Andre Hei Wang Law – 4017 5600

Laboratory #1: Answer of All Questions

What are the system calls to “open”, “read” and “close” a file?

Open: *int open(const char *pathname, int flags, mode_t mode);*

The ‘open’ system call, as its name suggests, opens a file. It first argument, the ‘pathname’, returns a file descriptor. The ‘flags’ argument specifies the mode in which the file should be opened (e.g., read-only, write-only, read-write, etc.). Lastly, the ‘mode’ argument specifies the permissions to be set if a new file is created.[1]

Read: *ssize_t read(int fd, void *buf, size_t count);*

The ‘read’ system call is used to read count bytes from a file from the file descriptor ‘fd’. It stores them in the memory location pointed to by argument ‘buf’ which represent the buffer. Then, the number of bytes read from the file is dictated by the ‘count’ argument. It should be noted that whenever the end of the file is reached, the final number of bytes accessed will be less than the argument ‘count’.[2]

Close: *int close(int fd);*

This last system call closes a file based on the argument ‘fd’ which is the file descriptor. After this system call, the file is closed and its file descriptor is invalid which means it can be reused by another file.[3]

Compare the system calls made by the C and JAVA programs. Why are the system calls so different if the two programs have the same functionality?

The system calls made by C and Java are different despite having the same functionalities. This is due to their differences in the underlying programming languages and the platform-independent nature of Java.

As a matter of fact, in C, it requires a direct system calls to the operating system to perform tasks such as reading and writing files. However, in Java, there are Java method (JVM – Java Virtual Machine) that performs that very same task. This is because Java provides higher-level abstractions and libraries that perform common tasks, and it is designed to be platform-independent.

What are some system calls made by the JAVA program that are not made by the C program and why might they be called?'

Examples of system calls made by Java program that are not made by the C program are methods related to the Java Virtual Machine (JVM). Mainly, they are methods to allocate memory, create threads, manage the execution of the program, etc.

These system calls made by Java programs are not made by C programs because C does not have the same level of abstraction or platform independence as Java. In C, the programmer must make direct system calls to allocate memory, create threads, and manage memory, which can be more complex and error-prone. On the other hand, the system calls made by Java programs through the JVM allow for more efficient and portable code, which is why they might be called.

References:

[1] https://man7.org/linux/man-pages/man2/open.2.html

[2] https://man7.org/linux/man-pages/man2/read.2.html

[3] https://man7.org/linux/man-pages/man2/close.2.html