DATA STORAGE TECHNOLOGIES & NETWORKS

(CS C446, CS F446 & IS C446)

LECTURE 23– STORAGE

Logical Unit Number [LUN]

- Each LU has unique ID called a LUN
- LUNs hide the organization and composition of the RAID set from hosts
- Conventional storage provisioning method of creating LUNs is referred to as thick LUNs
  - To distinguish from LUNs created by virtual provisioning method
- If LUNs are assigned to a non-virtualized host, a bus scan is required to identify the LUN
  - LUN appears as a raw disk to the OS
  - Formatted with a file system and FS is mounted

LUN

- In virtualized host environment
  - LUN is assigned to the hypervisor which recognizes it as a raw disk
  - Virtual disks are files on the hypervisor file system
  - Virtual disks are then assigned to virtual machines and appear as raw disks to them
  - LUN space may be shared and accessed simultaneously by multiple virtual machines
- LUN
  - All disks – physical and virtual – that are visible outside the disk subsystem

LUN

- LUN expansion: MetaLUN
  - A method to expand LUNs that require additional capacity or performance
  - Can be created by combining 2 or more LUNs
  - Consists of a base LUN and one or more component LUNs
  - Can be either concatenated or striped

LUN

- Concatenated MetaLUN
  - Concatenated expansion adds additional capacity to the base LUN. Component LUNs need not have the same capacity as the base LUN
  - All the LUNs in concatenated metaLUN must be either protected (parity or mirror) or unprotected (RAID 0)
  - RAID type within a metaLUN can be mixed
    - RAID 1/0 LUN can be concatenated with a RAID 5 LUN
    - But RAID 0 LUN can be concatenated only with RAID 0 LUN
  - Concatenated expansion is quick – does not provide any performance benefit

LUN

- Striped MetaLUN
  - Restripes the base LUN's data across the base LUN and component LUNs
  - All LUNs must be of same capacity and RAID level
  - Provides improved performance due to the increased number of drives being striped
- All LUNS in both concatenated and striped expansion must reside on the same disk-drive type: either all Fibre channel or all ATA

Virtual Provisioning

- Called thin LUN
- Enables creation and presenting a LUN with more capacity than physically allocated to it
- Do not require physical storage to be completely allocated to them at the time they allocated and presented to a host
  - Physical storage is allocated on demand from the shared pool of physical capacity
  - Shared pool can be homogeneous (single drive type) or heterogeneous (mixed drive type – flash, FC, SAS…)
- Improves efficiency in allocation of storage to host
- Enables over subscription

LUN [Logical Unit Number] masking

- LUN masking
  - Is a process that provides data access control by providing which LUNs a host can access
  - Limits / authorize the access to the hard disks that the disk subsystem exports to the connected server
  - Reduces the configuration complications
  - Server sees the hard disks what it requires
  - Acts as a filter between the exported hard disks and accessing servers

