Hardware Interrupt Processing

Daniel R. Cender

Grand Canyon University: CST-221-O500

Feb 8, 2020

**Process Diagram**

A picture containing text, map

Description automatically generated

*Figure 1.* Process diagram of a hardware interrupt

**Description**

In interrupt driven I/O systems, hardware devices (like a mouse or a keyboard) generate actions that result in interrupts from the device controller. If the system employs a DMA, then the DMA collects “datums” (bits of data from device) into a buffer and only generates an interrupt once a sequence of input has been completed (Gottlieb, n.d.). The hardware then saves any currently used registers that haven’t already been backed up to memory. It then sets up a context, potentially including a TLB, MMU, and a page table. Then, once the CPU sets up a stack for the interrupt handler, it signals the interrupt controller that a handler was found for the interrupt. If no centralized controller is found, interrupts are enabled again.

This next bit actually includes running the handler and returning control to another process. The processor first copies the registers to the process table, then runs the interrupt-service procedure. This will grab information from the device controller’s registers. After it finishes, the processor decides on the next process to run. Some setup is run, similar to the TLB/MMU context setup from before. The CPU loads the process’s registers and PSW, then runs the process.

The huge disclaimer here is that these steps may run in different orders than specified, and the list of steps may differ depending on the processor’s architecture and manufacturer.

References

Gottlieb, A. (n.d.). Principles of I/O Software. Retrieved February 7, 2020, from https://cs.nyu.edu/courses/spring03/V22.0202-002/lecture-13.html

Tanenbaum, A.S. & Bos, H. (2015).*Modern Operating Systems.*Chapter 5.