Operating System - P101

security violation : breach of confidentiality : unauthorized reading of data
breach of integrity : unauthorized modification of data
breach of availability : unauthorized destruction of data
theft of service : unauthorized use of resource
denial of service (DOS) : prevention of legitimate use

principle of least privilege : guiding principle of protection
program / user / system should be given just enough privilege to perform task
limit damage if entity has bug / get abused
can be static during life of system / process
dynamic : domain switching / privilege escalation by process as needed
rough-grained privilege management easier and simpler
but least privilege now done in large chunk
tranditional UNIX process have abilities
either of associated user, or of root
fine-grained management more complex and overhead
but more protective
file ACL (access control list)
protection rule applied to domain
domain can be user / process / procedure

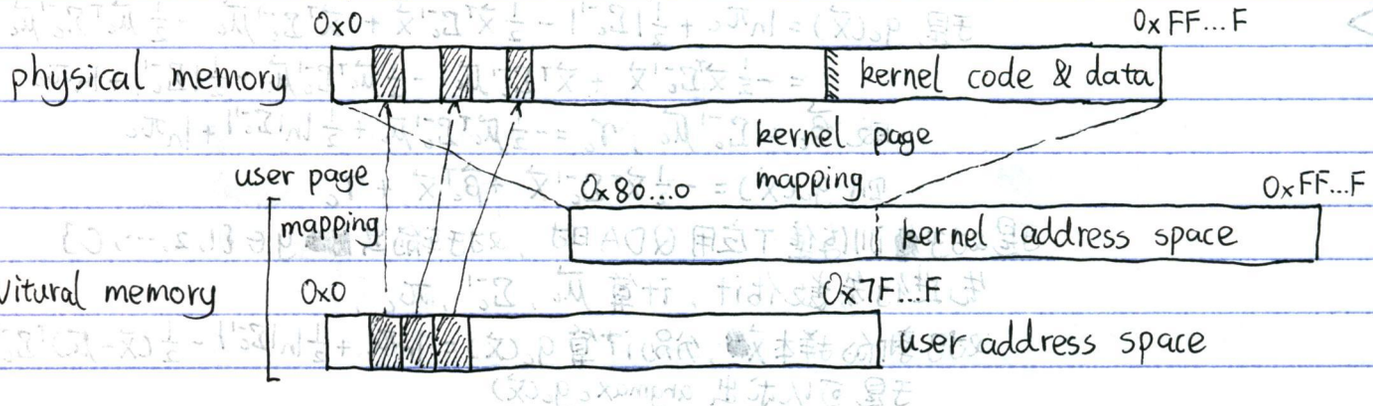
system secure if recource used / accessed as intended under all circumstances
unachievable
intruder (cracker) : attempt to breach security
threat : potential security violation
attack : attempt to breach security

violation method : masquerading (breach authentication)
pretending to be an authorized user to escalate privilege
replay attack : as is or with message modification
man-in-the-middle attack : intruder sit in data flow
masquerading as sender to receiver, and vice versa
session hijacking : intercept already-established session
to bypass authentication

Operating System - P102

meltdown (火宮断), exploit of security flaw in x86 hardware
declosed in Jan. 2018, vulnerability around years
take advantage of hardware optimization in the architecture
able to read from any location in memory
on unpatched Linux and Mac system
able to read from many sensitive memory locations
in unpatched Windows 10 system
Intel x86 / IBM POWER / ARM processor vulnerable
require attacker to have access to run program on target system
cloud service provider particularly vulnerable
possible exploit from JavaScript, attack code make system vulnerable
visiting web page / loading ad
software patch for all recent operating system
greatly reduce system performance
use combination of hardware feature
shared page table between user program and OS kernel
out-of-order execution
side channel to determine cache usage by user program

shared page table, modern operating system share page table with user program
until discovery of meltdown vulnerability
entire physical memory mapped into page table for kernel access
kernel-only page have User/Supervisor (U/S) bit set to 0
in page table entry and/or page table directory
permit supervisor mode only (privilege level 0)
user program access (privilege level 3) not allowed



