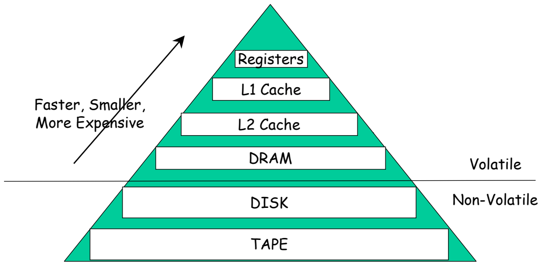
10: Storage and File System Basics

Last Modified:
10/8/2002 8:39:17 PM

Storage Hierarchy



Faster, Smaller,
More Expensive

Registers

L1 Cache

L2 Cache

DRAM

DISK

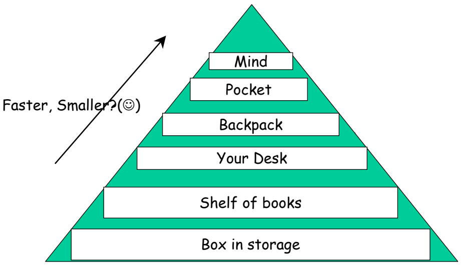
TAPE

Volatile

Non-Volatile

Each level acts as a cache of lower levels

Example



Faster, Smaller?(☺)

Mind

Pocket

Backpack

Your Desk

Shelf of books

Box in storage

Each level acts as a cache of lower levels

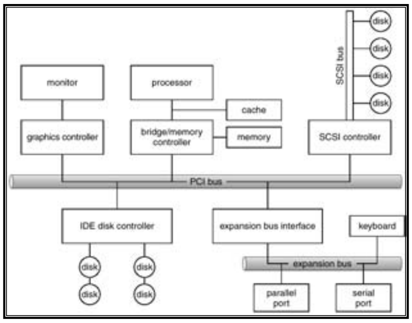
Secondary Storage

- "Secondary" because unlike primary memory does not permit direct execution of instructions or data retrieval via load/store instructions
- Usually means hard disks
- Tends to be larger, cheaper and slower than primary memory
- Persistent/Non-volatile
  - Like "durability" for transactions

Tertiary Storage Devices

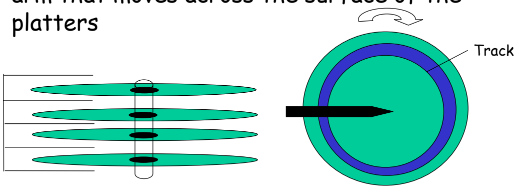
- Used primarily as backup and archival storage
- Low cost is the defining characteristic
- Often consists of *removable media*
  - Common examples of removable media are CD-ROMs, tapes, etc.
- As disks get cheaper and cheaper, duplicating data on multiple disks becomes more and more attractive as a backup strategy

Typical PC



Disk Basics

- ❒ Disk drives contain metallic platters spinning around a central spindle
- ❒ Read/write head assembly is mounted on an arm that moves across the surface of the platters



Terms

- ❒ Track = one ring around the surface of one of the platters
- ❒ Sector = one piece of a track (usually 512 bytes); More sectors in outer tracks
- ❒ Cylinder = all tracks at the same distance from the center of the platters (I.e. all tracks readable without moving the disk arm)

Disk Addressing

- ❒ Early disks were addressed with cylinder #, surface # and sector #
- ❒ Today disks hide information about their geometry
  - ❍ Disks export a logical array of blocks
  - ❍ Disk itself maps from logical block address (LBA) to cylinder/surface/sector
  - ❍ Allows disk to remap bad sectors (when formatted disk reserves some sectors to use as replacements)
  - ❍ Allows disk to hide the non-uniformity of the storage
    - More data on outer tracks, etc.
- ❒ Disks also have internal caches so that not all requests go to the media
  - ❍ On reads take advantage of multiple accesses to the same track
  - ❍ On writes, say write is "done" when it is memory inside the disk

Disk Formatting

- ❒ Low-level formatting involves dividing the magnetic media into sectors
  - ❍ Each sector actually consists of a header, data and a trailer
  - ❍ Header and trailer contain information like sector number and error correcting codes (ECC)
  - ❍ ECC is additional redundant bits that can often correct for bit errors in the stored value
- ❒ OS also formats drive
  - ❍ 1st divides into partitions – each partition can be treated as a logically separate drive
  - ❍ 2cd file system formatting of partitions (more on that later)

