Operating System - P112

directory implementation : special file containing information on the content inside directory
root directory always [redacted] at fixed location on disk
implementation keep pointer (i-node number)
to entry in the file metadata table

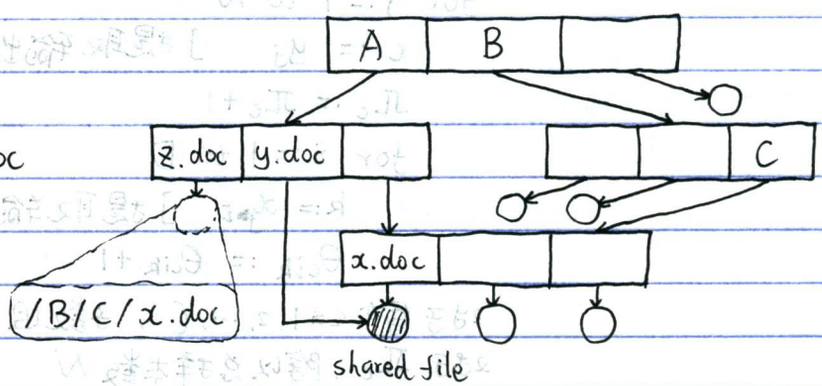
file name	ID in metadata table
.	<self>
..	<parent>
asg1	1
asg2	2
dir1	3

implementation keep all detail about file in directory entry
file metadata table [redacted] redundant

file name	start [redacted] block	length	other attribute
.	...	...	...
..	...	...	
asg1	10	2	...
asg2	24	3	...
dir1	16	2	...

finding file, /usr/[redacted]dir/file
look at entry in root directory, find i-node number of usr
check detail of usr in i-node table
read content of usr directory file, find i-node number of dir
check detail of dir in i-node table
read content of dir directory file, find i-node number of file
check detail of file in i-node table, then read

access the same file
using different paths
absolute path : /B/C/x.doc
hard link : /A/y.doc
symbolic link : /A/z.doc



Operating System - P113

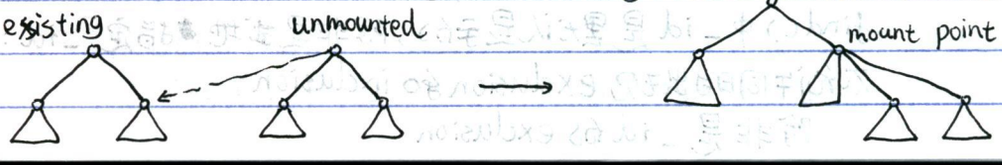
no cycle in directory structure : allow only link to file, not to subdirectory
garbage collection
every time new link added
use cycle detection algorithm to determine

sharing file with link, hard linking solution :
directory contain only i-node number
keep count of references exist to one i-node
soft (symbolic) linking solution :
create link file and put shared file in link file
when access link file, OS read and fetch path in link file
if directory contain disk block address of file
absolute path updated when parent add to file
hard link updated when parent add to file

remove file, file system operation lengthy
remove file directory entry
remove i-node used for file
release all blocks used by file