Operating System - P103

shared page table, OS kernel need access to all page in memory
changing page table is expensive
may invalidate TLB
new page table may not be in memory cache
in theory, kernel page safe from access by user program
hardware protection would cause fault
if program in user mode try to directly access kernel address space

out-of-order execution, modern processor fetch instruction ahead of currently executing one
may partially execute future instruction ahead of time
result discarded if turn out not needed
while long running instruction still executing on the result
system may fetch and execute faster instruction not depend
CDC 6600 : one of first implementation
IBM S/370 : work ahead on both branches of conditional jump
result from branch not taken discarded

side-channel data, side-channel refer to data inferred although not directly accessible
information revealed by timing
information revealed by observing other system behavior
cache side-channel : several means to determine if address in the cache
typical general method :
flush cache of target address
by accessing range of other address to push target out of cache
wait for / perform target address
probe cache line for target address, and time the probe
fast time : in cache, slow time : not in cache
method not completely reliable due to other process running in system
repeated probes increase confidence level

have no control over hardware feature
out-of-order execution
side-channel to determine cache usage by user program

Operating System - P104

eliminate shared page table by using separate page table for kernel
x86 architecture : small amount of kernel code
must be mapped into user page table
interrupt handler / switch page table
code can be designed to not contain any sensitive data
using separate page table can cause potentially large performance impact
every system call require one page table change to process
and another to return to user process

Spectre (幽灵), hardware vulnerability related in technique to Meltdown
affect almost all modern CPU
break isolation between different applications
allow attacker to trick error-free program into leaking secret
program follow best practices
safety check of best practice actually increase attack surface
may make application more susceptible to Spectre
harder to exploit than Meltdown, also harder to mitigate
possible to prevent specific known exploits through software patch
different variants of Spectre attack discovered
some cannot be fixed in software
and will require hardware to be updated

abstraction process for CPU : user program not have to know presense of other programs
logical memory for memory : user program not have to know physical memory
file for disk : user program not have to know where data reside on disk

basic disk access, disk as linear array of fixed size blocks
block size : parameter of file system implementation, typically 4KB
operation : read from block number k
write to block number k
find information
keep one user from accessing another user's data
trace free block

Operating System - P105

file — persistent logical unit of information on the disk
- abstraction from physical property of storage device
  - not care how actual information stored on disk
- viewed as contiguous entity
  - not care where actual information stored on disk

managed by operating system
- file system : detail of how file organized and managed on disk

user interface , file naming : how user refer to file
- name of variable length
- extension to help remember content in file

file structure : how data organized inside file
- sequence of byte
  - only application program can interpret the meaning
- sequence of record or tree of variable length record

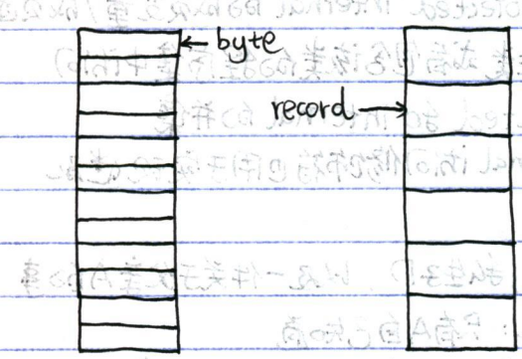
file type : categorizing content in file
- ASCII / binary / character , block special / directory / link

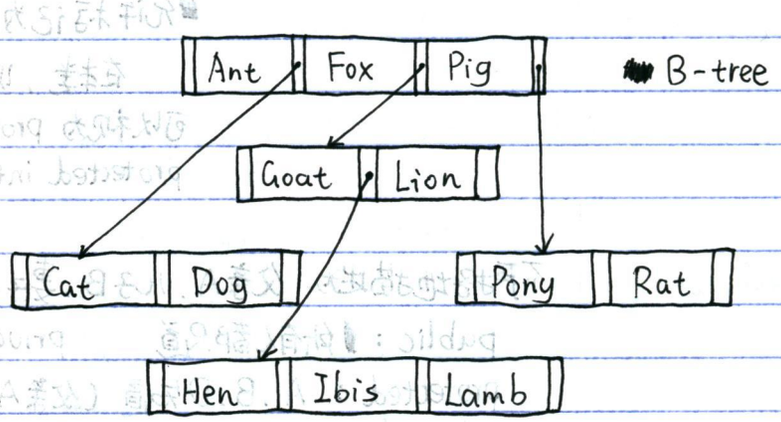
file structure , no structure : sequence of word / byte