Disk Interfaces

- ❒ Interface to the disk
  - ❍ Request specified with LBA and length
  - ❍ Request placed on bus, later reply placed on bus
- ❒ Device driver hide these details
  - ❍ Provide abstraction of synchronous disk read
- ❒ OS use the disk to provide services
  - ❍ Virtual memory
- ❒ OS exports higher level abstractions
  - ❍ File systems
- ❒ Some applications use the device driver interface to build abstractions of their own (get their own partition)
  - ❍ Database systems

Disk Performance

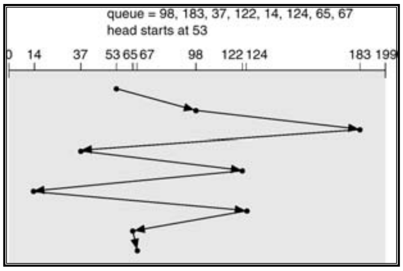
- ❒ Divide the time for an access into stages
  - ❍ Seek time – time to move the disk arm to the correct cylinder
    - How fast can mechanical arm move? Improves some with smaller disks but not much
  - ❍ Rotational delay – time waiting for the correct sector to rotate under the read/write head
    - How fast can spindle turn? RPMs go up but slowly
  - ❍ Transfer time – once head is over the right spot how long to transfer all the data
    - Larger for larger transfers
    - Rate determined by RPMs and by density of the bits on the disk (density going up very quickly!)
- ❒ Getting good performance from a drive (seeing impact of a "faster" drive" means avoiding seek and rotational delay)

Avoiding Seek and Rotational Delay

- To take advantage of higher transfer rate, OS must transfer larger and larger chunks of data at a time and avoid seek and rotational delay
  - Size and placement of virtual memory pages?
  - Size and placement of FS blocks?
- OS tries to avoid seek and rotational delay by placing things on disk together that will be accessed together
- Can also avoid seek and rotational delay by queuing up multiple disk requests and servicing them in an order that minimizes head movement (disk scheduling)
  - Like with CPU scheduling, there are many disk scheduling algorithms

First Come First Serve (FCFS)

Illustration shows total head movement of 640 cylinders

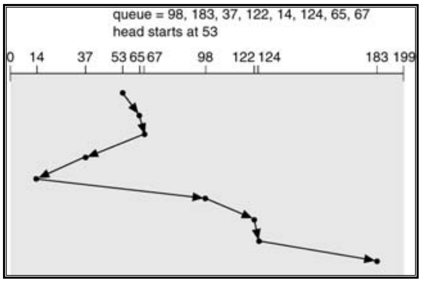


Shortest Seek Time First (SSTJ)

- Selects the request with the minimum seek time from the current head position.
- SSTF scheduling is a form of SJF scheduling; may cause starvation of some requests.

SSTF

Illustration shows total head movement of 236 cylinders

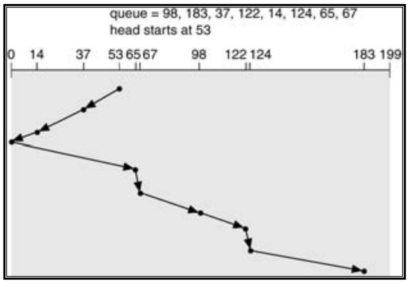


SCAN

- The disk arm starts at one end of the disk, and moves toward the other end, servicing requests until it gets to the other end of the disk, where the head movement is reversed and servicing continues.
- Sometimes called the *elevator scheduling*

SCAN (Cont.)

Illustration shows total head movement of 208 cylinders

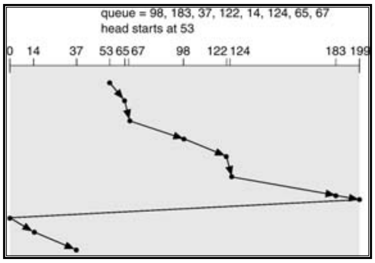


C-SCAN

- Provides a more uniform wait time than SCAN (with scan those in middle wait less)
- The head moves from one end of the disk to the other. servicing requests as it goes. When it reaches the other end, however, it immediately returns to the beginning of the disk, without servicing any requests on the return trip.
- Treats the cylinders as a circular list that wraps around from the last cylinder to the first one.

C-SCAN (Cont.)

Illustration shows total head movement of 382 cylinders



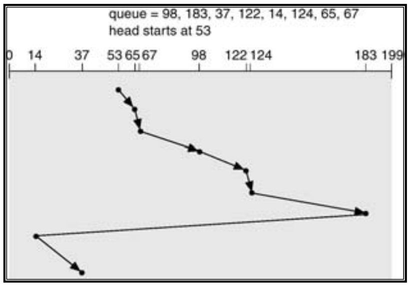
Misleading because seek time not a linear function of number of cylinders

C-LOOK

- Version of C-SCAN
- Arm only goes as far as the last request in each direction, then reverses direction immediately, without first going all the way to the end of the disk.

C-LOOK (Cont.)

Illustration shows total head movement of 322 cylinders



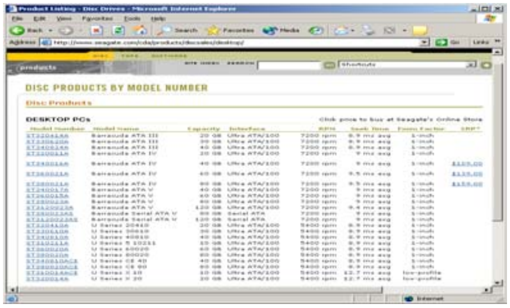
Selecting a Disk-Scheduling Algorithm

- SSTF is common and has a natural appeal
  - Starvation not observed to be a problem in practice
- SCAN and C-SCAN perform better for systems that place a heavy load on the disk.
- Performance depends on the number and types of requests.
- Requests for disk service can be influenced by the file-allocation method.
- Either SSTF or C-LOOK is a reasonable choice for the default algorithm.

Tracking Technology Trends

- Exact comparison between technologies changes all the time
  - How much slower is disk than main memory?
  - Variation even in disks and various memory technologies
- Tracking these things takes a fair amount of work

Current Drive Specs



Product Listing - Disc Drives - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://www.seagate.com/cda/products/discsales/desktop/
Go Links
products
Shortcuts
Internet

DISC PRODUCTS BY MODEL NUMBER

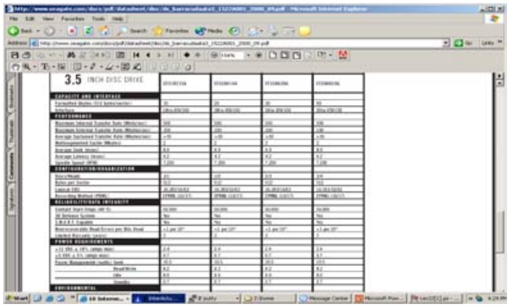
Disc Products

DESKTOP PCs

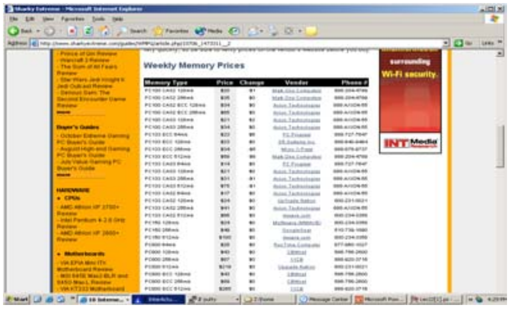
Click price to buy at Seagate's Online Store

