

# Disk I/O

## Major/Minor Device Numbers

All special files have a *major device number* and a *minor device number*.

- Major Device Number—specifies the device class, floppy disk, hard disk, terminal (selects device driver)
- Minor Device Number—specifies specific device within the class. Is passed as a parameter to the device driver. Which disk, which terminal.

Look at specific device drivers in some detail.

## RAM Disks

Store blocks of data in preallocated memory rather than on physical disk. Why?

- fast access
- always available
- volatile (goes away)

## RAM Disk Driver

Map kernel data to file system. Example is `/proc` in Linux, which allows access of process information.

Also Unix RAM disk devices:

- 0: `/dev/mem`—read memory contents
- 1: `/dev/kmem`—read kernel memory contents (used by system commands). One question is who can look at the memory contents. Varies between machines.
- 2: `/dev/null`—a bottomless pit, accepts input and throws it away. `cat /etc/motd > /dev/null`.

## Setup of CCC machine

```
< wpi /usr9/cew 1 >ls -lg /dev/kmem
cr--r-----  1 root    news      2,  1 Jul 14  1997 /dev/kmem
```

```
< wpi /usr9/cew 2 >ls -lg /dev/mem
cr--r-----  1 root    news      2,  0 Jul 17  1998 /dev/mem
```

```
< wpi /usr9/cew 3 >ls -lg /dev/null
crw-rw-rw-   1 root    wheel     2,  2 Feb 22 13:17 /dev/null
```

```
< wpi /usr9/cew 4 >ls -lg /bin/ps
-rws--x--x   1 root    tty       73728 Dec 29  1997 /bin/ps*
```

```
< wpi /usr9/cew 5 >ls -lg /usr/ucb/uptime
-rws--x--x   2 root    tty       278528 Dec 29  1997 /usr/ucb/uptime*
```

# Disks

Look at hardware picture for a hard disk drive. Disk organization:

- organized into *cylinders*
- each level of the cylinder is a *track*
- tracks are divided into *sectors* More space for sectors towards outer rim.
- Newer disks are divided into *zones* with more sectors on outer
- magnetic storage devices (*read-write head* changes or senses the magnetic coating)
- *arm* moves the read-write head(s)
- disk heads do not normally touch the surface. Mechanical shock or dust particles can cause contact and a *head crash*.

Disk controller hides details of disk geometry from operating system by presenting view of  $x$  cylinders,  $y$  heads (tracks) and  $z$  sectors per track.

Look at Fig 5-17 for comparison of disk parameters.

Why use disk storage?

- storage capacity is larger
- disk storage is less expensive
- it is permanent, long-term storage (nonvolatile)

## Disk Head Scheduling

Disk access time = seek time + latency time (rotational delay) + transmission time.

Typical delays:

- seek time:  $m \cdot n + s$  when there are  $n$  tracks,  $m$  is a constant (0.3 msec on PC disk, 0.1 msec on larger disks) and  $s$  is start-up time (20 msec on PC disk, 3 msec on larger disk).
- rotational delay: most disks: 3600 rpm for a revolution of 16.7 msec, hence an average of 8.3 msec. Floppy disks rotate 300-600 rpm for an average delay of 100-200 msec.

Want to minimize disk access time. One technique is interleaving so the next sector will be available for transfer when the previous is complete.

See Fig 5-26.

Because the CPU can generate disk requests much faster than the hardware can service them, several requests may be queued at any given time. The use of *disk-head scheduling policies* can improve performance by minimizing the seek time.

## FCFS

A first come first served (FCFS) policy is the simplest to implement:

- it is fair
- results in a seek for almost every operation
- works well under light load, but saturates quickly under heavy load

## SSF

Alternatively, we can use the shortest seek latency first (SSF) policy:

- of all outstanding requests, schedule the one requiring the shortest seek
- while minimizing seek time, it is unfair
- leads to high delay variance

## Elevator (Scan)

Look at elevators in a tall building. Same problem.

As a compromise, the *elevator* policy sweeps the head from one side of the disk to the other, servicing requests as the head passes over the relevant cylinder:

- the delay is bounded to be no longer than two sweeps
- optimization: sweep only as far as needed, reversing direction in the absence of further requests in the current direction
- optimization: (C-Scan) sweep in only one direction, to reduce variance

## Disk Cache

Disk driver or controller may have a track cache to speed up access.

Independently disk driver maintains a cache of disk blocks in memory.

## Error Handling

Errors and what to do:

- programming error (request for nonexistent sector)—trap and return an error
- transient checksum error (dust on head)—retry
- permanent checksum error (disk block physically damaged)—include block as part of a special “damaged” block file that is never accessed. If no reads of this special file the block is never accessed. Backup could be done a track at a time, which causes problems. Also maintain a few free tracks to substitute for tracks with bad sectors.
- seek error (arm in wrong place)—recalibrate the arm
- controller error—must reset itself