simple record structure :
- line : fixed length or variable length

complex structure
- formatted document / relocated load file

decided by : operating system / program





Operating System - P106

user interface, file access : how file be accessed
sequential access / random access
file attribute : metadata on file
size / access right / protection / time of creation, modified, accessed
name : only information kept in human-readable form
identifier : unique tag / number within file system
type : needed for system support different types
location : pointer to file location on device
size : current file size
protection : control who can read / write / execute
time, date, user identification :
data for protection / security / usage monitoring
file operation : action allowed on file
file is abstract data type.
allow set of operation performed on file
create / delete / open / close / read / write / append / seek
/ rename / get attributes / set attributes
/ truncate / reposition within file

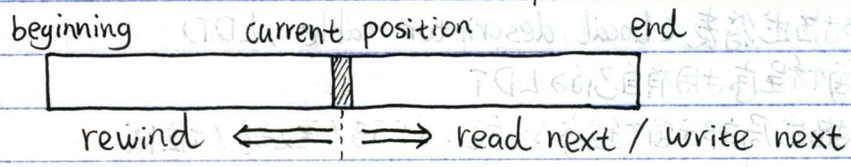
open file
file pointer : pointer to last read / write location
per process has the file open
file-open count : counter of number of times one file is open
allow removal of data from open-file table
when last process close file
disk location : cache of data access information
access right : per-process access mode information

locking
provided by some operating system / file system to mediate access to file
- shared : several processes can acquire the lock concurrently
- exclusive : only one process at one time
- mandatory : access denied depending on lock held and requested
- advisory : process can get status of lock and decide what to do

Operating System - P107

access

sequential access : based on tape model



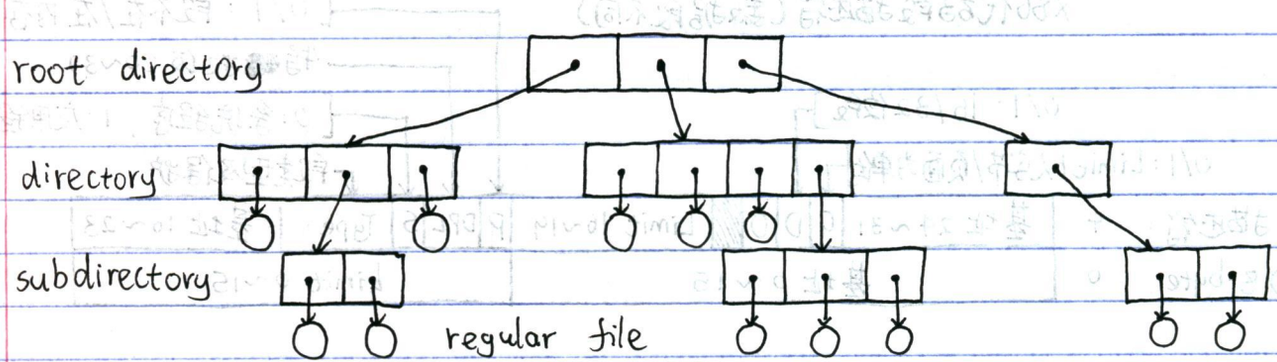
direct access : based on disk model
data in the form of logical block (record)
read / write / position to relative block number

sequential access	direct access
reset	n := 0
read next	read n ; n := n+1
write next	write n ; n := n+1

directory

symbol table that translate file name into directory entry
directory entry hold information about file or other directory
special type of file : keep track of other file
help user manage other file
OS graphical interface show file and directory differently
OS know how to read content of directory

single-level : only one directory in system
hierarchical : multiple level



path : sequence of directory leading to one file
absolute path : sequence begin at root directory
relative path : sequence begin at another directory
(current working directory)

