average utilization, the time-series pattern indicates greater variation during the recorded interval. In contrast, the Windows 11 system maintained a more consistently high utilization level. These observations indicate that average CPU utilization alone does not fully describe the temporal behavior of the workload.

4.2 Discussion

The results must be interpreted in light of the hardware constraint described in Section 3.1. The two systems use processors from different generations and with different core and thread configurations. The Windows 10 system uses a 6th-generation Intel Core i7-6600U with two cores and four threads, whereas the Windows 11 system uses an 11th-generation Intel Core i7-1165G7 with four cores and eight threads.

Therefore, the observed differences cannot be attributed to the operating system version alone. The hard page fault results illustrate this limitation particularly clearly. Windows 10 recorded 50,539 hard page faults, compared with 22,480 on Windows 11. The lower count on Windows 11 indicates less hard page fault activity during the recorded period. However, this difference cannot by itself establish that Windows 11 provides superior memory management because processor architecture, memory subsystem characteristics, storage behavior, background activity, and workload execution can also influence page fault activity. The difference in monitoring duration further complicates direct comparison. Normalizing cumulative metrics by monitoring duration can reduce the effect of unequal recording periods, but it cannot remove the underlying hardware confound.

The CPU utilization traces also reflect both workload execution and hardware characteristics. Both systems reached 100% CPU utilization, indicating that the workload placed a substantial computational demand on each processor. However, the two processors have different levels of parallel processing capability. The four-core, eight-thread processor in the Windows 11 system can execute more concurrent threads than the two-core, four-thread processor in the Windows 10 system. Consequently, the observed utilization patterns cannot be interpreted as a direct measurement of differences in Windows scheduling behavior. The latency measurements provide a different pattern. Windows 11 recorded a higher maximum interrupt-to-process latency than Windows 10, at 1,430.70 µs compared with 1,147.60 µs. The average latency was also higher on Windows 11, at 22.44 µs compared with 10.10 µs. In contrast, the highest DPC execution time was higher on Windows 10, at 1,339.13 µs compared with 738.94 µs on Windows 11. These results show that the different latency indicators do not all favor the same operating system.

The Windows 10 LatencyMon report also identified a network-related DPC issue associated with ndis.sys. However, the Windows 10 system still recorded a lower maximum interrupt-to-process latency than Windows 11. This observation indicates that the presence of a flagged driver does not necessarily correspond directly to the highest latency value in this comparison. The result should nevertheless be interpreted cautiously because the two systems use different hardware and driver environments. The difference in monitoring duration is another factor that should be considered when interpreting the maximum latency values. Windows 10 was monitored for 15 minutes and 19 seconds, whereas Windows 11 was monitored for 26 minutes and 54 seconds. The longer Windows 11 recording period may have provided more opportunity for transient latency events to be captured. Therefore, the difference in maximum latency should not be interpreted as definitive evidence that Windows 11 has poorer kernel responsiveness.

The CPU utilization results similarly do not support a direct conclusion that one operating system is more efficient than the other. The average CPU utilization was 70.52% for Windows 10 and 67.10% for Windows 11, a relatively small difference. Because the measurements were obtained from different processors and from a single recorded session for each device, the difference may reflect workload timing, background activity, or processor characteristics rather than a systematic operating system effect. The results do not support a simple conclusion that Windows 11 is faster or that Windows 10 is more responsive. Windows 11 showed lower hard page fault activity and a more sustained CPU utilization pattern, while Windows 10 showed lower average and maximum interrupt-to-process latency and a lower highest ISR execution time. At the same time, Windows 10 recorded a higher highest DPC execution time and a substantially higher hard page fault count. Because each operating system was tested on different processor hardware and only one monitoring session was used per device, these differences should be regarded as observations under the tested conditions rather than definitive evidence of operating system superiority. A comparison using identical hardware and a dual-boot configuration would be required to isolate the effect of the operating system version.

