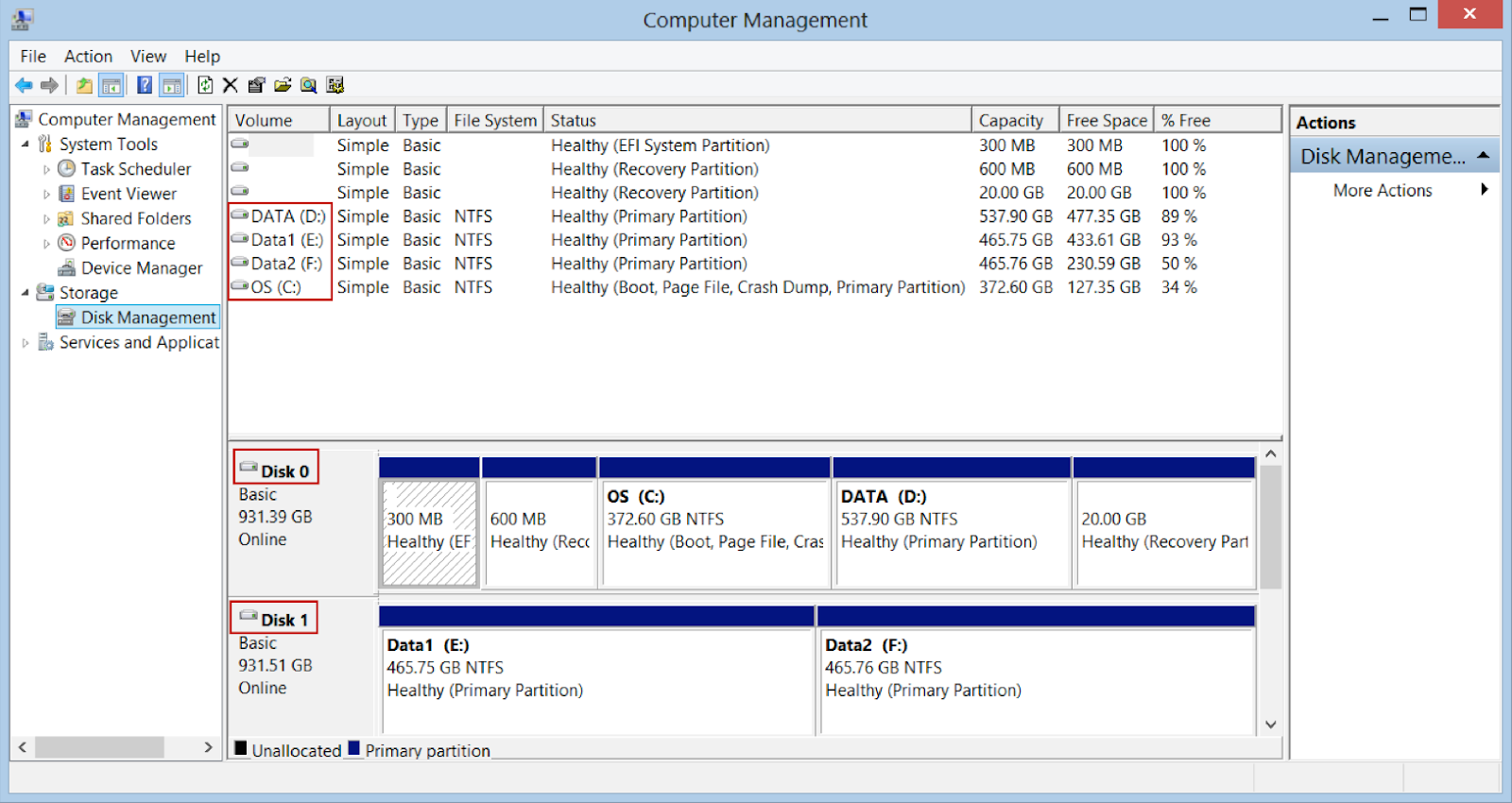
**19.2.6 Physical vs. Virtual Filesystems**

So far, the discussion regarding filesystems has been specific to physical filesystems. A physical filesystem is the database structure installed on a partition to organize files and directories.

