

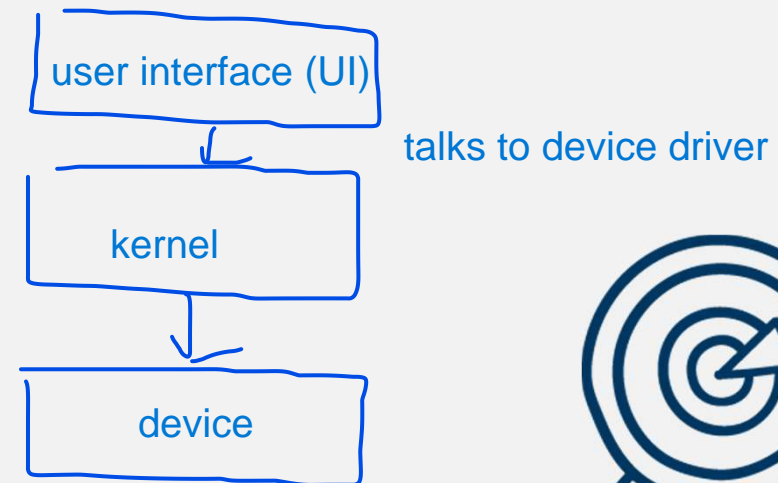
Guide to Operating Systems, 6th Edition

Module 6: Devices and Device Drivers

Learning Objectives

By the end of this module, you should be able to:

- Describe how OSs communicate with various categories of I/O devices
- Explain the need for device drivers
- Install device drivers
- Install printers



I/O Device Communication and Categories

- This module focuses on the input and output (I/O) tasks, which include:
 - Handling input from the keyboard, mouse, and other input devices
 - Handling output to the screen, printer, and other output devices
 - Controlling information storage and retrieval using storage devices like disk drives
 - Communicating with remote computers through a network

I/O Device Communication Strategies (1 of 5)

- A **register** is a storage location associated with the I/O device that data can be read from and written to
- The **I/O controller** reads commands written to the control register and performs the requested action controller controls which registers are there, what needs to be done (communication, buffer)
- The **control register** is where the device driver writes commands such as *read* or *write* for the I/O controller to act upon
- The **data register** is where the device driver writes data that the device outputs and reads data that the device inputs

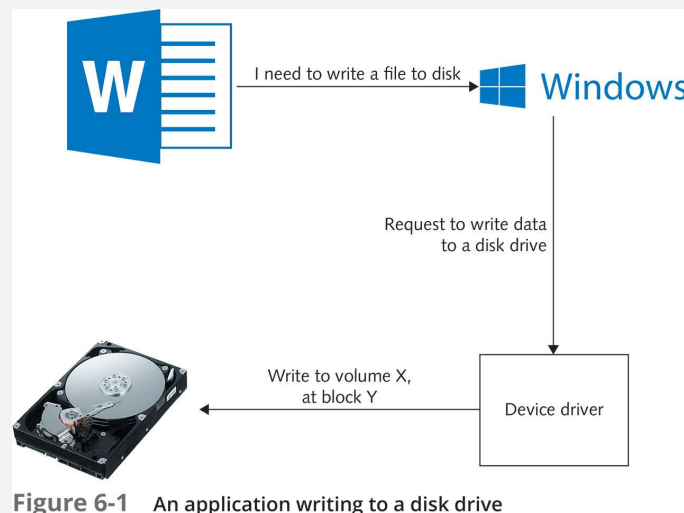
remember CPU registers are on the die

I/O Device Communication Strategies (2 of 5)

can it handle the next call?

- A **status register** might indicate to the device driver whether the device is ready to accept a command, that the device is busy, or that an interrupt has occurred
- Most control and data registers are a byte or a **word** in length
 - A word might be two or four bytes
- In some cases, the data registers might be supplemented by a **first-in first-out (FIFO) chip**, in which multiple bytes or words of input or output data are stored in the order they were received

I/O Device Communication Strategies (3 of 5)



I/O Device Communication Strategies (4 of 5)

- Port-Mapped Versus Memory-Mapped I/O
 - **Port-mapped I/O** uses dedicated memory addresses, referred to as ports, and special CPU instructions to communicate with I/O devices
 - With **memory-mapped I/O**, the control and data registers are mapped into the computer's main memory space and the CPU uses the same instructions for reading and writing I/O registers as it does for reading and writing RAM

I/O Device Communication Strategies (5 of 5)

- Programmed I/O Versus Direct Memory Access

CPU is taking control and handling stuff for the I/O device (lesser load like keyboards and mics)

- I/O devices that require the CPU to read and write each byte or word of data from and to the device registers are called **programmed I/O (PIO)** devices
- Modern computers use a process called **direct memory access (DMA)** to transfer large blocks of memory to and from I/O devices with little involvement by the CPU
 - A DMA controller transfers data between the I/O device and a block of memory, allowing the CPU to attend to other tasks

DMA kinda does the work and then sends it back to the CPU?

