---------------Part 6 Industrial Workstation File System------------

\*The file system comprises of these components

--Namespace = a way to name things and organize them in a hierarchy

--API = a set of system calls for navigating and manipulating objects

--Security Model = a scheme for protecting, hiding, and sharing things

--Implementation = software to tie the logical model to the hardware

--Modern kernels define an abstract interface that accommodates many different back-end filesystems. Some portions of the file tree are handled by traditional disk based implementations. Others are fielded by separate drivers within the kernel. This is involved with remote filesystems as well.

--The architectural boundaries are not clearly drawn. Device files furnish a way for programs to communicate with drivers inside the kernel. They are not really data files, but they’re handled through the filesystem and their characteristics are stored on disk.

--Another complicating factor is that the kernel supports more than one type of disk-based filesystem. Ext3 and ext4 serve as Linux’s main filesystem implementations. FAT and NTFS file systems are used by windows.

\*Pathnames

--The filesystem provides a single unified hierarchy that starts at the directory / and continues down through an arbitrary number of subdirectories. / is also called the root directory. This hierarchy is different than windows as it retains the concept of partition specific namespaces.

--Pathnames can either be relative or absolute based upon using the current directory as the start or not.

--Spaces in filenames is allowed but not recommended as it Linux uses white space to separate the different commands. Quotes or backslashes are the best ways to deal with spaces in filenames.

\*Filesystem mounting and unmountnig

--File tree is this regard is the entire file structure while a file system is just a chunk of it.

--Most file systems are disk partitions or disk-based logical volumes, but they can be anything that obeys the proper API: network file servers, kernel components, memory based disk emulators…

--Filesystems are attached to the tree using the command ‘mount’ which maps a directory within the existing file tree, called the mount point to the root of the newly attached filesystem. Lists of the filesystems that are customarily mounted on a particular system is on /etc/fstab

--You detach file systems with the unmounts command, the file system must not currently be being used or unmounting will not occur. It is possible to do a “lazy” unmounts as well as a “forced” unmounts.

\*Things stored in the root directory

--/etc used for critical system and configuration files

--/sbin and /bin for important utilities

--tmp for temporary files

--/dev is a real directory but may be virtualized or overlaid with other file systems for device support

--/lib is used for storing shared libraries as well as the c preprocessor. Sometimes this is just a symbolic link to /usr/lib

--/usr is where most standard programs are kept along with various on-line manuals and libraries. For convenience, this directory is usually its own filesystem.

--/var houses spool directories, log files, accounting information, and various other items that grow or change rapidly and that vary on each host. It is a good idea to put /var on its own filesystem.

--The hier man page outlines some general guidelines for the layout of the filesystems

\*\*\*\*\*\*See page 147 for overview of all import directories\*\*\*\*\*\*

\*Linux provides 7 different types of files

--Regular files = consist of a series of bytes with no structure on their contents. Text files, data files, executable files. Both sequential and random accesses are allowed.

--Directories = contains named references to other files. You can create directories with mkdir and delete them with rmdir if they are empty. You can also use rm –r. The special entries ‘.’ and ‘..’ refer to the directory itself and to its parent directory. They cannot be removed. A file’s name is stored within its parent directory.

--Character device files

--Device files = lets programs communicate with the system’s hardware and peripherals. The kernel loads driver software for each of the systems devices. This software takes care of the messy details of managing each device so that the kernel proper can remain relatively abstract and hardware independent.

--Device drivers present a standard communication interface that looks like a regular file. When the filesystem gives a request that refers to a character or block device file, it simply passes the request to the appropriate device driver. Device files are just the rendezvous points that communicate with drivers.

--Character device files allow their associated drivers to do their own input and output buffering. Block device files are used by drivers that handle I/O in large chunks and want the kernel to perform buffering for them.

--Device files are characterized by two numbers, called the major and minor device numbers. The major device number tells the kernel which driver the file refers to and the minor device number tells the kernel which physical unit to address. Drivers can interpret the minor device number that is passed to them in whatever way they please.

--/dev was a generic directory and the device files within it were create with mknod and removed with rm. A script called MAKEDEV helped standardize the work of create device files.

--Block device files

\*\*\*Scattered throughout text above\*\*\*

--Local domain sockets = connections between processes that allow processes to communicate hygienically. Unix defines multiple sockets most of which are involved for the local network. Sockets cannot be read or written to by a process not involved in the connection. Socket system call creates and they can be removed by rm command or the unlink when they have no users.

--Named Pipes (FIFOS) = Named pipes allow communication between two processes running on the same host. They are known as “FIFO” files. You can create them with the mknod and remove them with rm. As with local domain sockets, real-world instances of named pipes are few and far between.

--Symbolic links

--Hard vs. Soft links