CIS 751 Lecture Assignment 7

Chuck Zumbaugh

October 17, 2025

Exceptions in Linux are handled using the Interrupt Service Routine (ISR), which uses both a top half (the ISR handler) and a bottom half (more extensive tasks that are scheduled by the top half. Ex. workques). When an interrupt is triggered, the system choose one to execute (assuming they are not masked). The contents of the registers are saved and full control is given to the ISR handler. All other pending interrupts are masked during this time, and the interrupt handler is handled synchronously. Thus, the system waits for the top half, which is responsible for handling any immediate needs and scheduling the bottom half if needed. Once the ISR has finished, the registers are restored to their previous values, other interrupts are unmasked, and execution continues.

Because some handlers may need more extensive processing and time to complete, the bottom can be used for this purpose. Pending interrupts are not masked when the bottom half is run, and the system does not wait for the bottom half to finish (it is handled asynchronously). Long running functions, for example sleep, are expected to be completed here and normal execution has resumed when the bottom half is running..

The flow of interrupt handling is:

Interrupt occurs → Save registers and execute ISR handler → ISR handler completes quick work, such as scheduling bottom half → Registers restored and normal execution continues → Bottom half is run