I/O Device Categories (1 of 3)

- Most I/O devices can be categorized based on the following:
 - How data is accessed (randomly or sequentially)
 - How much data is accessed – one byte per transfer or a block of data per transfer

I/O Device Categories (2 of 3)

- Random Access Versus Sequential Access
 - A **random-access device** allows data accesses to occur directly at any location of the device's storage, without having to start at the beginning each time a data access occurs **RAM**
 - **Sequential-access devices** include serial ports, network interfaces, and tape drives **Linked List (can't get to the end unless we go through each one)**
 - In order to access the data at location 100 on a sequential-access device, it must first pass through locations 1 through 99

I/O Device Categories (3 of 3)

- Character Versus Block Devices

- A device that transfers data ^{serial} **one byte at a time** is called a **character-stream device**
 - Keyboards, mice, serial ports, and sound cards are examples
- A **block device** works with a collection of bytes, usually a fixed size ^{hard drive}
 - Storage devices are the most common block devices
- A network interface is a special case
 - It most closely resembles a block device

Why an OS Needs Device Drivers (1 of 2)

lets us talk to our devices

- A device driver is software that enables the OS and application software to access specific computer hardware
- The OS provides basic I/O support for devices, but it doesn't support specific features of each device
OS manufacturers must give access so hardware manufacturers can write stuff for their dev drivers
 - For specific features to be supported and work properly, a device driver is needed for the device

Why an OS Needs Device Drivers (2 of 2)

- Advantages of using device drivers:
 - Only OS-specific functionality is built into the OS kernel for maximum performance
 - New devices can come on the market without requiring extensive updates to OSs
 - The number of I/O devices can expand to offer users a broad range of device selections and features
- You should use the manufacturer's driver instead of the one supplied with your OS
 - OS comes with bundled drivers
 - Manufacturers come out with new device drivers
 - Device drivers should match with manufacturer of the device rather than OS

Knowledge Check 1

- A device that transfers data one byte at a time is called which of the following?
 - A) random-access device
 - B) block device
 - C) sequential-access device
 - D) character-stream device

QUESTION



Knowledge Check 1: Answer

- A device that transfers data one byte at a time is called which of the following?

- D) character-stream device

ANSWER



Device Driver Installation (1 of 2)

- The procedure for installing drivers varies slightly based on the driver and the OS you are using
- If you download a new driver from a manufacturer's Web site, you may have to uncompress it
 - Microsoft includes built-in zip support
- Some drivers may come in executable file format (with an .exe file extension)
- Mac OS (since version 10.3 Tiger) has built-in zip support
- Linux users may retrieve drivers in a tar format

Device Driver Installation (2 of 2)

- Once the driver is located, you generally have three options for installation:
 - Your OS's installation utility
 - The Plug and Play (PnP) feature plug in and the device will be ready (already installed usually)
 - The installation utility provided by the hardware manufacturer
- Procedures differ among different OSs and with different equipment
 - The general process is very similar

Manufacturer Driver Installation

- When you use a hardware manufacturer's installation utility, the process is usually automated and well documented
 - Each manufacturer has a different procedure update driver in device manager?
- Generally, the procedure is to double-click an installation program or insert a DVD/CD-ROM into a drive
 - Wait for a program to start automatically or run a setup or install utility

Windows Driver Installation (1 of 11)

- The easiest way to install a driver is to use the PnP capability to automatically detect new hardware
- You can also use the Add a device wizard in Windows
- Other options are to use Device Manager to install a new driver, update an existing one, or roll back a driver to a previous version

Windows Driver Installation (2 of 11)

- Using PnP to Install a Device and Driver
 - Install any software drivers that are required
 - Connect the printer to the computer
 - Plug the printer into a power outlet and turn it on
 - Follow any configuration instructions
- Windows may try to find the built-in driver for the device on the Windows distribution disk or DVD/CD-ROM

Windows Driver Installation (3 of 11)

