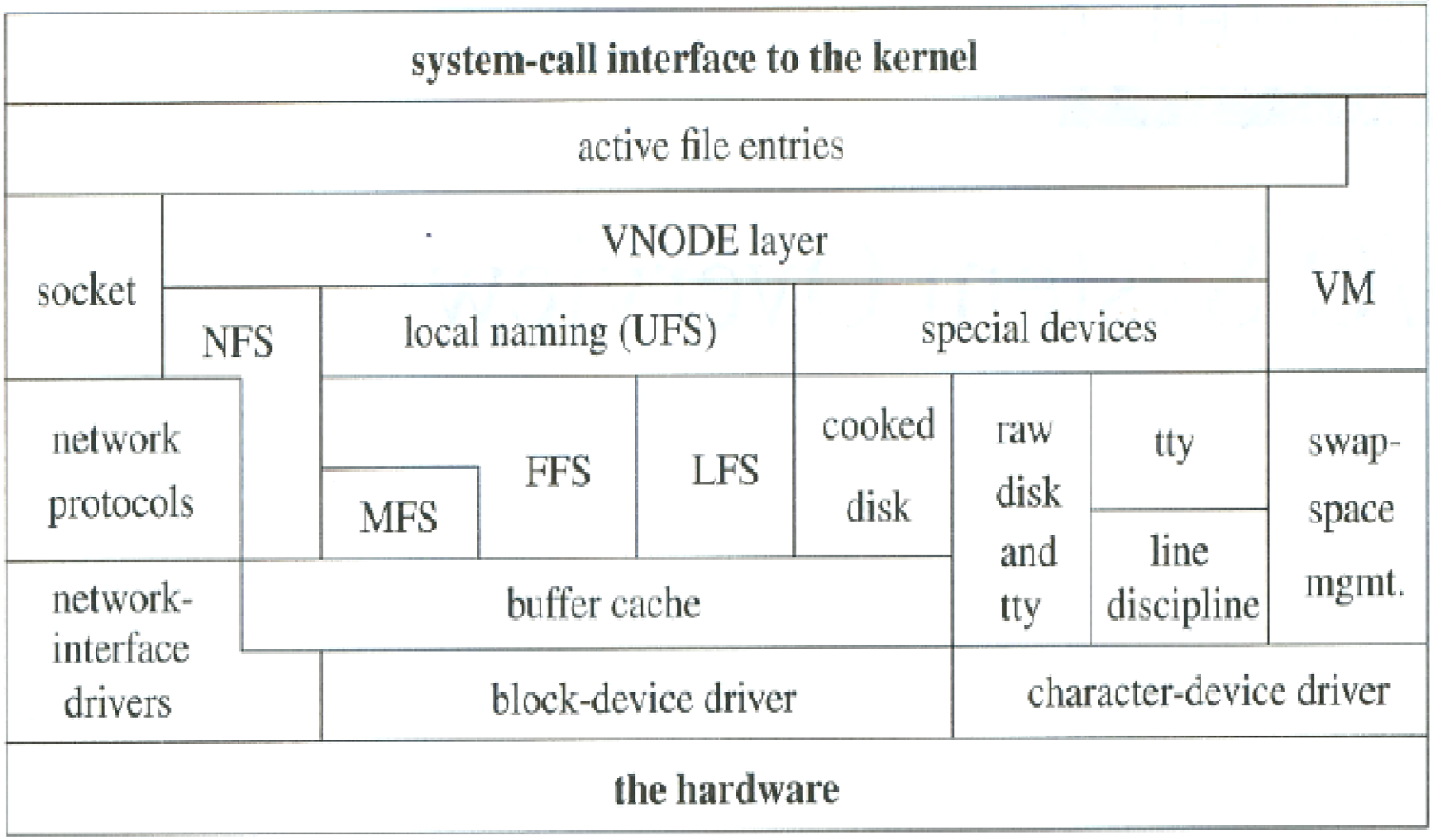
DATA STORAGE TECHNOLOGIES & NETWORKS

(CS C446, CS F446 & IS C446)

LECTURE 24– STORAGE

Kernel I/O structure



I/O in Unix - Devices

- I/O system used for
  - Network communication and Virtual Memory (Swap space) as well.
- Block Devices and Character Devices
  - Character devices – i.e. devices treated as byte streams
    - Include terminals, line printers
    - Unstructured or raw interface – no buffering by kernel
  - Block Devices – i.e. block-addressable devices
    - E.g. Disks and Tapes
    - Structured interface – block-sized accesses – buffering done by kernel.