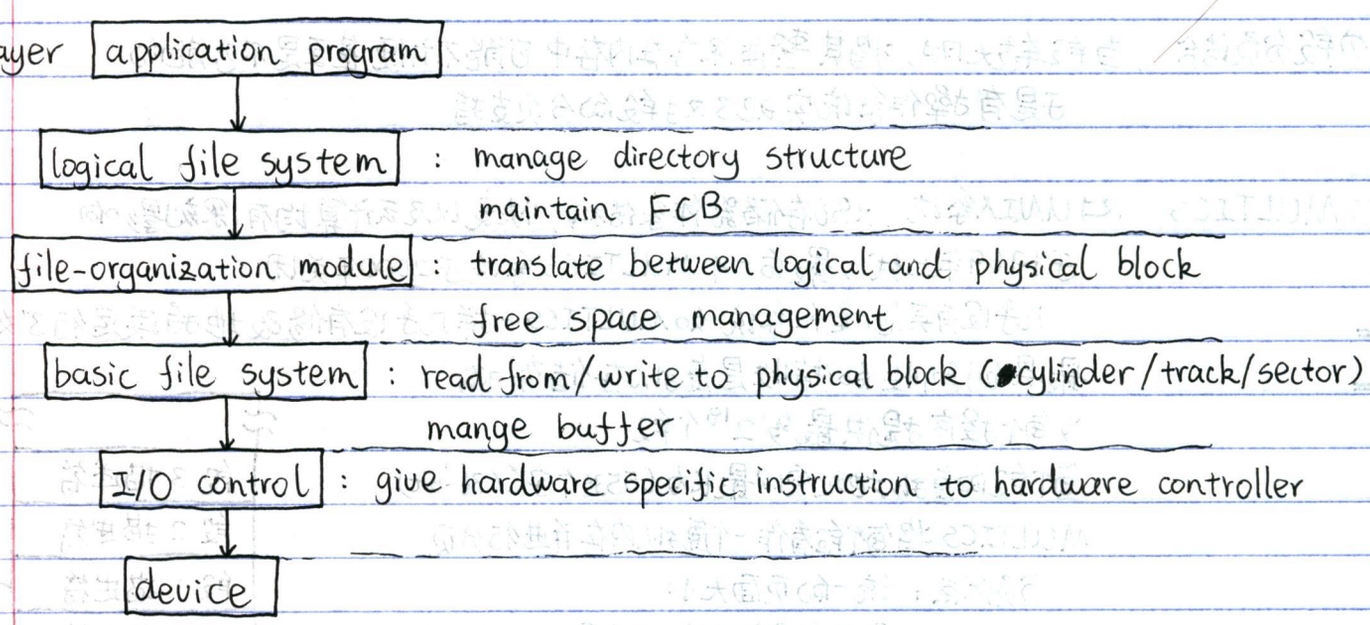
journaling file system, write log entry before doing sequence of operation
remove entry once done
if entry exsist in log after system recovery
then operation not finish
repeat : necessary that all operation idempotent (冪等)
can be repeated without any side-effect

mounting, file system must be mounted before can be accessed
unmounted file system mounted at mount point
UNIX : directory (usually empty)
Window : directory / (cusually) drive letter (盘符)



Operating System - P114

file system layer



FCB (file control block), UNIX : i-node

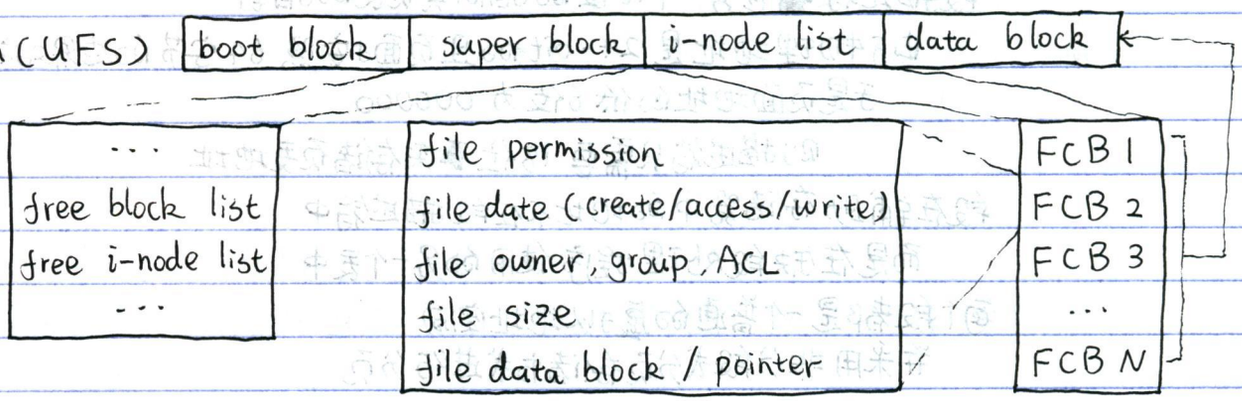
storage structure consisting of information about file