- Using a Windows Wizard to Install a Device or Driver
 - Use the Add a device wizard if:
 - Windows doesn't recognize newly installed hardware
 - You want to conduct an installation manually
 - In Windows 7 and later versions, open Control Panel and click Add a device under the Hardware and Sound category

Windows Driver Installation (4 of 11)

- Using Device Manager to Install or Update a Driver
 - You can also use Device Manager to:
 - Determine the location of device driver files
 - Check to make sure a device is working properly
 - Determine if there is a resource conflict for a device

Determine driver manufacturer

Windows Driver Installation (5 of 11)

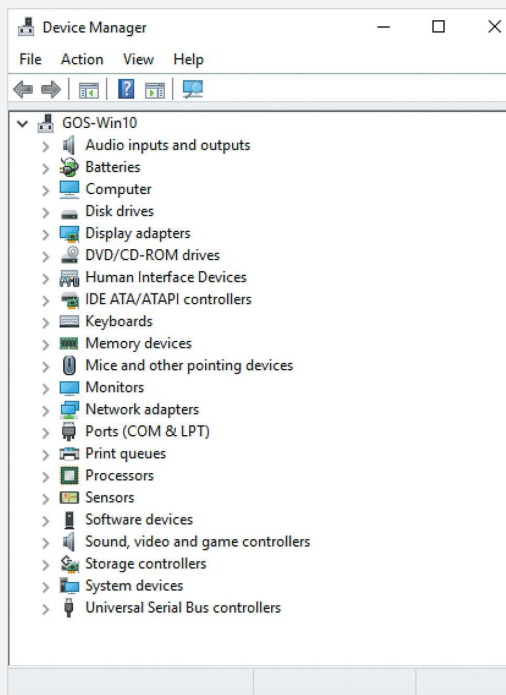


Figure 6-2 Device Manager in Windows 10

Windows Driver Installation (6 of 11)

- Figure 6-3 shows an example of a display driver in Windows 10 that is working properly

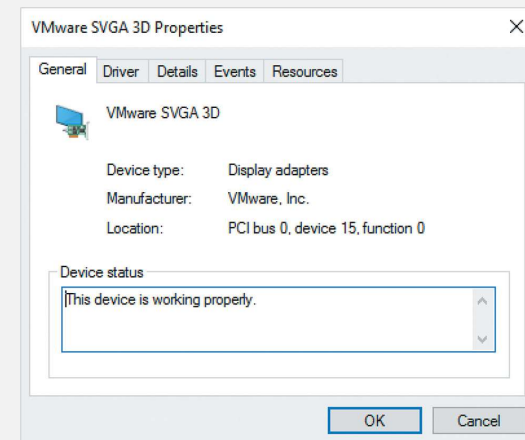
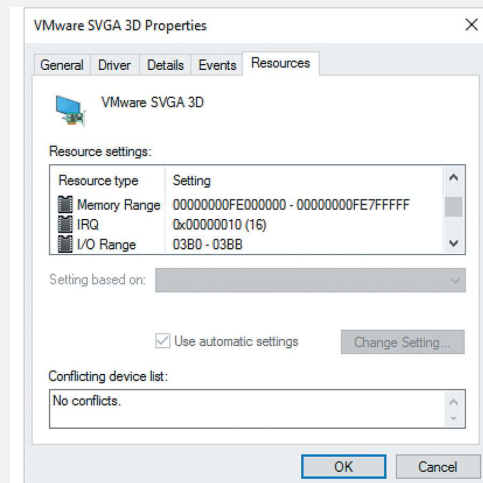


Figure 6-3 The device status of a display adapter in Device Manager

Windows Driver Installation (7 of 11)

- Using Device Manager to Install or Update a Driver (continued)
 - A conflict could occur if more than one device is assigned the same IRQ or I/O address
 - usually the OS assigns them as devices come in (provides address range for users to write their info to)
 - An **interrupt request (IRQ) line** is a channel within the computer that is used for communications with the CPU
 - An **I/O address range** is memory reserved for use by a particular device

Windows Driver Installation (8 of 11)



Notice an IRQ is assigned to it

Figure 6-4 Viewing the resources used by a device

Windows Driver Installation (9 of 11)

- Updating and Rolling Back Drivers
 - To update a driver from Device Manager, double-click the device and click the Driver tab
 - You can search your computer or the Internet for an updated driver
 - To roll back the driver to an earlier version, click the Roll Back Driver button

