

**+ UNIX:**

**- (UNICS): Uniplexed Information & Computing Services**

- DEC-PDP-7

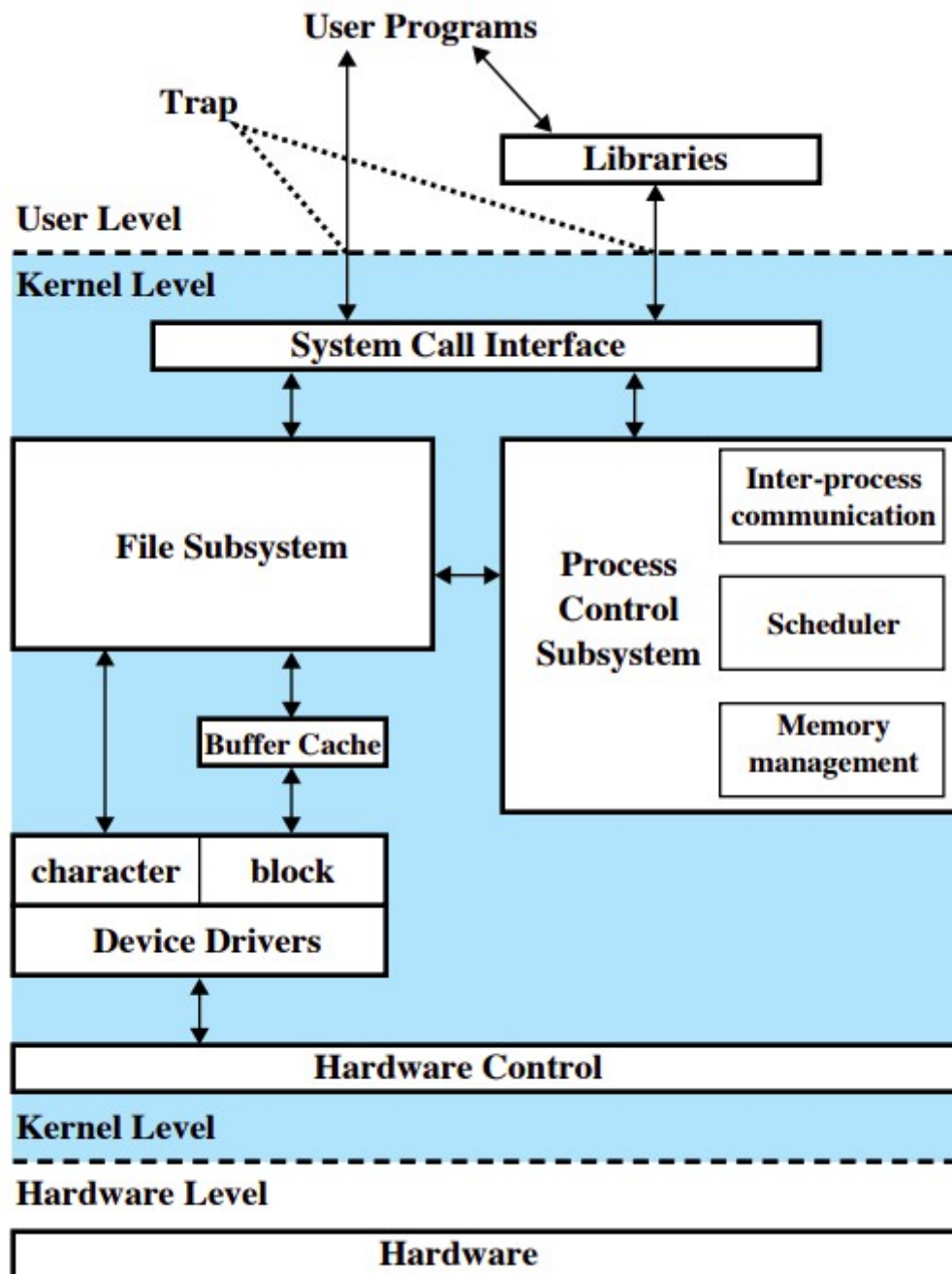
- UNIX was designed by the developers for the developers

- All modern OS's like Linux, Windows, MAC OS X etc..., follows system arch design of UNIX.

**- UNIX : "File has Space and Process has Life"**

- From UNIX point of view, whatever that can be stored is considered as a file, whereas whatever is an active state is considered as a process.

- UNIX treats all devices as a file



### # UNIX System Architecture Design:

- As system architecture design of UNIX is used/followed in all modern operating systems like Linux, MAC OS X, Windows etc... , so UNIX is also referred as mother of all modern operating systems.
- At **User Level** user programs or application programs may be there.

