УДК 811.111:004.92:004.4

CHOOSING AN OPERATING SYSTEM FOR A BACKEND, FRONTEND AND DEVOPS PROGRAMMER

Kernazhytski A. U., student
Scientific supervisor – Vanik I. Y., senior lecturer
English language department № 1
Belarusian National University of Technology
Minsk, Republic of Belarus

Choosing an operating system for software development requires balancing performance, tool compatibility, and usability. Let's examine the pros and cons of Linux, macOS, and Windows from the perspective of frontend, backend, and DevOps programming.

Let's start with the most OS-neutral position: frontend development. Frontend developers have relatively simple OS requirements, as their work often revolves around browser-based tools, development environments, and Node.js. The primary needs include compatibility with various browsers and graphic design software.

Linux OS is known for its speed, efficiency, and lightweight nature, Linux allows source code and scripts to run smoothly. However, some graphical tools and browsers may not work out of the box, requiring extra customization, which can be challenging for newcomers. Despite these hurdles, Linux remains a solid option for developers who favor a performance-oriented system.

MacOS offers a highly optimized user experience with seamless multitasking and efficient use of hardware resources. It’s designed for running multiple tasks simultaneously and switching between them easily. The system’s high price, however, is a significant drawback, as it runs on only expensive Apple hardware. On the other hand, hardware and software integrate perfectly, providing enhanced performance and optimization.

Windows is user-friendly OS and has a broad range of development tools, graphic editors, and other useful applications. For instance, Windows offers a multimedia framework called DirectX API. It includes a wide set of applications for game development. Thus, DirectX is highly used by game developers for creating Windows and Xbox video games.

Compared to Linux and macOS, Windows is slower in running scripts, switching between tasks, and composing code [1].

Having considered systems for frontend development, we can move on to backend development. Backend developers generally prioritize performance and memory efficiency. They don't require extensive graphical tools but do need access to various services and efficient system resource allocation [2].

Linux is very efficient and highly optimized regarding resource utilization, excelling in resource optimization with many lightweight distributions that ensure most of the system's power is available for development tasks. The OS's versatility allows it to run smoothly on both low-end machines and high-performance servers. Linux's robust multi-threading, parallel processing, and advanced scheduling features make it an excellent choice for backend work.

MacOS is built on top of Unix, especially BSD variant, meaning that the Unix operating system is the foundation for macOS's core. It offers powerful terminal capabilities that are crucial for backend development. The OS efficiently handles multi-threading and has advanced memory management, which optimizes system resource usage. MacOS also provides superior graphics rendering, which can be beneficial for multimedia and visually intensive applications. Moreover, MacOS is known for its robust security strategies.

Windows support frameworks like ASP.NET for web development and includes DirectX API for game development. While Windows has broad tool compatibility, it falls short of Linux and macOS in terms of performance efficiency, especially for multi-threaded applications [3].

For DevOps engineers, flexibility, customization, and command-line capabilities are critical. They often work with pipelines that deploy to servers, so the OS choice can significantly impact productivity.

The main drawback of relying on Windows is that, well, it's Windows. Compared to Linux, Windows offers very little opportunity to customize, and that restraint can be a drawback from a DevOps perspective.

When it comes to core tasks, such as working from the command line, you're stuck with the tools Microsoft offers you.

Linux is highly customizable. You can set up your system in the way that best suits your workflow. You can use a wide array of tools. You can make lots of changes to add extra security. Given that DevOps values

choice, this is a big benefit. In many cases, DevOps engineers are supporting software delivery pipelines that deploy to Linux server environments. Being able to work from a native Linux environment can make it easier to do things such as test and stage those applications on your local device when it becomes necessary.

Last but not least is macOS. It gives you a native Unix-like environment. You can't run your Linux containers directly on a Mac, but you can at least use some native Unix tools.

Drawbacks for macOS are many. It's not very customizable. It lacks compatibility with some applications and tools. It comes with its own learning curve if you have never used macOS before [4].

When choosing an OS for frontend development, all three options work, though Linux may need extra configuration for graphic tools. Backend developers benefit from Linux's efficiency or macOS's Unix-based capabilities. DevOps engineers generally prefer Linux for its flexibility, security options, and native support for containerized environments, macOS is a solid middle ground, particularly for those who value a Unix-based system and efficient resource management. Windows, while versatile and user-friendly, often lags behind in script performance, efficiency, and system flexibility.

Each OS has strengths and weaknesses, so the choice ultimately depends on the specific needs of your development work.

References

1. What Is the Best OS for Web Design and Web Development – URL: https://ewm.swiss/en/blog/what-best-os-web-design-and-web-development (date of access: 14.03.2025).

2. Which Operating System Is Best Suited for Your Development – URL: https://www.bairesdev.com/blog/which-os-best-suited-for-your-development/ (date of access: 21.03.2024).

3. The Best OS for Programming; But It's for Developers in 2023 – URL: https://cloudzy.com/blog/best-os-for-programming/ (date of access: 09.03.2025).

4. Which Desktop OS Is Best for DevOps? – URL: https://devops.com/which-desktop-os-is-best-for-devops/ (date of access: 17.03.2025).