Suppose an operating system consists of five partitions. In order to be able to access these partitions easily, their filesystems need to be merged together into a virtual structure.

On Microsoft Windows, this is accomplished by assigning drive letters to each partition. The first partition is typically referred to as the C: drive (which is poorly named as it is actually a partition, not an entire drive). Newer Microsoft Windows systems may come with multiple partitions, resulting in partitions such as D:, E:, and F:. Note that A: and B: were previously used for floppy drives and are not typically used for hard drive partitions.

To understand this better, consider the following diagram of the Disk Management component of Microsoft's Computer Management tool:

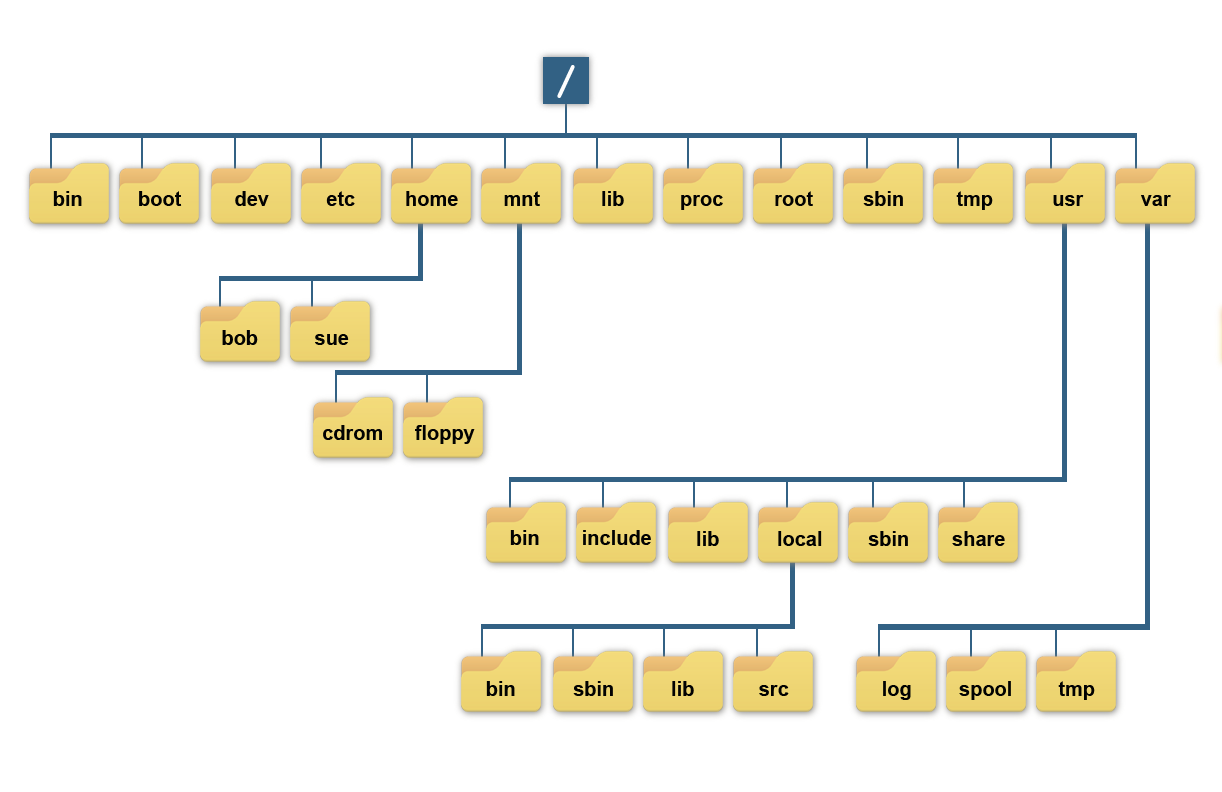


For this system, there are two hard drives named Disk 0 and Disk 1. Disk 0 has five partitions: three hidden, one assigned C: and another assigned D:. Disk 1 has two partitions one assigned E: and another assigned F:. Note that the tool also indicates that the type of filesystem that is installed on these partitions is NTFS.