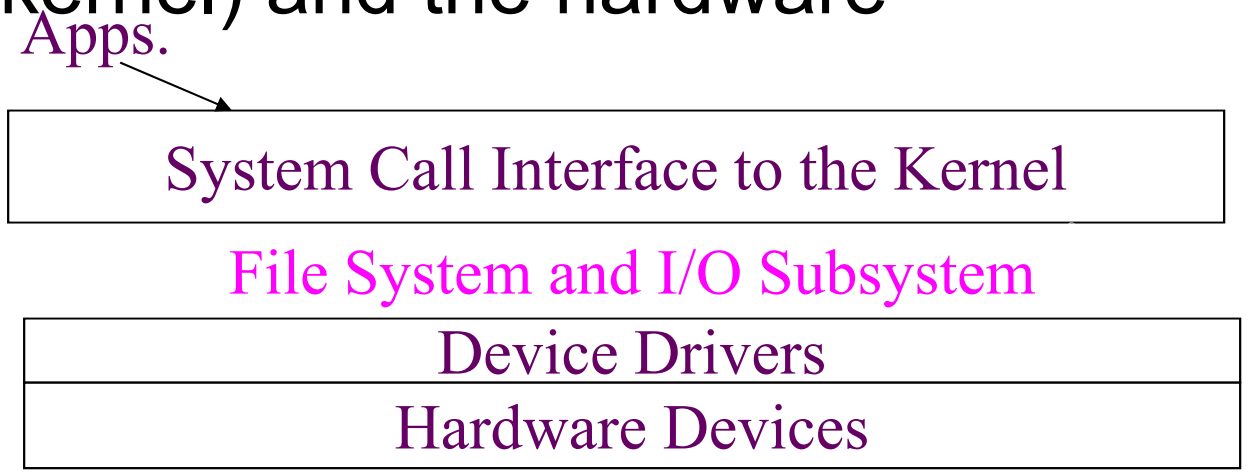
LUN for RAID

- For RAID
  - LUNs are slices of RAID sets which are spread across all the physical disks belonging to that set
  - Usable capacity of the physical volumes is determined by the RAID type of the RAID set
- Capacity of a LUN can be extended by aggregating other LUNs with it
  - The resultant LUN is known as meta LUN
- Mapping of LUNs to their physical location on HDDs is managed by the operating environment of an intelligent storage system

File System

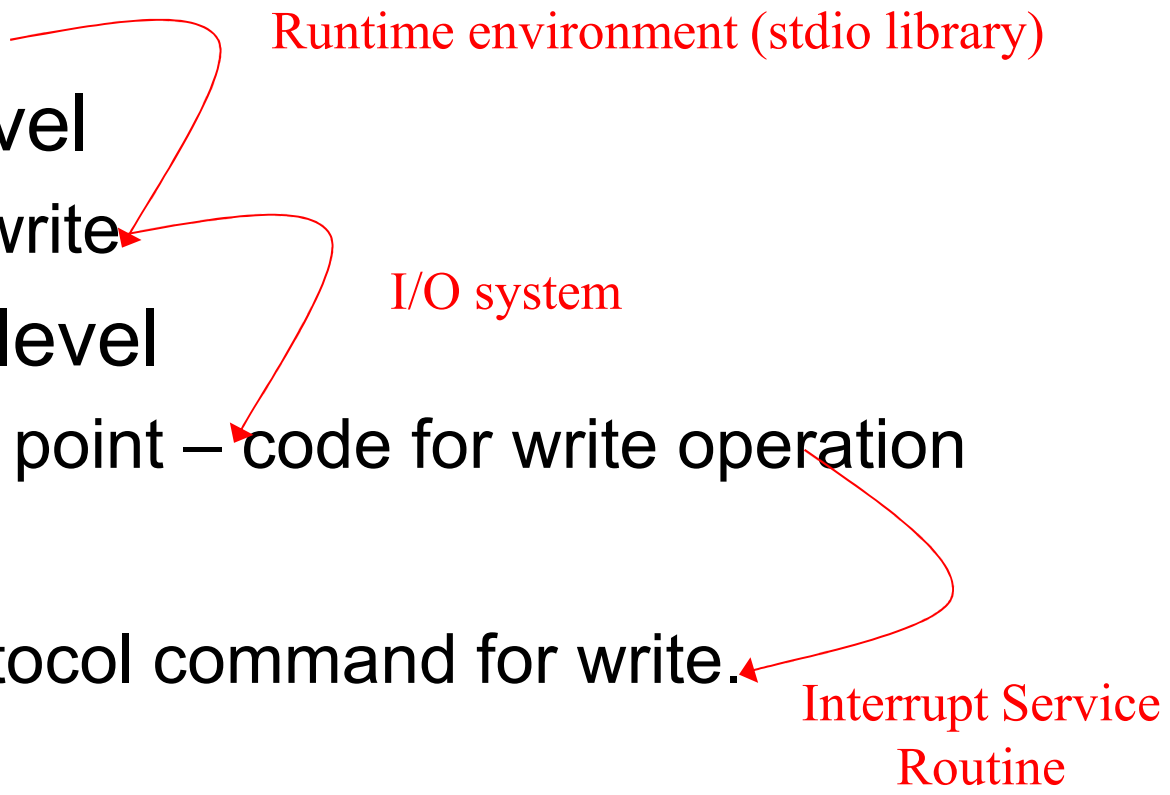
I/O in Unix - Basics

- System calls form the interface between applications and kernel
  - File System and I/O system are responsible for
    - Implementing system calls related to file management and handling input/output.
- Device drivers form the interface between the OS (kernel) and the hardware



I/O in Unix - Example

- Application level operation
  - E.g. printf call
- OS (kernel) level
  - System call bwrite
- Device Driver level
  - Strategy entry point – code for write operation
- Device Level
  - E.g. SCSI protocol command for write.