volume control block, contain volume detail

UNIX : super block

NTFS (new technology file system) : master file table

UNIX file system (UFS)



in-memory file system structure

mount table : information about each mounted volume

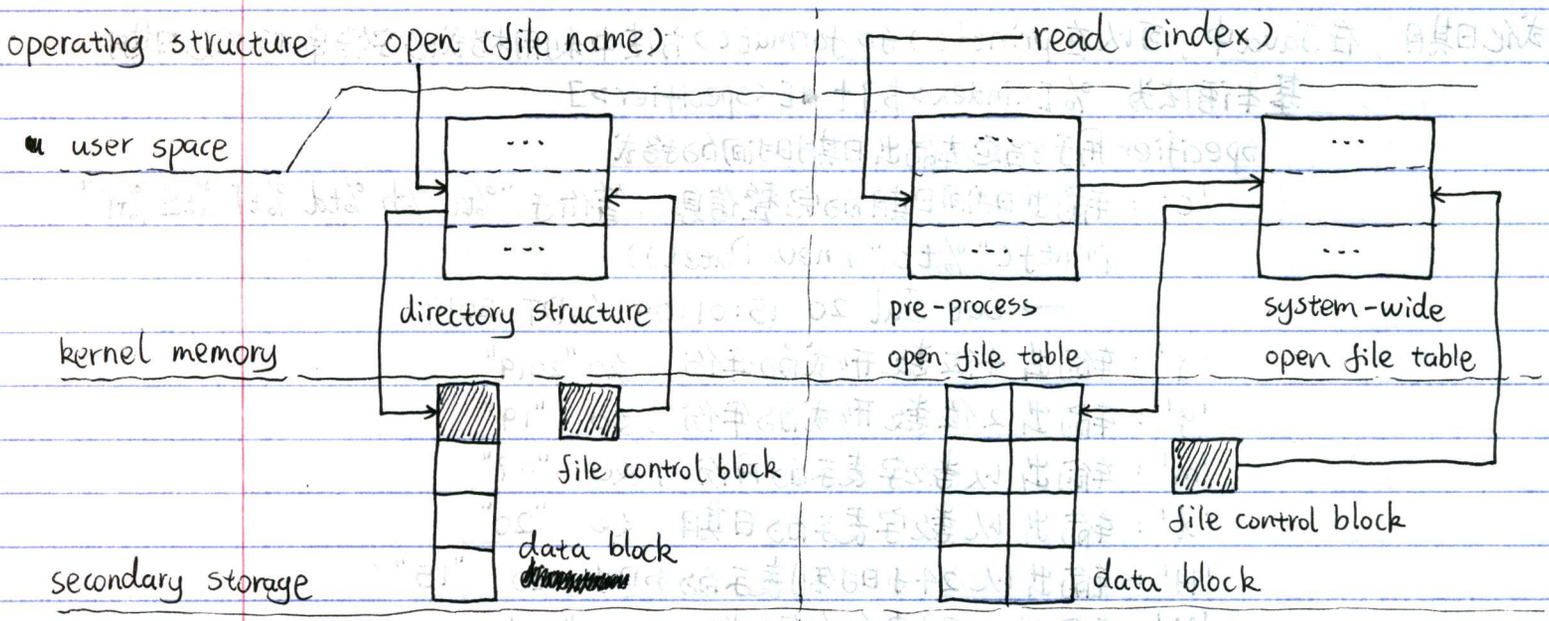
directory-structure cache : information of recently accessed directory