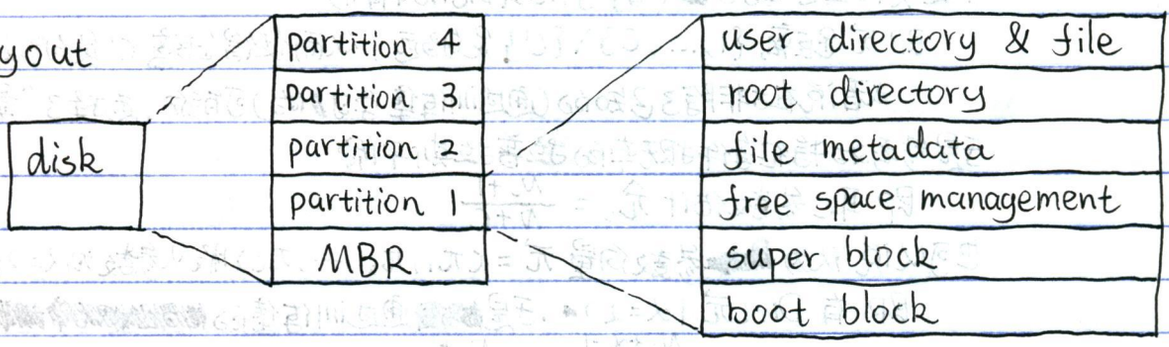
Operating System - P108

directory operation , similar to file
create / delete / opendir / closedir / readdir / rename / link / unlink

symbolic link , special file contain path of another file
space allocated to store the pathname
OS read path information to reach real file

hard link , maintain a reference count for the file pointed to by the links
no space allocated

file system layout



MBR (master boot record , 主引导记录) , contain information on partitions

file system component inside partition
boot block : contain information needed by the system to boot OS in the partition
super block : contain parameter information on the layout
Windows NTFS : master file table
number of blocks in the partition
size of block
free block count
location of root directory
free space management : tracking free block
file metadata : which blocks go with which file
othe file attribute
root directory : file / directory in root directory

Operating System - P109

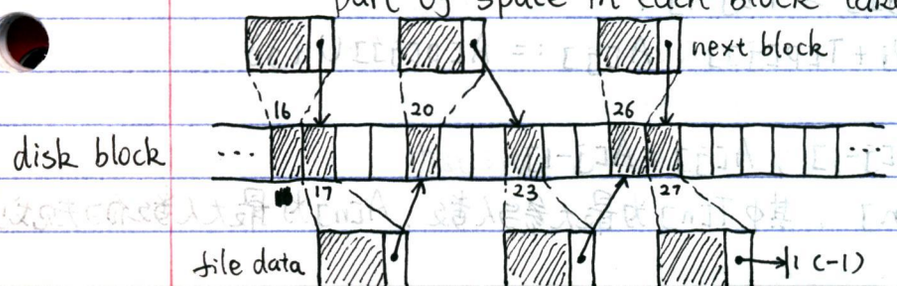
file-to-block mapping , track which blocks belong to a file

contiguous allocation , each file occupy set of contiguous blocks on the disk
only require : starting location (block number)
length (number of blocks)
random access
dynamic storage-allocation problem
external fragmentation
handle growth of file

file metadata	id	start block	length	other attribute
asg1	1	11	2	...
asg2	2	24	6	...
dir1	3	16	2	...

extent based system , modified contiguous allocation scheme
used to handle problem due to file size increse
allocate disk blocks in extents
extent : contiguous blocks of disk
allocated for file allocation
one file consist of one or more extent