5. Conclusion and Recommendations

This study compared Windows 10 and Windows 11 under a reproducible HandBrake video transcoding workload using Performance Monitor for resource monitoring and LatencyMon for kernel-level latency measurement. The Windows 11 system, which used an 11th-generation quad-core processor, recorded fewer hard page faults and showed a more sustained high-CPU-utilization pattern during the recorded workload. However, its average CPU utilization was slightly lower than that of the Windows 10 system. For the interrupt-to-process latency measurements, Windows 10 recorded a lower highest latency of 1,147.60 µs compared with 1,430.70 µs on Windows 11. The average interrupt latency was also lower on Windows 10, at 10.10 µs compared with 22.44 µs on Windows 11. Thus, for these measured latency indicators, Windows 10 recorded lower values under the tested conditions. Both systems remained below the conventional real-time suitability threshold used in the study. The Windows 10 system also recorded a substantially higher number of hard page faults, with 50,539 events compared with 22,480 on Windows 11. This difference indicates lower hard page fault activity on Windows 11 during the recorded period, but it cannot be interpreted as direct evidence of superior operating system memory management because the two systems used different processor generations, core configurations, and hardware environments. Most importantly, the findings cannot be attributed to the operating system version alone. The Windows 10 system used a 6th-generation Intel Core i7-6600U with two cores and four threads, whereas the Windows 11 system used an 11th-generation Intel Core i7-1165G7 with four cores and eight threads. Therefore, the measured differences represent the combined effects of operating system, hardware, driver, and workload conditions.

The main contribution of this study is the combination of resource monitoring and kernel-level latency measurement during a reproducible video transcoding workload. Performance Monitor provided information on CPU utilization, while LatencyMon provided interrupt-to-process latency, ISR and DPC execution times, and hard page fault activity. Another contribution is the explicit treatment of the hardware constraint. Rather than attributing all observed differences directly to Windows 10 or Windows 11, the study identifies the different processor generations as a major confounding factor. The results therefore demonstrate the importance of combining resource utilization measurements with kernel-level responsiveness indicators and of controlling hardware when comparing operating system versions. This study has several limitations. Windows 10 and Windows 11 were tested on different devices with different processor generations, core counts, and hardware characteristics. This prevents the operating system version from being isolated as the sole independent variable. In addition, the quantitative results were obtained from a single recorded monitoring session for each device, so statistical significance testing could not be performed. The monitoring durations also differed between the two systems, requiring the duration of observation to be considered when comparing cumulative metrics. These limitations mean that the findings should be interpreted as indicative observations under the tested conditions rather than as statistically conclusive evidence of the superiority of one Windows version over the other. The relative results may differ when the two operating systems are tested on hardware with matched specifications.

Future research should use a dual-boot configuration on the same physical device, with Windows 10 and Windows 11 installed on separate partitions. Using the same processor, memory, and storage would substantially reduce the hardware confounding present in the current study and provide a stronger basis for assessing the contribution of the operating system version. Future experiments should also include multiple independent monitoring sessions for each operating system and use equal monitoring durations. Repeated measurements would provide a basis for calculating descriptive variability and, where the experimental design and sample size permit, conducting statistical significance testing. Driver-related latency should also be investigated through controlled tests. For example, network-related DPC activity associated with ndis.sys could be examined by comparing configurations with and without the relevant network adapter or driver activity while keeping other conditions constant. This would help determine whether driver activity contributes to the observed latency differences. Finally, future studies should extend the workload beyond video transcoding to include memory-bound and latency-sensitive tasks. Using several workload types would provide a broader assessment of whether the observed differences remain consistent across different computational conditions. These measures would allow the contribution of the operating system to be evaluated with substantially less influence from hardware, driver, and workload differences.

Acknowledgment

The authors thank the Department of Informatics Management at Institut Teknologi dan Bisnis Bina Sriwijaya for providing the facilities used in this study, and the lecturers of the Operating Systems course for their guidance and methodological feedback during the research.

