Operating Systems Problem Set One

Carl Cortright

September 19, 2016

1 Traps

The trap instruction allows for what's called a trap interrupt. This instruction allows for a lot of things to happen, mainly centered around input and output operations. When a trap instruction is called, the operating system stops whatever it is doing, looks up the trap in the trap table, and calls the interupt handler associated with the specific trap code. Usually, that trap handler grabs some type of input from a peripheral device like a keyboard and notifies the relevant process that the interrupt has been raised and that there is new input to process.

The trap interrupt process happens in stark contrast with the old method of IO called polling. In polling what the running process does is wait for input in a while loop. In the case of something like user input from a keyboard, this is incredibly inefficient because a user types much slower than the clock speed of the CPU...

2 The Trap Table

When a interrupt is raised, the system indexes into the trap table to call the appropriate method associated with the trap. The trap table is a fixed peice of memory and usually exists at the bottom of the system memory near 0x00000000. Each entry in the trap table points to the appropriate interrupt handler routine.

3 Adding a new Device

The key to adding a new device without having to recompile the kernel is through Loadable Kernel Modules (LKMs) or more specifically, Reconfigurable Device Drivers. These drivers allow system administrators to add a device driver to the os without having to recompile the kernel. To create a LKM and load it in the kernel you need to do the following:

1. Write a C file with the following properties:

```
MODULE_AUTHOR();
MODULE_LICENSE(); // The licence must be open source

int init_module(); // Called when the module is loaded
void cleanup_module(); // Called when the module is unloaded
```

2. Compile the Kernel Module
3. Add it to the kernel

```
sudo insmod helloworld.ko
```

Step 3 is the key step. The insmod utility allows the admin to load the module in the kernel without recompiling. Basically, it automatically adds a new entry in the trap table and calls init module to load the module in to kernel without recompile.

This method does have some issues of course, like dealing with dependencies. Fortunately, there is another utility that automatically deals with those called modprobe.

When we want to unload the module, we can use the rmmod utility.

4 Interrupt Driven IO and DMA

There are two different types of IO besides polling, interrupt driven and direct memory access (DMA). Both of these methods capitalize on the fact that the CPU is much faster than most IO devices and can be doing other things while IO is being sent or received.

Interrupt Driven IO - This type of IO is characterized by CPU interrupts when a device is ready to perform IO. Basically when the device is ready to perform IO, the device sends an interrupt to the CPU. The CPU performs halts the running process, performs the IO, and resumes execution.

Direct Memory Access - Direct Memory Access or DMA is when there is special hardware support for devices to have direct access to the memory. In DMA when a device wants to perform IO it sends an interrupt to the CPU which in turn sends back the section of memory to be filled. Then, while the CPU is doing other stuff, the device itself directly loads or reads what it needs to from memory. This keeps the CPU free to run other processes while IO is happening.