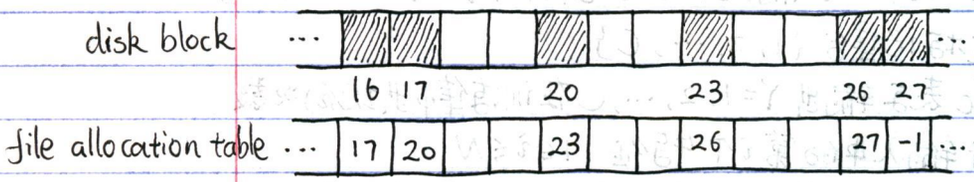
linked allocation , each disk block store address of file's next block
blocks scattered anywhere on the disk
only need starting address (block) of file
no external fragmentation
no random access : block can be accessed only by traversing previous blocks
part of space in each block taken by pointer



file metadata	id	start block	...
...	...	...	...
asg	11	16	...
...	...	...	...

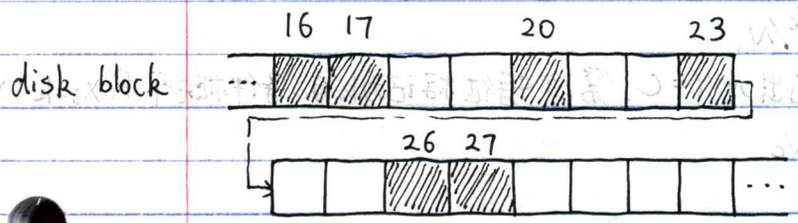
Operating System - P110

FAT (file allocation table), reserve section of disk to contain table of block address
each entry indicate where the next block of data for the file



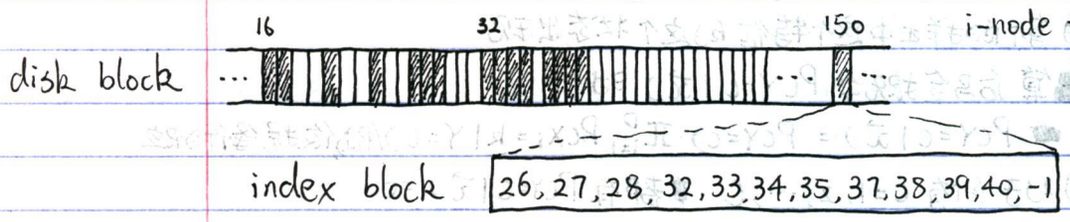
file metadata	id	start block	...
[redacted]	...	...	...
asg	11	16	...
...	...	...	...

indexed allocation, maintain an array of disk-block address
store all disk blocks of one file in one place
index-node (i-node): each row in i-node table



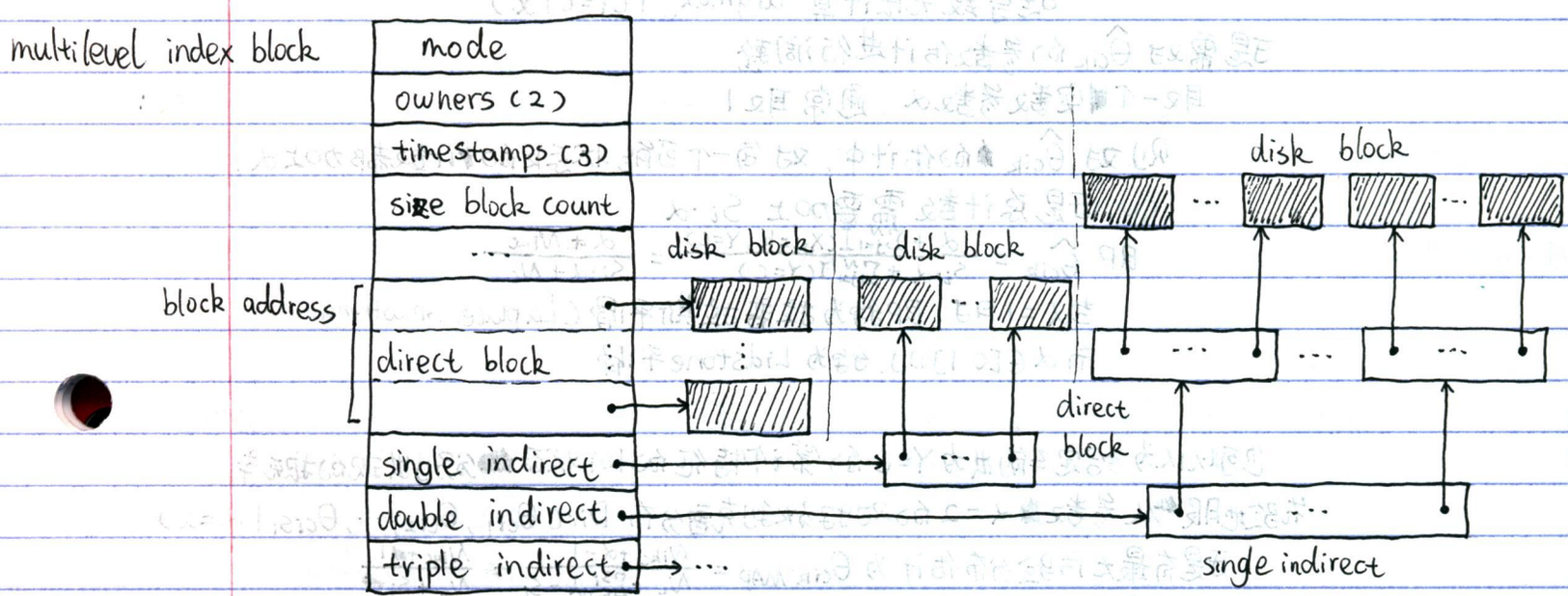
i-node table	id	file block	...
...	...	...	...
asg	11	16, 17, 20, 23, 26, 27	...
...	...	...	...