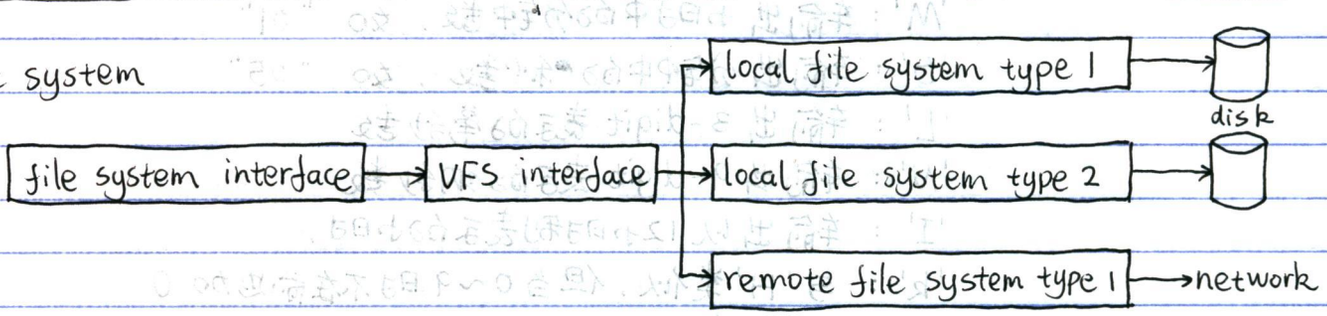
system-wide open-file table : copy of FCB of each open file

pre-process open-file table : pointer to appropriate entry in system-wide table

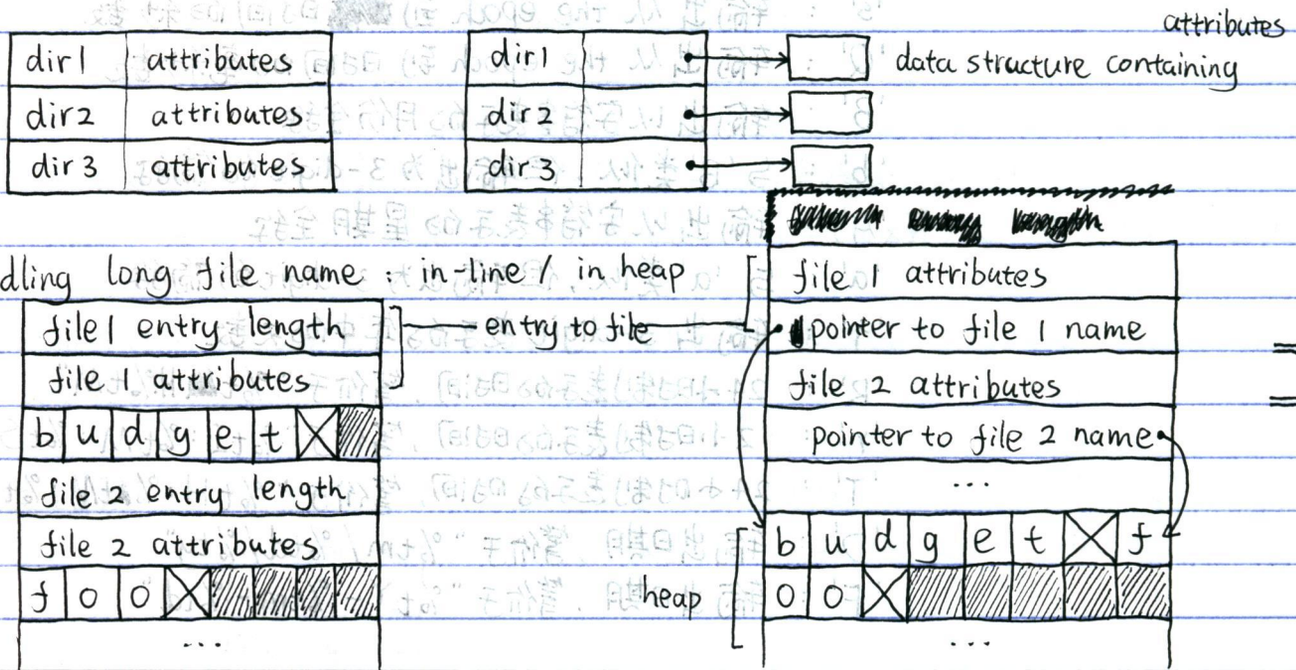
buffer hold data block when read from / written to disk

Operating System - P115





directory implementing : simple directory containing fixed-size entries
with disk address and attribute in directory entry
directory of each entry just refer to i-node