Model Number	Model Name	Capacity	Interface	RPM	Seek Time	Form Factor	SRP*
ST320414A	Barracuda ATA III	20 GB	Ultra ATA/100	7200 rpm	8.9 ms avg	3.5-inch	
ST330620A	Barracuda ATA III	30 GB	Ultra ATA/100	7200 rpm	8.9 ms avg	3.5-inch	
ST340824A	Barracuda ATA III	40 GB	Ultra ATA/100	7200 rpm	8.9 ms avg	3.5-inch	
ST320011A	Barracuda ATA IV	20 GB	Ultra ATA/100	7200 rpm	9 ms avg	3.5-inch	
ST340016A	Barracuda ATA IV	40 GB	Ultra ATA/100	7200 rpm	9 ms avg	3.5-inch	$109.00
ST360021A	Barracuda ATA IV	60 GB	Ultra ATA/100	7200 rpm	9.5 ms avg	3.5-inch	$119.00
ST380021A	Barracuda ATA IV	80 GB	Ultra ATA/100	7200 rpm	9.5 ms avg	3.5-inch	$155.00
ST340017A	Barracuda ATA V	40 GB	Ultra ATA/100	7200 rpm	9 ms avg	3.5-inch	
ST360015A	Barracuda ATA V	60 GB	Ultra ATA/100	7200 rpm	9 ms avg	3.5-inch	
ST380023A	Barracuda ATA V	80 GB	Ultra ATA/100	7200 rpm	9 ms avg	3.5-inch	
ST3120023A	Barracuda ATA V	120 GB	Ultra ATA/100	7200 rpm	9.4 ms avg	3.5-inch	
ST380023AS	Barracuda Serial ATA V	80 GB	Serial ATA	7200 rpm	9 ms avg	3.5-inch	
ST3120023AS	Barracuda Serial ATA V	120 GB	Serial ATA	7200 rpm	9 ms avg	3.5-inch	
ST320410A	U Series 20410	20 GB	Ultra ATA/100	5400 rpm	8.9 ms avg	3.5-inch	
ST330610A	U Series 30610	30 GB	Ultra ATA/100	5400 rpm	8.9 ms avg	3.5-inch	
ST340810A	U Series 40810	40 GB	Ultra ATA/100	5400 rpm	8.9 ms avg	3.5-inch	
ST315211A	U Series 5 15211	15 GB	Ultra ATA/100	5400 rpm	8.9 ms avg	3.5-inch	
ST360020A	U Series 60020	60 GB	Ultra ATA/100	5400 rpm	8.9 ms avg	3.5-inch	
ST380020A	U Series 80020	80 GB	Ultra ATA/100	5400 rpm	8.9 ms avg	3.5-inch	
ST340810ACE	U Series CE 40	40 GB	Ultra ATA/100	5400 rpm	8.9 ms avg	3.5-inch	
ST380020ACE	U Series CE 80	80 GB	Ultra ATA/100	5400 rpm	8.9 ms avg	3.5-inch	
ST310014ACE	U Series X 10	10 GB	Ultra ATA/100	5400 rpm	12.7 ms avg	low-profile	
ST320014A	U Series X 20	20 GB	Ultra ATA/100	5400 rpm	12.7 ms avg	low-profile	

More details!



Memory Types and Prices



Sharky Extreme - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Buyer's Guides
HARDWARE
CPUs
Motherboards
surrounding Wi-Fi security.
INT Media

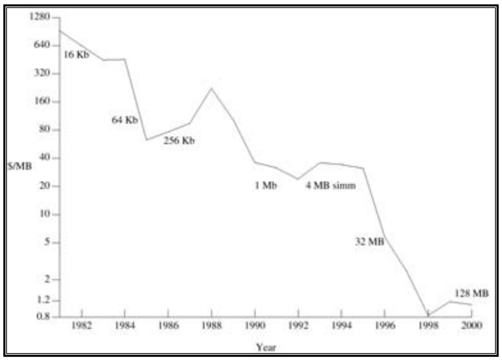
Weekly Memory Prices

Memory Type	Price	Change	Vendor	Phone #

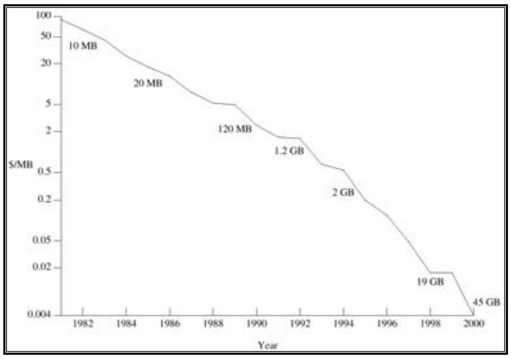
Two random points (10/07/02)

- Memory: 128 MB, PC 133, SDRAM, $45
  - $0.35/MB
  - ~8 nanosecond access time
- Disk: 20 GB, Ultra ATA/100, $109
  - $0.005/MB (1/2 penny per MB!!)
  - 9.5 ms average seek (what is average? Seek time increases with number of tracks moved but not linearly)
  - 4.16 ms average latency (1/2 rotation at 7200 RPM?)
  - 100 MB/sec burst transfer (25-41 MB/sec sustained transfer)
- Disk/Memory Ratios
  - Price: 1/70
  - Size: 160/1
  - Speed (Access time): 13 ms/8ns = 1625000/1

Price per Megabyte of DRAM, From 1981 to 2000



Price per Megabyte of Magnetic Hard Disk, From 1981 to 2000



OS adapts to performance trends?