index block, used for growing file
containing remaining blocks of file



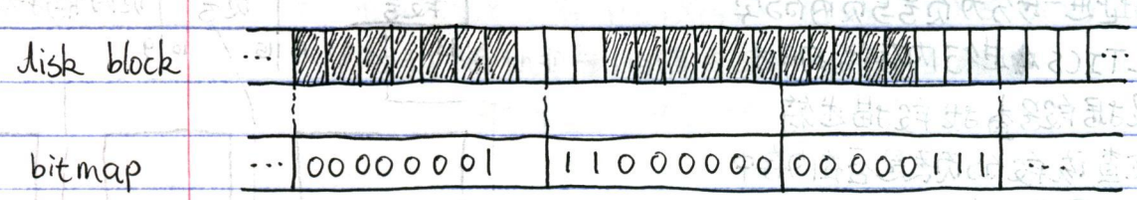
i-node table	id	file block	...
...	...	...	...
asg	11	16, 17, 28, 23, 150	...
...	...	...	...

multilevel index block



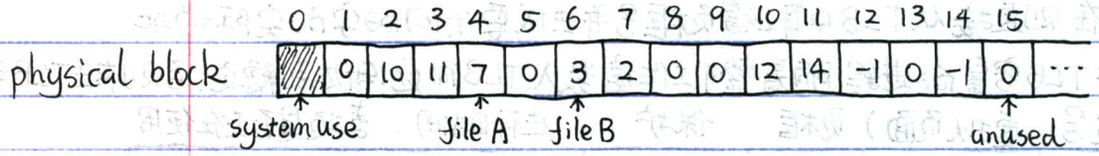
Operating System - P111

free space management bitmap, keeping track of free block on disk



linked lisk based free space, store block number of all blocks free
- reserve blocks to store block number
- store stare free block number and number of free blocks from there
  - if free block appear in long run
  - not good when disk heavily fragmented
- only part of free list need to be in memory at one time
- don't require any dedicated disk space
  - list size shrink/grow as free size

free block using FAT file system, file allocation table with one entry per disk block
- file is linked list of block index in the FAT array
  - with special value to indicate end of list
- also used to track free block
  - Microsoft set free block to 0
  - necessary to search FAT to find free block



free block using FAT file system - linked list
- variants FAT use linked list to track free space
- advantage: just like one file
  - constant time to find free block
- disadvantage: hard to find contiguous free space
  - possibly harder to recover from memory corruption