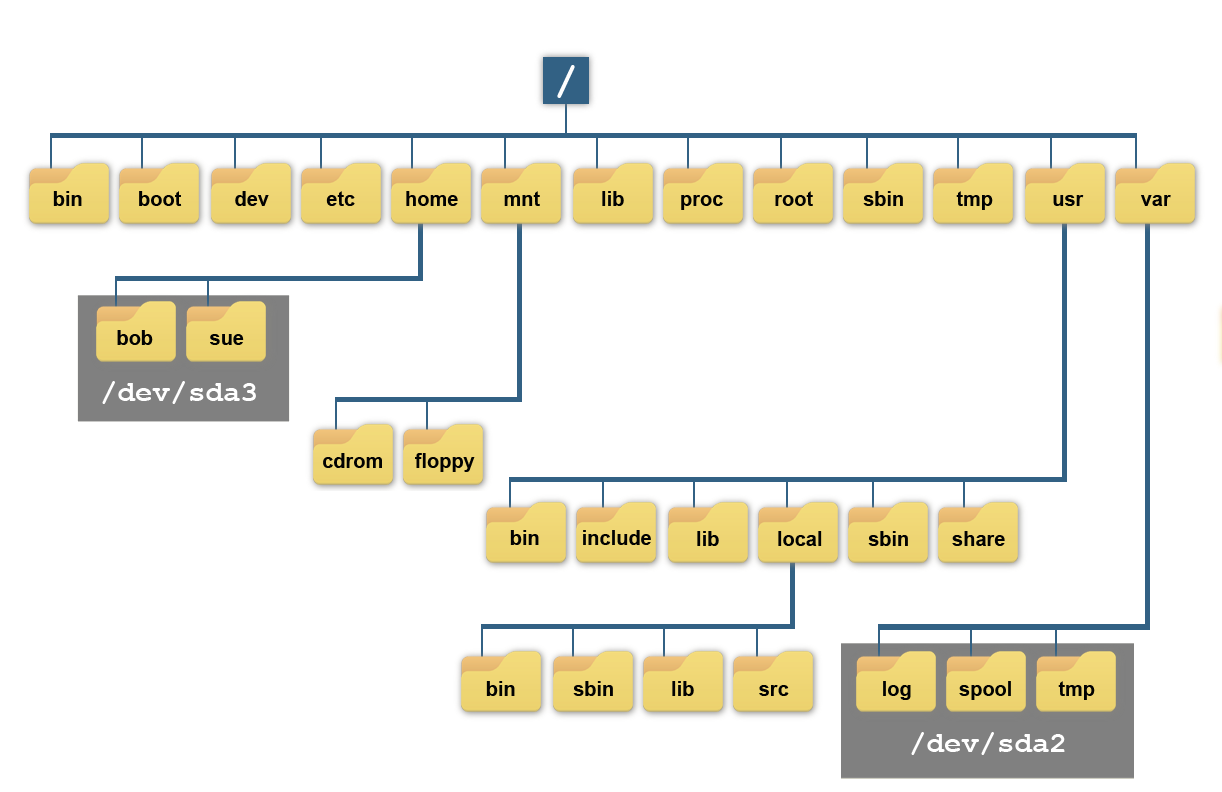
In Linux, there are no drive letters. Instead, each partition is assigned a device file name as mentioned previously:

| **Device Types** | **Partition Name** | **Example** |
| --- | --- | --- |
| SATA, SCSI, USB | /dev/sda\* | /dev/sda1  /dev/sda2 |
| PATA/IDE | /dev/hda\* | /dev/hda1  /dev/hda2 |

Users do not access the files that are stored on these partitions by directly using these device file names. After all, it's not convenient to remember what files are located on /dev/sda1 versus what files are stored on /dev/sda2. Instead, these device files are merged into the directory structure by *mounting* the partition's filesystem under a directory tree. To understand what this means, first consider the following diagram:



From a regular user point of view, all of the directories in the previous graphic appear to be from a single hard drive. However, these directories could represent a collection of separate filesystems merged together into a single directory structure, which is called the *virtual filesystem*. Consider the following diagram:



‌⁠​​⁠​

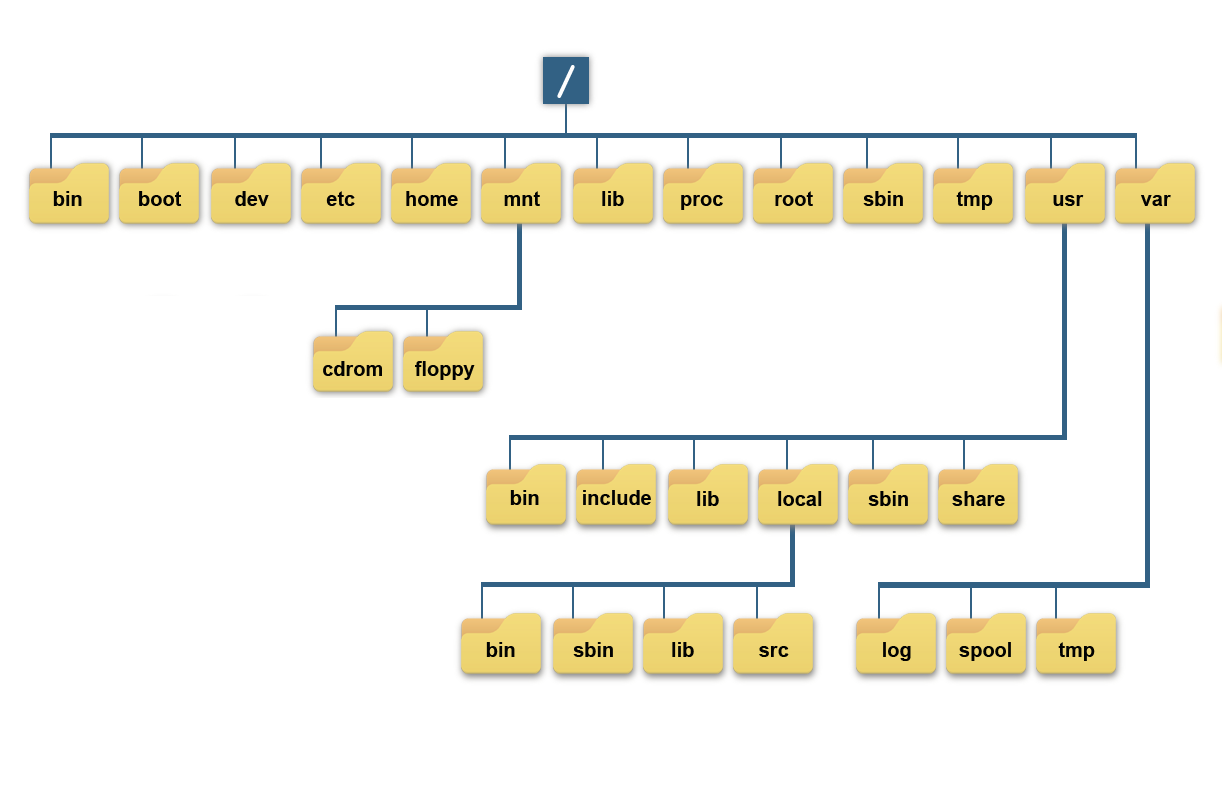
As demonstrated from the previous graphic, all of the files under the /home directory are really stored in the filesystem that is on the /dev/sda3 partition. All of the files under the /var directory are stored in the filesystem that is on the /dev/sda2 partition. This makes /home and /var mount point directories. The / directory is also a mount point directory, likely for the filesystem on the /dev/sda1 partition.

The process of placing a filesystem under a mount point is called *mounting*. This is accomplished either automatically at boot or manually with the mount command.

**Note**

*Mounting is covered in greater detail later in the course.*

The concepts of the virtual filesystem can be a bit difficult to comprehend at first, so consider this. If the administrator were to unmount the /dev/sda3 filesystem from the previous example, then the virtual filesystem would look like the following:



Notice that now there is nothing under the /home directory. This is normal because a mount point directory should be empty, as anything originally contained in a mount point directory will be replaced by the contents of the mounted partition's filesystem, and therefore be inaccessible. If the /dev/sda3 filesystem is mounted under the /home directory, then the bob and sue directories would again be available. Another way of thinking about it:

