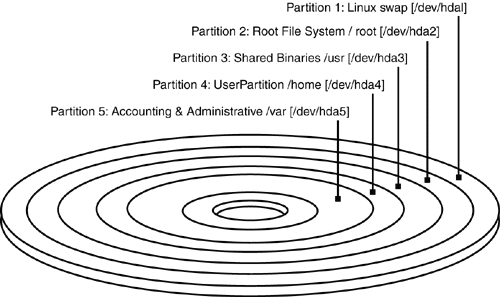
**Overview of file system**

A **filesystem** is the methods and data structures that an operating system uses to keep track of files on a disk or partition; that is, the way the files are organized on the disk, that helps in efficient storage and retrieval of data. The basic file system for UNIX is **UNIX File System, or UFS**, or sometimes called the Berkeley Fast File system. It introduced the concept of inodes which contain the metadata that describes a file



EACH PARTITION WILL CONTAIN FILES. EACH FILE HAS INODE INFORMATION WITH ATTRIBUTES AND BLOCK ADDRESSES TO LOCATE FILES.

EACH PARTITION WILL HAVE ITS OWN INODE BLOCK.

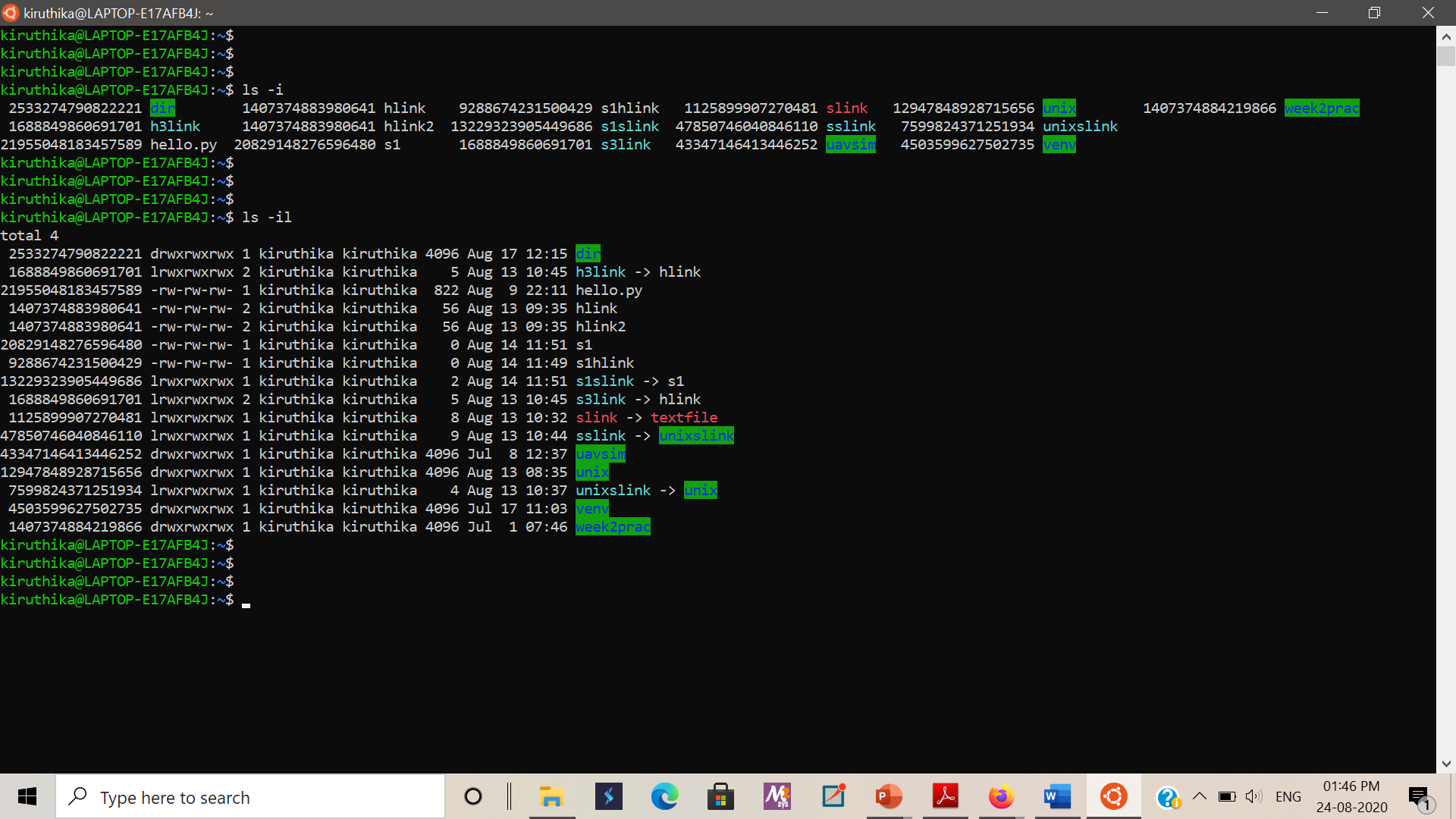
ALL THE INODE INFORMATION OF FILES PRESENT IN A PARTITION WILL BE AVAILABLE IN INODE BLOCK OF THAT PARTITION.

A Unix file is a store house of information. It is simply a sequence of bytes. Files are divided into these categories.

**File Types**

The UNIX filesystem contains several different types of files:

* Ordinary Files
  + Used to contain only data, source programs, Unix commands as well as any files created by user.
  + This is the type of file that you usually work with.
* Directories
  + Branching points in the hierarchical tree
  + Used to organize groups of files
  + May contain ordinary files, special files or other directories
  + Never contain "real" information which you would work with (such as text). Basically, just used for organizing files.
  + All files are descendants of the root directory, ( named / ) located at the top of the tree.
  + They usually map the filenames they contain with their respective inode number



* Symbolic link
  + A symbolic link is a reference to another file. This special file is stored as a textual representation of the referenced file's path (which means the destination may be a relative path, or may not exist at all).
  + A symbolic link is marked with an l (lower case L) as the first letter of the mode string, e.g.

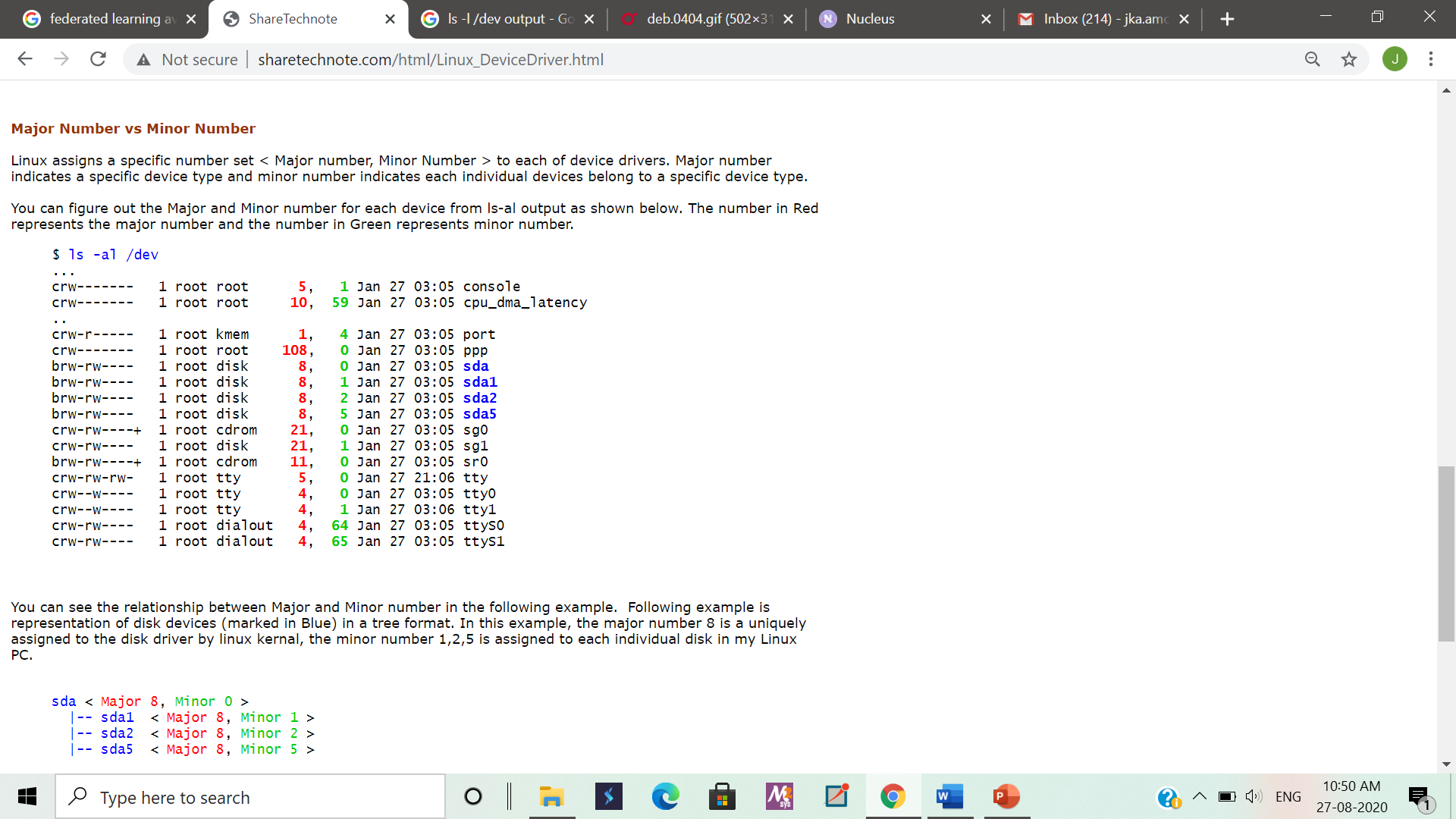
lrwxrwxrwx ... termcap -> /usr/share/misc/termcap