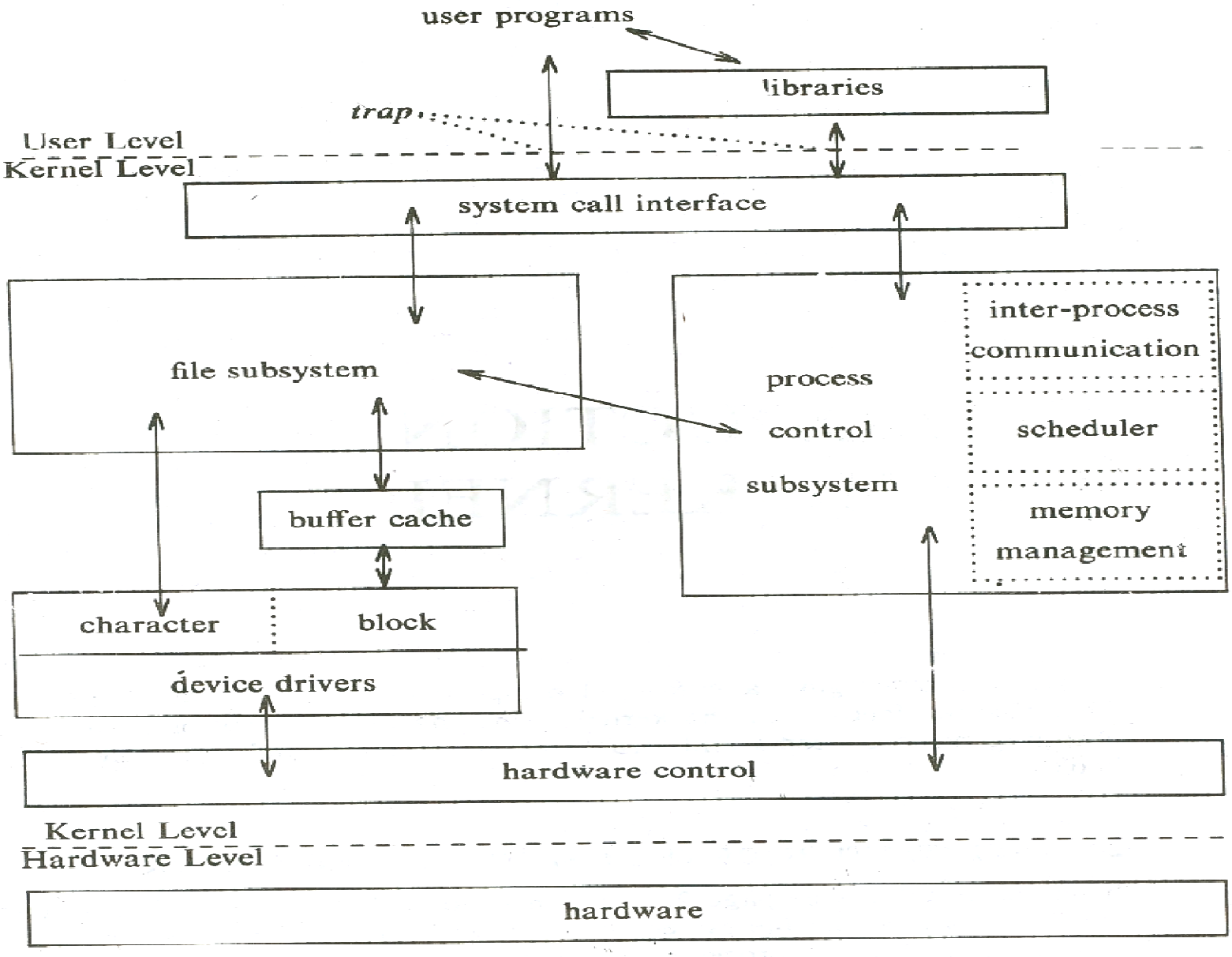


Figure 2.1. Block Diagram of the System Kernel