- For the OS to make the right choices if needs to be aware of the trade-offs
  - Is the speed comparison between registers, DRAM and disk like the difference between your mind, your pocket and your book shelf *OR* is more like the difference between your pocket, the bookstore and Pluto?
  - How much computation/meta-data storage is reasonable to do to avoid a disk access?
  - Should we use DRAM as a file cache or to store more memory page for processes?
- "Right" answer changes with new generations of technology and OS source lives much longer than that?
- Can OS measure performance and be coded to react to measurements?

File Systems

- Today talked a bit about disk internals
- Despite complex internals, disks export a simple array of sectors

- How do we go from that to a file system?
- What do we exactly do we expect from a file system?

File System Basics

- FS are probably the OS abstraction that average user is most familiar with
  - Files
  - Directories
  - Access controls (owners, groups, permissions)

Files

- A file is a collection of data with system maintained properties like
  - Owner, size, name, last read/write time, etc.
- Files often have "types" which allow users and applications to recognize their intended use
- Some file types are understood by the file system (mount point, symbolic link, directory)
- Some file types are understood by applications and users (.txt, .jpg, .html, .doc, …)
  - Could the system understand these types and customize its handling?

Basic File Operations

UNIX	Windows
create (name)	CreateFile(name, CREATE)
open (name, mode)	CreateFile(name, OPEN)
read (fd)	ReadFile(handle)
write(fd)	WriteFile (handle)
sync(fd)	FlushFileBuffers(handle)
seek(fd, pos)	SetFilePointer(handle)
close(fd)	CloseHandle(handl)
unlink (name)	DeleteFile(name)
rename (old, new)	CopyFile(name)
	MoveFile(name)

Directories

- Directories provide a way for users to organize their files *and* a convenient way for users to identify and share data
- Most file systems support hierarchical directories (/usr/local/bin or C:\WINNT)
  - People like to organize information hierarchically
- Recall: OS often records a current working directory for each process
  - Can therefore refer to files by absolute and relative names

Directories are special files

- Directories are files containing information to be interpreted by the file system itself
  - List of files and other directories contained in this directory
  - Some attributes of each child including where to find it!!
- How should the list of children be organized?
  - Flat file?
  - B-tree?
- Many systems have no particular order, but this is extremely bad for large directories!

Path Name Translation

- To find file "/foo/bar/baz"
  - Find the special root directory file (how does FS know where that is?)
  - In special root directory file, look for entry foo and that entry will tell you where foo is
  - Read special directory file foo and look for entry bar
  - Find special diretory file bar and look for entry baz
  - Finally find baz

Next time

- How do you take an array of sectors and build a file system?
  - Free lists
  - Root directory
  - Finding the data blocks of a file
  - Caching commonly accessed data (like the root director ☺)

Outtakes

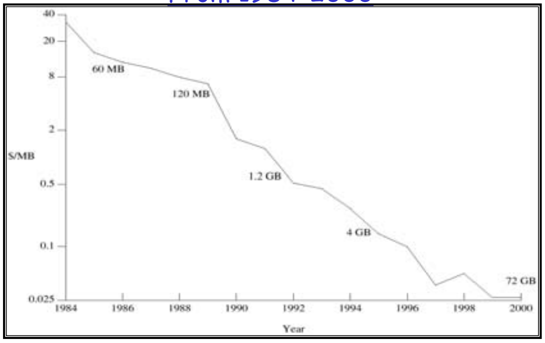
Disk Scheduling

- FCFS
  - Service requests in order they arrive
  - No possibility of data inconsistency
- SSTF (Shortest seek time first)
  - Do closest request first
  - Unfairly favors middle tracks
- SCAN (elevator scheduling)
  - Service requests in one direction until done, then reverse
- C-SCAN
  - Like SCAN, but when done do not reverse, return to the beginning

Cost

- Main memory is much more expensive than disk storage
- The cost per megabyte of hard disk storage is competitive with magnetic tape if only one tape is used per drive.
- The cheapest tape drives and the cheapest disk drives have had about the same storage capacity over the years.
- Tertiary storage gives a cost savings only when the number of cartridges is considerably larger than the number of drives.

Price per Megabyte of a Tape Drive, From 1984-2000


$/MB
Year
60 MB
120 MB
1.2 GB
4 GB
72 GB