lrwxrwxrwx ... S03xinetd -> ../init.d/xinetd

* + Symbolic links are depicted in the above screenshot.
* Special Files / Device files
  + Device files allow user programs to access hardware devices on the system through the kernel.
  + you can read from them, write to them
  + When you access such a device "file," the kernel recognizes the I/O request and passes it to a **device driver** with some parameters, which performs some operation, such as reading data from a serial port, or sending data to a sound card.
    - The routines for a specific device are known as device drivers
  + Device files provide a convenient way to access system resources without requiring the applications programmer to know how the underlying device works. Under Linux, as with most Unix systems, device drivers themselves are part of the kernel.
  + Device files are located in the directory /dev on nearly all Unix-like systems. Each device on the system should have a corresponding entry in /dev.
    - For example, /dev/ttyS0 corresponds to the first serial port,; /dev/hda2 corresponds to the second partition on the first IDE drive.
    - In fact, there should be entries in /dev for devices you do not have. The device files are generally created during system installation and include every possible device driver. They don't necessarily correspond to the actual hardware on your system.
  + When using ls -l to list device files in /dev, you'll see something like the following:

brw-rw---- 1 root disk **3, 0** May 19 1994 /dev/hda

* + This is /dev/hda, which corresponds to the first IDE drive. First of all, note that the first letter of the permissions field is b, which means this is a block device file. (Recall that normal files have a - in this first column, directories a d, and so on.) Device files are denoted either by b, for block devices, or c, for character devices.
    - A block device is usually a peripheral such as a hard drive: data is read and written to the device as entire blocks (where the block size is determined by the device; it may not be 1024 bytes as we usually call "blocks" under Linux), and the device may be accessed randomly.
    - In contrast, character devices are usually read or written sequentially, and I/O may be done as single bytes. An example of a character device is a serial port.
  + Also, note that the size field in the ls -l listing is replaced by two numbers, separated by a comma. The first value is the **major device number** and the second is the **minor device number**. When a device file is accessed by a program, the kernel receives the I/O request in terms of the major and minor numbers of the device.
    - The major number generally specifies a particular driver within the kernel, and the minor number specifies a particular device handled by that driver.
    - For example, all serial port devices have the same major number, but different minor numbers. The kernel uses the major number to redirect an I/O request to the appropriate driver, and the driver uses the minor number to figure out which specific device to access



* + The great exception is network devices, which do not turn up in the file system (no device files) but are handled separately.

POSIX standard includes Sockets and Named Pipes as file types too.

**File Names**

UNIX permits file names to use most characters, but avoid spaces, tabs and characters that have a special meaning to the shell, such as:

**& ; ( ) | ? \ ' " ` [ ] { } < > $ - ! /**

Case Sensitivity: uppercase and lowercase are not the same! These are three different files:

**NOVEMBER November november**

Length: can be up to 256 characters

Extensions: may be used to identify types of files

**program.c  *- C language source file***

**alpha2.f  *- Fortran source file***

**xwd2ps.o  *- Object/executable code***

**mygames.Z  *- Compressed file***

Hidden Files: have names that begin with a dot (.) For example: ls -ali

**.cshrc .login .mailrc .mwmrc**

Uniqueness: as children in a family, no two files with the same parent directory can have the same name. Files located in separate directories can have identical names.

Reserved Filenames:

**/ *- the root directory (slash)***

**. *- current directory (period)***

**.. *- parent directory (double period)***

**~ *- your home directory (tilde)***