Lab One

Joseph McDonough

Joseph.McDonough1@marist.edu

September 7, 2021

1 Problem One

What are the advantages and disadvantages of using the same system call interface for manipulating both files and devices?

A system-call interface posses a table that has indexes for each system-call. Therefore, the user does not need to know anything about the indexes or how the interface is implemented. This can serve as an advantage because files and devices can be manipulated from the same point of contact. It increases simplicity for the programmer as there is no different calls to be made, regardless of if it is a file or device in question. An additional advantage is that the system-call interface acts as a sort of buffer between the user mode and kernel mode. Giving direct access to kernel mode can be dangerous, so by having the user and the applications interact with an interface, it protects the kernel. The system-call interface introduces a safer way to expand on the functionality of an operating system.

One of the disadvantages to a system-call interface and its manipulation of both files and devices is that it handles two different things. Files and devices are not necessarily processed and manipulated in the same manner. Assuming that files are the simpler of the two to manipulate, the interface has to either add additional complexity to files to make up for what is needed for devices, or reduce the capabilities of devices to meet what is needed for files. Ultimately, having two different needs being met on one interface is not ideal.

2 Problem Two

Would it be possible for the user to develop a new command interpreter using the system call interface provide by the operating system? How?

The purpose of the command interpreter is to allow users to make requests and the system-call interface has functionality that would allow for that. Within the system-call interface, there are basic functions that allow for both process control and file and device manipulation. Here, there are services to allow for program control, status requests, and I/O request. As the system-call interface has a direct line into the kernel, it is possible to get and execute commands there. Through process control, scheduling can be done and programs can be ran when it is best. Additionally, as noted earlier, the system-call interface can manipulate files, therefore, developing a command interpreter using a system-call interface would allow for file manipulation. All the functionality that is required by the command interpreter can be serviced through the system-call interface.