REAL TIME OPERATING SYSTEMS

Lesson-7:

File System Organization and Implementation

1. File

File

- **a named entity on a magnetic disc, or optical disc, or system memory** or memory stick
- **contains has the data, characters and text**

OSes differing abstractions of a file

- i) A *file* may be a named entity that is a structured record named entity as on a disk, which has having random access in the system.
- ii) May be a structured record on a RAM analogous to a disk and may also be either called separately called as ‘*RAM disk’* or simply ,simply, as a ‘file’ itself (virtual device).

OSes differing abstractions of a file

- (iii) May be an unstructured record of bits or bytes.
- (iv) A *file* device may be a pipe -like device

2. File organization in a file system

File organization in a system

- File is organized in a way according to a file system, which has set of command functions for operations on the file

Standard set of interfaces

- Should a file having integers differ from a file having bytes?
- Should a file having bytes differ from file having characters?
- Due to the differing approaches to device and file management interfaces, the development of a set of standard interfaces must.
- Then only then can systems can be portable.

Two type of file systems– Block File System

- Block File System. Its application generates records to be saved into the memory.
- First structured into a suitable format and then translated into block-streams.
- A file pointer (record) points to a block from the start to the end of the file.

Two type of file systems ─ Byte Stream File System

- Byte Stream File-System. Its application generates record stream.
- Stream are to be saved into the memory.
- These are first structured into a suitable format and then translated into byte-stream.
- A file pointer (byte index) points to a byte from the start index = 0 to N-1 in a file of N bytes

POSIX

- A standard set of interfaces.
- File operations are as the operations on a linear sequence of bytes

POSIX

- POSIX stands for Portable Operating System Interface IEEE standard for coding the programs when using the multiple threads.
- The X after I is because of the interfaces being similar to ones in UNIX like.
- It is according to the definitions at A T & T UNIX System V Interface.

Set of Command Functions in the POSIX File System

- open ( )
- write ( ) in the byte stream at the present pointer position
- read ( ) from the byte stream present pointer position
- lseek (List seek) or set the file pointer in the byte stream
- fcntl ( )
- close ( )

UNIX and LINUX File Systems

- Unix has a structured file system with an unstructured hardware interface.
- Linux supports different standard file-systems for the system

File Device

- File devices are block devices in UNIX.
- Linux permits the use of a block devices as a char devices also. This is because between *block-device* to *char-device, Linux has an additional* interface.
- Kernel interface is identical for the char and block devices in Linux and not in UNIX.

RAM disk

- File on the RAM that is hierarchically organized is is known as RAM disk.
- RAM memory storage is analogous to that on the disk and accessing is also analogous to a disk.
- For example, path for accessing a file is directory, then subdirectory, then folder and then subfolder. There is hierarchical tree like filing organisation

3. File descriptor

File descriptor— fd

- An integer, which returns on opening a file and which points to a data structure.

File describing data structure

- Identity
- Creator or Owner
- State
- Locks and Protection fields
- file Info
- Sharing Permission
- Count
- Storing Media Details

Summary

We learnt

- File Manager has functions to create, open, reads, seek a record, write, and close a file.
- A file has a file descriptor.

End of Lesson 7 of Chapter 8