Windows Driver Installation (10 of 11)

if something goes wrong,
windows goes into safe
mode and turns off 3rd
party drivers, but you may
need to uninstall and roll
back a driver

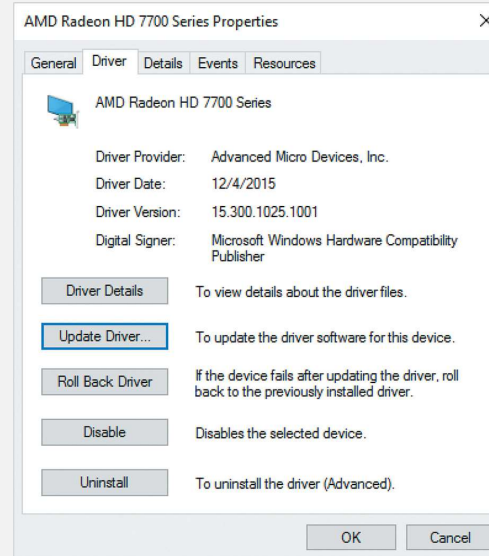


Figure 6-5 Updating or rolling back a driver

Windows Driver Installation (11 of 11)

- Configuring Driver Signing
 - When a new I/O device is installed under Windows, you have the option to make sure the driver has been verified by Microsoft
 - When it has been verified, a unique digital signature is incorporated into that driver (a process called **driver signing**)
 - In Windows 10, unsigned drivers are not permitted [legitimacy](#)
 - Using driver signing helps to ensure that the driver works properly with the device and in conjunction with other devices

[some manufacturers don't sign it so there must be some override](#)

Linux Driver Installation (1 of 3)

- The concept of drivers is a little different in Linux
- The central portion of the OS, the kernel, is where most of the device drivers are loaded
- Drivers are either in the form of:
 - Kernel modules – pieces of code that must be linked into the kernel
 - **Loadable modules** – pieces of code that are not linked into the kernel, but are loaded when the OS is started
- Device support in most UNIX/Linux versions is limited compared to other OSs

Linux Driver Installation (2 of 3)

- Linux devices are managed through the use of **device special files**, which contain information about I/O devices
- There are three types of device special files: block config (data blocks - large chunks of info)
 - **Block special files** are used to manage random access devices that involve handling blocks of data (hard drives, DVD/CD-ROM drives)
 - **Character special files** handle byte-by-byte streams of data (USB connections such as mice, keyboards, printers, etc....) similar to character-stream
 - **Named pipes** for handling internal communications, such as redirecting file output to a monitor outputting to the screen

everything's represented in the file system in Linux

Linux Driver Installation (3 of 3)

- Device special files are usually stored in the `/dev` directory
- If you need to create a device file for a new device, use the `mknod` command
- To view the I/O device special files on your system, use the `ls` command to see all of the files in the `/dev` folder

similar to device manager view

MacOS Driver Installation

similar to Windows

zip, unzip, executable

- Mac OS systems come with device drivers for most hardware
- When you obtain new hardware, follow these general steps:
 - Shut down the OS and turn off the computer
 - Attach the new hardware
 - Restart the computer and OS
 - Insert the DVD/CD-ROM for the hardware
 - Run the installer program on the DVD/CD-ROM for that hardware

Knowledge Check 2

- Files that are used for handling internal communications, such as redirecting file output, are known as which of the following?
 - A) block special files
 - B) named pipes
 - C) interrupt files
 - D) character special files

QUESTION



Knowledge Check 2: Answer

- Files that are used for handling internal communications, such as redirecting file output, are known as which of the following?

- B) named pipes

ANSWER



Installing Printers

different from other devices (hard-lined
or over a network, not really PnP)

- Most new printers come standard with a USB port, but many printers support wireless printing
- Many printers have a direct network connection option that lets you place the printer on a LAN
- It is more efficient to use a direct network-attached printer rather than a printer attached to a computer and configured for sharing on a network
 - The network interface is always on, making the printer always available to network users

Installing Windows Printers (1 of 2)

- In Windows, many printers can be installed by connecting the printer to the computer and letting PnP initiate the installation
- You can also connect the printer and use the Add a device wizard
- The best approach is to insert the DVD/CD-ROM that came with the printer and follow the instructions
- If you need to perform a manual installation or to initiate automatic detection and setup, use the Add a Printer option through Control Panel

Installing Windows Printers (2 of 2)

you can search for them, you can troubleshoot and find them
- maybe install manufacturer's driver for it