I/O in Unix – Devices

Device Identification – major device #, minor device #

- Major Device # refers to the type of device and hence is used to identify a device driver.
  - When a device file is opened, Linux examines its major number and forwards the call to the driver registered for that device.
- Minor Device # refers to a specific device and is interpreted by a device driver.
  - Minor number is used to identify the specific device instance if the driver controls more than one device of a type.

- Block Device table and Character Device Table

I/O in Unix – Device Drivers

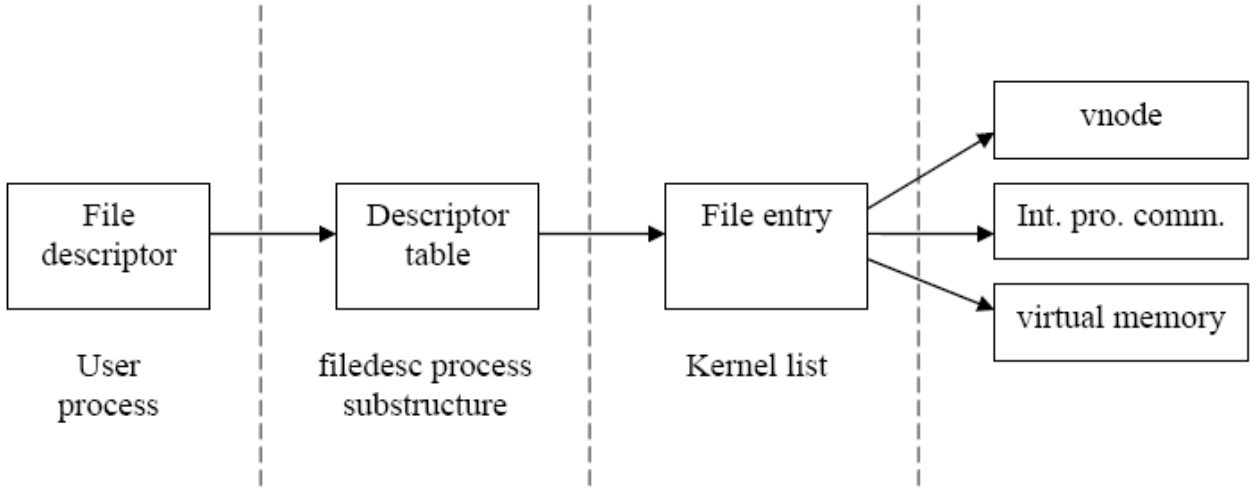
- Device Driver Sections
  - Auto-configuration and Initialization Routines
    - Probe a device for presence
    - Initialize device and associated software
  - Routines for servicing I/O requests
    - Invoked because of system calls or VM requests
    - Synchronous execution; may block
    - Referred to as the “top-half” of the driver
  - Interrupt service routines
    - Invoked by interrupt from a device
      - Cannot depend on per-process state
      - Cannot block
    - Referred to as the “bottom-half” of the driver
  - (Optional) Crash-Dump routine

Open Files

- Several pieces of data are needed to manage open files:
  - File pointer: pointer to last read/write location, per process that has the file open
  - File-open count: counter of number of times a file is open – to allow removal of data from open-file table when last processes closes it
  - Disk location of the file: cache of data access information
  - Access rights: per-process access mode information

Descriptor management and services

- The file descriptor is used by kernel to index into descriptor table for the current process to locate file entry or file structure
- file entry provides a file type and a pointer to an underlying object for the descriptor



Open File Entries

- Contain information necessary to access the underlying objects and maintains common information
- Each entry contains a type and an array of function pointers
  - Function pointers translate generic operation of file descriptors into the specific action associated with their type
  - BSD 4.4 support 2 descriptor types (files, sockets)
  - Operations
    - Read from the descriptor
    - Write to the descriptor
    - Select on the descriptor
    - Do ioctl operations on the descriptor
    - Close and possibly deallocate the objects associated with the descriptor

Descriptor

- Changes to a descriptor (fcntl system call)
  - Duplicate a descriptor (dup)
  - Get or set the close-on-exec flag
  - Set descriptor into nonblocking mode
  - Force all write to append
  - Send a signal to process when it is possible to do I/O
  - Send a signal to process when exception condition arises
  - Set or get PID or process – group ID
  - Test or change the status of a lock on a range of bytes within an underlying file