System Calls in Endless OS

Endless OS is a Linux-based operating system designed for simplicity and accessibility, especially for users with limited internet access. Like other Linux distributions, Endless OS supports system calls, which are essential for communication between user-space applications and the kernel.

System calls in Endless OS are based on the Linux kernel, adhering to POSIX standards. These calls provide the primary interface for performing low-level operations such as:

1. Process Control

- fork(): Create a new process.

- exec(): Execute a new program.

- wait(): Wait for a process to change state.

- exit(): Terminate a process.

2. File Management

- open(), read(), write(), close() for handling files.

- stat(), fstat() for file information.

- mkdir(), rmdir() for directory management.

3. Device Manipulation

- ioctl(): Control devices.

- read(), write(): Interact with hardware devices.

4. Information Maintenance

- getpid(): Get process ID.

- getuid(), setuid(): Get and set user ID.

- gettimeofday(): Get the system time.

5. Communication

- pipe(), shmget(), shmat(): Inter-process communication.

- socket(), bind(), connect(): Network communication.

System Call Interface in Endless OS

Endless OS, utilizing OSTree and Flatpak, emphasizes immutability and sandboxing. This affects system calls slightly:

- Applications run in sandboxed environments with limited access to system calls.
- Flatpak applications rely on portals for secure, user-mediated access to system resources.

Conclusion

Despite its user-friendly interface, Endless OS is built upon a robust Linux foundation. Its system call interface ensures efficient and secure system operation, leveraging the full power of the Linux kernel and POSIX compliance while enhancing usability and safety with modern technologies like Flatpak and OSTree.