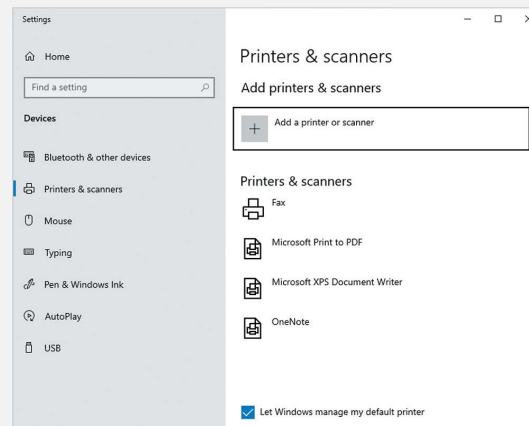


Figure 6-6 Adding a printer option in Windows 10

Installing Linux Printers (1 of 2)

- When a **print job** is sent from an application, a **print spooler** accepts the job and stores it in a **print queue** until it can be sent to a printer
- The most common Linux printing system is the Common Unix Printing System (CUPS)
- Most CUPS configuration files are in /etc/cups
- The CUPS daemon will usually detect directly connected printers and install them automatically
- For undetected printers, you can edit the printers.conf file directly, but it is better to run the Printers tool in Fedora Linux

Installing Linux Printers (2 of 2)

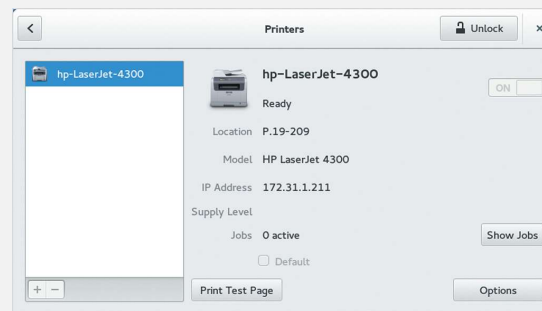


Figure 6-10 Adding a printer in Linux

Installing macOS Printers (1 of 2)

- In macOS, most printer drivers are already installed when you install the OS
 - If not, use the DVD that came with the printer
- To set up a printer, use the Printers & Scanners utility in System Preferences
- You can configure the following types of printers:
 - Default
 - IP (a network-attached printer that uses Internet Printing Protocol)
 - Windows

Installing macOS Printers (2 of 2)

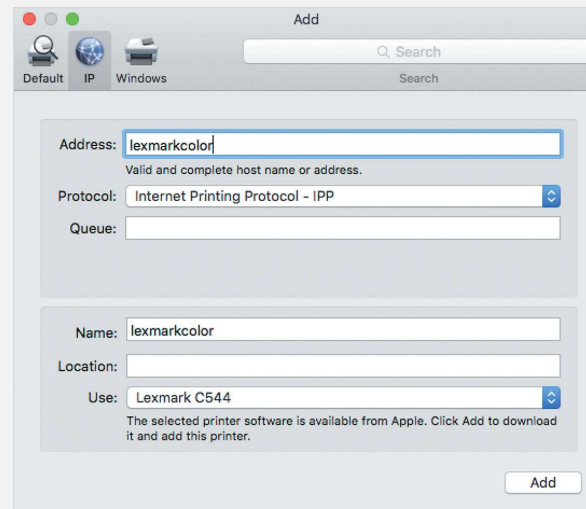


Figure 6-11 Adding a printer in macOS

Source: Apple Inc.

Summary (1 of 3)

- In modern multitasking systems, device drivers provide communication services between the physical device and the OS and applications
- Most I/O devices operate using three basic components: a controller, a control register, and one or more data registers
- I/O devices that require the CPU to read and write each byte or word of data from and to the device registers are called programmed I/O (PIO) devices
- Most devices can be categorized based on how data is accessed and how much data is accessed

more intense, smaller load
(mouse, etc.), not on hard drive



Summary (2 of 3)

- The OS provides generic drivers for your keyboard, mouse, and other standard devices, but it does not support unique features of individual devices
- The advantages of using device drivers are that only OS-specific functionality is built into the OS kernel, new devices can be easily added to a computer, and the number of I/O devices can expand in virtually unlimited directions
- The procedure for installing drivers varies slightly with the source of the driver and the OS you are using
- Printers are common output devices



Summary (3 of 3)

- Most OSs include tools for installing printers, such as the Add a device wizard in Windows, the Printers tool in Fedora Linux, and the Printers & Scanners utility in macOS