Operating System - P116

searching directory entry, no particular order of directory entry on older operating system
- lookup require linear search
- expect no huge number of files in any directory

recent Linux file system (ext3, ext4) structure the directory
- efficiently handle effectively unlimited number of files
- ext2 system: effective limit of few thousand files per directory
- structured directory can handle millions of files per directory

use HTree, based on B-Tree concept
- constant depth of one or two level
  - high fan-out factor
  - two levels needed only for larger directory
- based on hash of filename (not hash table)
- no balancing required

file system backup, physical: backup each block of disk
- use free block and bad block information from file system

logical: backup selected file and directory
- typically only user created data

full: backup entire target (full disk / selected part)

incremental: backup only data changed since last backup
- first backup always full backup
- thereafter, file: if file changed
  - directory: if file, sub-directory, changed or file/directory inside sub-directory

file system consistency, check consistency at system boot

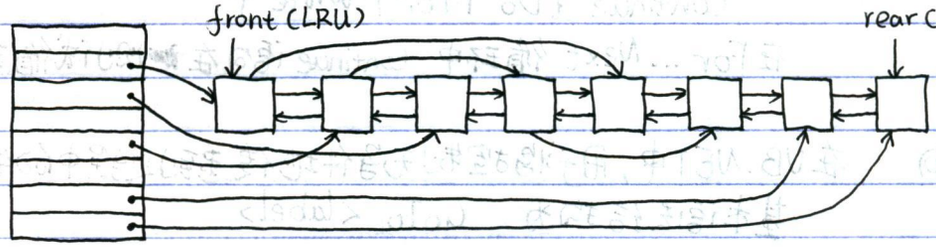
block consistency: block must be either in free list or part of one single file
- restore missing block by adding to free list
- remove duplication in free list
- remove duplication in file metadata by copying block

hard link consistency: number of times one i-node appear in directory must be equal to link count stored in the i-node
- restore link count

Operating System - P117

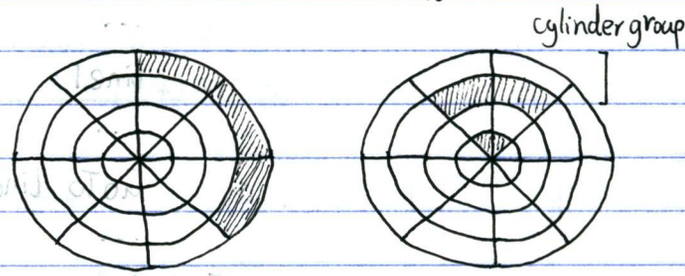
file system performance, caching : keep copy of blocks in memory
use replacement algorithm when cache full
block likely to be needed again soon
block essential for file system consistency
synchronize periodically
write-through cache : synchronized immediately
read block into cache ahead of time
performance improvement if file access sequential and file block contiguous
reduce disk-arm latency in mechanical hard drive
allocate block from same cylinder group to file (no seek after first block)
keep i-node close to file block

buffer cache data structure



some blocks rarely referenced two times within short interval
lead to modified LRU scheme

reduce disk arm motion: i-node placed at start of disk
disk divided into cylinder group
each with own block and i-node



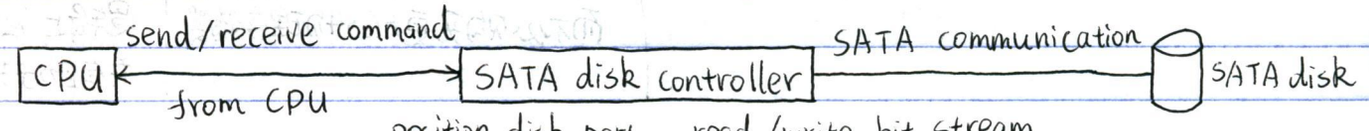
file sharing : remote file system,
distributed information system (distributed naming service)
implement unified access to information needed for remote computing
LDAP (lightweight directory access protocol, 轻量目录访问协议)
multiple failure possibility
disk failure, network failure, server failure
recovery from failure can involve state information
about status of each remote request
stateless protocol (NFS) include all information in each request
allowing easy recovery but less security

Operating System - P118

file sharing : consistency semantics, specify how multiple users access shared file simultaneously
- UNIX file system (UFS) semantics
  - write to open file visible immediately to other user of same open file
  - sharing file pointer to allow multiple users to read/write concurrently
- Andrew file system (AFS) session semantics
  - write to open file not visible immediately to other user
  - write only visible to session starting after file closed

I/O device
- block device, ex. hard disk
  - store information in fixed-size block
  - transfer in unit of entire block
- character device, ex. printer
  - deliver / accept stream of character, without regard to block structure
  - not addressable, doesn't have any seek operation
- other, ex. clock

device controller, electronic component talk to the device
- language standardized : SATA / SCSI / USB / Thunderbolt
- device can be built independent of controller

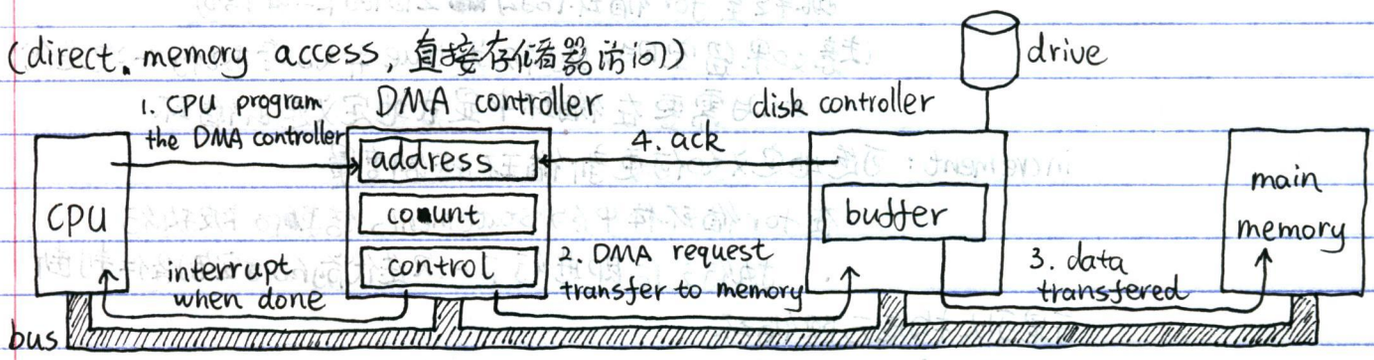


talking to controller, each controller has control register
- controller respond to data/command written to register
- controller can also have data register and buffer
  - used to hold data intermediately during transit from memory to device
  - OS can read from data buffer
  - Os can trigger controller to read device buffer to memory
- I/O port : controller register assigned specific number
  - OS can read from/write to controller register using the number
- Memory-mapped I/O : controller register/buffer mapped to specific physical memory address range
  - OS read from / write to memory region
  - controller write to / read from memory region

Operating System - P119

memory-mapped I/O, can communicate with device controller with memory read/write
no assembly coding necessary
OS can allow user program to control device
set up page table accordingly
all instruction that can access memory can access device
can easily check state of register
instead of first reading to memory location then checking
must be able to selectively disable memory caching

DMA (direct memory access, 直接存储器访问)



interrupt controller, interrupt generated by device read by interrupt controller
device assert one signal on one assigned interrupt line
controller look out for signal
put number on address line and assert interrupt line going to CPU
service routine notify controller of service completion
by writing to special port of controller
controller can notify device of service completion
and attend to pending interrupt from another device

programmed I/O, cpu determine if device available
and issue command to controller for every byte/word needed to read/write

interrupt-driven I/O, remove polling or busy waiting
if device ready for next command
can be used controller can (be programmed to) generate interrupt on command completion
controller notify CPU when command complete
CPU then carry out next command

Operating System - P120

I/O using DMA , interrupt-driven I/O require interrupt handling for every byte/word of data
programmed I/O with hardware support (DMA)
let DMA controller talk to device controller
and transfer data (whatever DMA controller buffer can hold)
DMA controller issue interrupt to CPU when buffer full

I/O software layer

software	user-level I/O software : write to file D:\test.txt
	device-independent operating system software : find disk D:\
	device driver : issue command to disk
	interrupt handler : wake up driver when command complete
	hardware

interrupt handler , interrupt : mechanism for controller to inform CPU of event
interrupt handler run when interrupt generated
OS responsibility : save process state
set up execution environment for handler
run handler
restore state

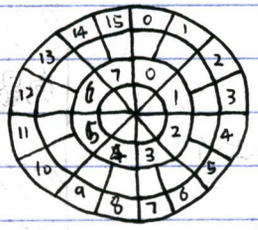
device drive , program know command to issue to extract specific functionality
from specific device controller
device controller expect command from CPU : specific to device
run as part of the kernel : OS can implement driver
allow for driver installation
OS define standard interface used to interact with driver

device-independent I/O software , perform I/O function common to all device
provide common interface to user program
uniform interface : device driver
I/O device naming
device protection
buffer : making data available in expected size
error reporting
managing request for device

Operating System - P121

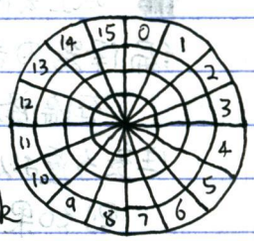
user-space I/O software, system call allow interaction with device
read / write / print / scanf
implemented as part of library
may also perform formatting or spooling
spooling : collect data from multiple processes and handle one by one

magnetic disk



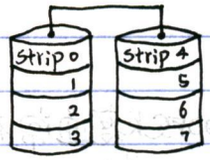
physical geometry of disk
with two zones

possible virtual geometry for disk

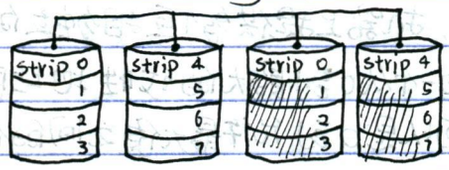


RAID (redundant array of inexpensive disk, 廉价冗余磁盘阵列)
又称为独立冗余磁盘阵列 (redundant array of independent disk)