- At **Kernel Level** system contains major blocks like **System Call Interface, File Subsystem, Process Control Subsystem ( IPC, Memory Management, CPU Scheduling etc..), Buffer Cache, Device Driver, Hardware Control Interface/HAL i.e. Hardware Abstraction Layer.**
- For functioning of whole UNIX system all these subsystems works together.
- At **Hardware Level** contains actual hardware i.e. Processor, Memory Devices and I/O Devices.
- From UNIX point of view "**everything is a file**", and which ever that is not file considered as a "process".
- Whatever that can be stored, UNIX system consideres it as "file", and whatever which is active considered as a "process"
- "file has space and process has life".**
- "**buffer cache**" -- temp buffer maintained by the kernel in the main memory for storing most recently accessed disk contents to achieve maximum throughput in minimum h/w movement.
- **System Calls** - are the functions defined in C, C++ & Assembly language which provides interface of the services made available by the kernel for the user ( programmer user ).
- OR in other words, if user want to use services made available by the kernel, it can be directly called through system calls or indirectly through set of library functions which internally makes call to the system calls.
- In **UNIX** -- 64 system calls
- In **Linux** -- >300 system calls
- In **Windows** -- >3000 system calls
- system calls can be catagorised into 6 catagories:
- 1. fileio system calls:** open(), close(), read(), write(), lseek(), unlink(), rename(), etc...

**2. process control system calls:** fork(), exec(), wait(), \_exit(), etc....

**3. inter process communication system calls:** pipe(), shmget(), shmat(), shmdt(), shmctl(), msgget(), etc.....

**4. accounting information system calls:** getpid(), getppid(), stat() etc....

**5. protection & security system calls:** chmod(), chown(), etc...

**6. device manipulation system calls:** open(), read(), write(), close(), ioctl(), etc...

**+ Dual Mode Operation:**

- A modern computer operating system usually segregates virtual memory into "**kernel space**" and "**user space**".
- Primarily, this separation serves to provide memory protection and hardware protection from malicious or errant software behaviour.
- Kernel space is strictly reserved for running a privileged operating system kernel, kernel extensions, and most device drivers. In contrast, user space is the memory area where application software and some drivers execute.
- The unrestricted mode is often called "**kernel mode**", but many other designations exist (master mode, supervisor mode, privileged mode, etc.).
- Restricted modes are usually referred to as user modes, but are also known by many other names (slave mode, problem state, etc...)

**+ Kernel mode:** In kernel mode, the CPU may perform any operation allowed by its architecture; any instruction may be executed, any I/O operation initiated, any area of memory accessed, and so on.

- In the other CPU modes, certain restrictions on CPU operations are enforced by the hardware. Typically, certain instructions are not permitted (especially those –including I/O operations – that could alter the global state of the machine), some

memory areas cannot be accessed, etc. User-mode capabilities of the CPU are typically a subset of those available in kernel mode, but in some cases, such as hardware emulation of non-native architectures, they may be significantly different from those available in standard kernel mode.

#### **+ FileSubsystem:**

**- In UNIX/Linux 7 types of files are there:**

- 1. regular file (-):** text file, image file, audio file, source file, video file etc...
- 2. directory file (d)**
- 3. character special device file (c)**
- 4. block special device file (b)**
- 5. linkable file (l)**
- 6. pipe file (p)**
- 7. socket file (s)**

**- What is file?**

- file is a named collection of logically related data/information
- file is a container which contains logically related data/information
- file is a stream of bits/bytes
- **file = data + metadata**
  - data = file contents ( data exists inside the file)
  - metadata = information about the file (metadata exists inside FCB)

- When we create a new file, very first one structure gets created/assigned for that file in which all the information which is required to control file operations can be kept, this structure is referred as FCB (File Control Block).

- In UNIX , FCB is called as iNode.

**- FCB/iNode mainly contains:**

- iNode number - an unique identifier of a file onto the filesystem
- name of the file
- size of the file
- type of the file
- no. of data blocks allocated for the file
- time stamps: last accessed modified, creation time etc...
- access perms
- etc....

- In Linux "stat" command is used to display information about a file as well as filesystem.
- stat command internally makes call to stat() system call and it displays contents of an iNode.
- Per file one FCB/iNode gets created,
- No. of iNodes = No. of Files vice-versa

#### **- What is filesystem?**

It is a way to store data onto the disk in an organized manner so that data can be accessed efficiently and conveniently.

- filesystem gets stored onto the disk (any storage device), and onto the filesystem data ( data + metadata ) can be stored.
- filesystem is a logical.

#### **- Filesystem Structure:**

- Filesystem divides disk/partition logically into sectors

1. boot sector/boot block
2. volume control block/super block
3. master file table/inode list block
4. data block