References

Al Fajar, R., Lestari, A., & Teknologi Informasi, J. (2025). Analisis perbandingan sistem operasi Windows 11 dan Linux Ubuntu menggunakan metode studi literatur (Studi kasus: Kinerja sistem, keamanan dan biaya). *Jurnal Bitwise, 1*(2). https://jurnal-bitwise.org/

Ali Ridho, A. A. S. (2025). *Analisis manajemen memori sistem operasi Windows dalam menjalankan game open-world*.

Amin Hidayat, D. K. E. N. G. S. (2025). *Analisis implementasi Windows 10 dan 11 di pendidikan: Evaluasi biaya, usability, kinerja*.

Ananda, R. A., & Nama, G. F. (2024). Analisis dan perancangan layanan streaming film berbasis web langganan menggunakan framework NextJS. *Jurnal Informatika dan Teknik Elektro Terapan, 12*(1). https://doi.org/10.23960/jitet.v12i1.3967

Apriana, S. P., Hidayat, R., & Rilvani, E. (2024). Optimalisasi pengguna memori virtual pada sistem operasi Windows 10. *Journal Global Technology Computer, 4*(1), 44–51. https://doi.org/10.47065/jogtc.v4i1.6512

Bagas Sulistio, D., Riski, N., & Aryani, Y. (2025). *Analisis kinerja CPU dan manajemen proses pada sistem operasi Windows saat beban penuh menggunakan Task Manager* (Vol. 1, No. 2). https://jurnal.dharmadhikara.com/sifra.index

Fauzia Fredella, & Ulya Rahman. (2025). Penerapan virtual memory terhadap kinerja CPU, GPU, dan respons multitasking pada Windows 10. *MARS: Jurnal Teknik Mesin, Industri, Elektro dan Ilmu Komputer, 3*(5), 168–178. https://doi.org/10.61132/mars.v3i5.1133

Ghufron Malik, K., Wahyudi, K., Sakti, F. T. A., Ghofar, A., & Anshor, A. H. (2024). Analisa perbandingan manajemen proses multitasking pada sistem operasi Windows dan Linux. *Jurnal Teknik Mesin, Industri, Elektro dan Informatika, 3*(4), 190–199. https://doi.org/10.55606/jtmei.v3i4.4538

Haritsah, M., Widjajarto, A., & Almaarif, A. (2023). Analisis karakteristik antivirus berdasarkan aktivitas malware menggunakan analisis dinamis. *Journal of Information System Research (JOSH), 4*(2), 693–700. https://doi.org/10.47065/josh.v4i2.2908

Jurnal, H., Wildan Kemal, M., & Farizy, S. (2022). Analisis penyebab bottleneck pada suatu komputer menggunakan aplikasi benchmarking CrystalDiskMark dengan metode SWOT. *JUPTI, 1*(3).

Rahmika, B., Aulia Az Zahrah, K., Anggraini, A., Bisnis Bina Sriwijaya, D., & Palembang, K. (2025). *Analisis kinerja perangkat I/O disk (HDD): Perbandingan performa disk menggunakan Task Manager pada aktivitas transaksi file* (Vol. 1, No. 2). https://jurnal.dharmadhikara.com/sifra.index

Verano, D. A., & Purnamasari, E. (2025). Benchmarking kinerja sistem operasi Linux dan Windows dalam pengelolaan proses dan memori. *SMARTICS Journal, 11*(2), 91–96. https://doi.org/10.21067/smartics.v11i2.13850

Wijaya, W., Panjaitan, F., Rizal, S., & Ulfa, M. (2024). Perbandingan kinerja sistem operasi Network Attached Storage: Studi kasus TrueNAS dan XigmaNAS. *Jurnal Indonesia: Manajemen Informatika dan Komunikasi, 5*(1), 255–261. https://doi.org/10.35870/jimik.v5i1.468

Yusnanto, T., Wahyudiono, S., & Wicaksono, H. A. (2022). Analisa kinerja sistem operasi Windows 10 dengan Linux Mint menggunakan aplikasi ZXT CAM, GNOME System Monitor. *Jurnal Riset Ilmiah, 1*(2). https://www.ibm.com/cloud/learn/api