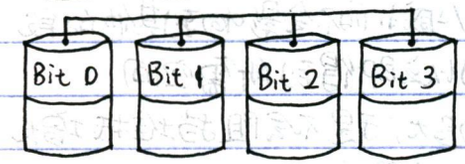
RAID 0



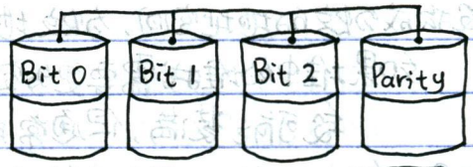
RAID 1



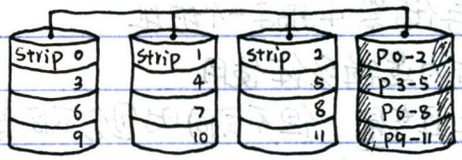
RAID 2



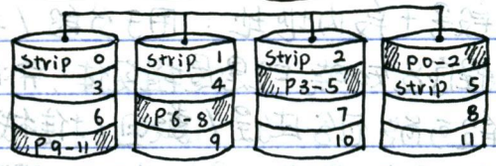
RAID 3



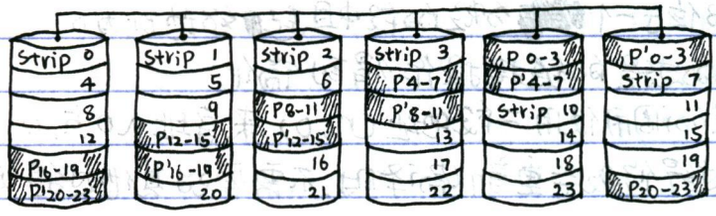
RAID 4



RAID 5



RAID 6



direction of disk rotation

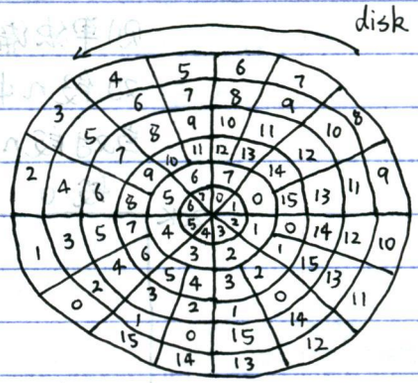


illustration of cylinder skew

disk formatting, sector

interleaving

no single double

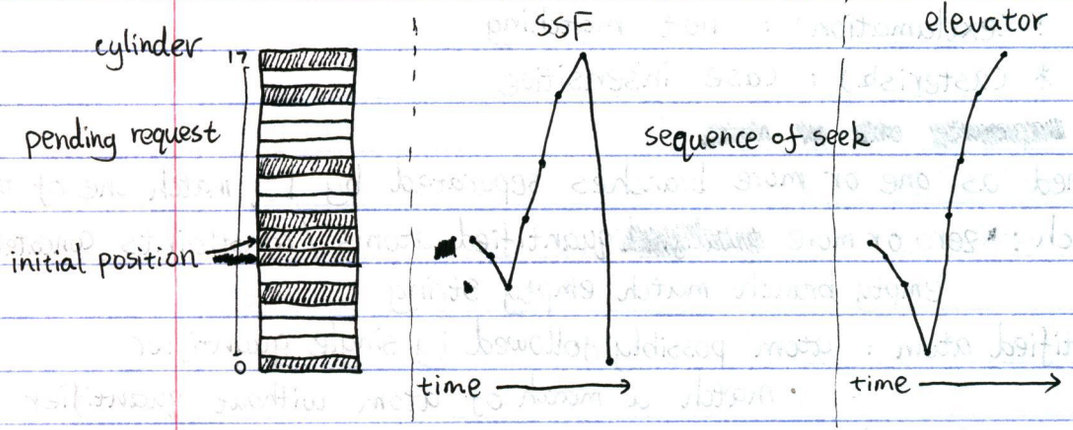
Operating System - P122

disk arm scheduling algorithm, factor of disk block read / write

seek time : time to move arm to proper cylinder

rotational delay : time for the proper sector to come under the head

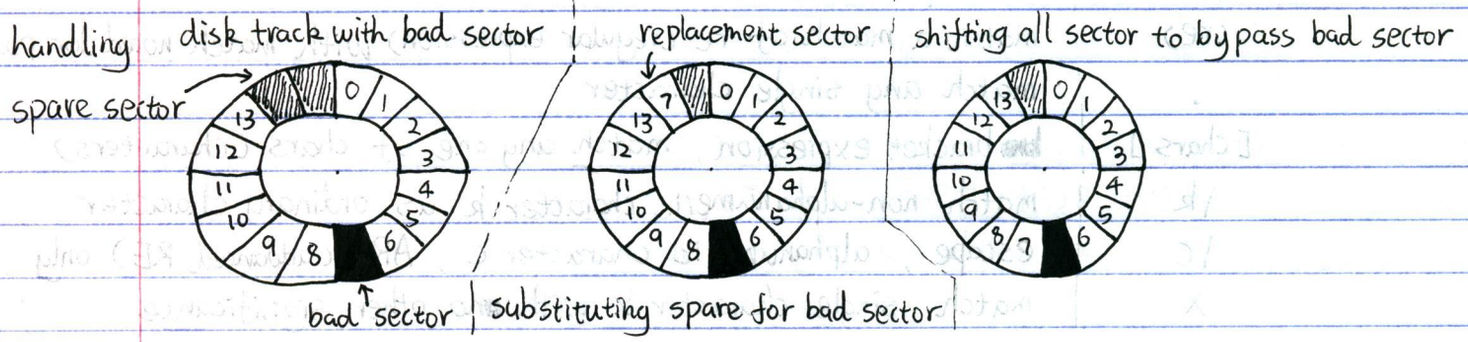
actual data transfer time



shortest seek first (SSF) disk scheduling algorithm

elevator algorithm for scheduling disk request

error handling



type of multiple processor system

shared-memory multiprocessor

message-passing multiprocessor

wide area distributed system