Margaret Connor
CSS 430

1.19 What is the purpose of interrupts? How does an interrupt differ from a trap? Can traps be generated intentionally by a user program? If so, for what purpose?

Interrupts are triggered by either software or hardware. When the CPU receives and interruption it stops what it is doing an transfers execution to a provided location, after complete it resumes the previous computation. A trap is triggered only by software, traps are generated by a user program or an error. Traps and interrupts are a technique that allow multiple for asynchronous I/O.

1.23 Consider an SMP system similar to the one shown in Figure 1.6. Illustrate with an example how data residing in memory could in fact not have a different value in each of the local caches.

The MESI protocol is use to ensure that memory and local caches always have the same values. MESI is a Invalidation-based cache coherence protocol which means that all cache blocks have a four state finite state machine (M - modified, E - exclusive, S - shared, and I - invalid). A write may only occur when the cache line has a state of modified or exclusive, if it is in shared state all copies of the cache must be invalidated first. When a cache is in the modified state, it intercepts all reads to the main memory and returns its data. Depending on the data's state it will write back to main memory keeping the two updated with one another and change states appropriately.

2.6 What system calls have to be executed by a command interpreter or shell in order to start a new process?

A type of system call called a process control has to be executed by a command interpreter of shell in order to start a new process. Specifically, the fork() call in Unix and CreateProcess() call in Windows. After the process call is used the selected program is loaded into memory with the exec() system call and the program is executed.

2.21 What is the main advantage of the microkernel approach to system design? How do user programs and system services interact in a microkernel architecture? What are the disadvantages of using the microkernel approach?

The microkernel is a system design approach that moves all non essential part of the kernel onto the user-level as programs. Advantages of the microkernel is that once all non essential components have been removed it is easier to modify the operating

system and port between hardware designs. Since the microkernel is simpler and smaller any changes or alterations are easier to handle and produce less complexities. The microkernel also is more secure and reliable since most components are running as user processes, meaning failure will be contained from the rest of the operating system. The disadvantage of the microkernel is that the performance suffers due to system-function overhead. User programs and system services interact in a microkernel through message passing. The communicating processes exchange messages with one another to transfer information, in the microkernel architecture the program and services never directly interact with each other. Instead they pass indirectly through the microkernel.

2.22 What are the advantages of using loadable kernel modules?

Loadable kernel modules provide kernel with essential components and links additional components via modules. This allows for the kernel to provide essential services while the others are implemented dynamically as they are needed. The advantages of using loadable kernel model is that the core kernel does not need to be recompiled when a change is made and it also allow components to be called only when they need to used. This idea is similar to the microkernel in the fact that only the core components are included in the kernel, but is more efficient because modules do not use message passing.

Choose one System from the slides in Lecture One (Eniac, System 360, Multics, Android, etc..) and describe in detail (one paragraph) its importance to the evolution of Operating Systems

System 360 was a batch processing operating system created by IBM it was announced in 1964 and was the first operating system and mainframe computer series that could run the same programs across different computers. The system had a number of features that would go on to become industry standards, such as the Byte-addressable memory, 32-bit words, Commercial use of microcoded CPUs, and IBM Floating Point Architecture. The computers also supported multiprogramming, the process of switching to run another job when the current process was waiting for I/O operations to complete. This allowed users to make the most of their computer time, ensuring that the processor was always in use. These features would go on to pave the path to current systems making great advancements to computing at the time of its release.

Build and compile both a java and C++ application on a Linux system (do not turn in)

I was able to build and compile some Java and C++ applications on the UW Linux system. I feel more familiar with Java on the linux servers.