

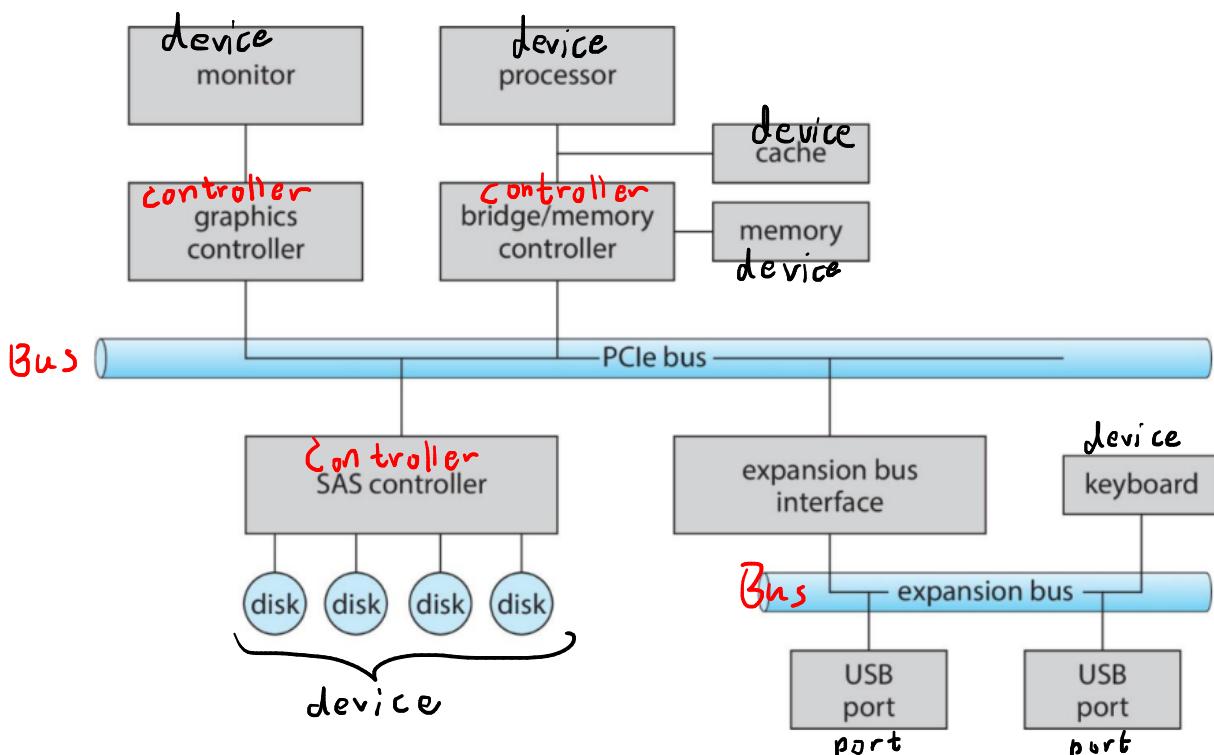
I/O Systems

ຈິງຂອງດ້າຍ

- Storage (HDD, SSD, Flash Drive)
- Transmission (LAN, Wi-Fi, Bluetooth)
- Human - Interface (Printer, Mouse, Keyboard)

Concept ແລ້ວ ຂອງ I/O

- Port
- Bus → PCIe
- Controller → ອຸປະນົກຄັງກຳ port, bus



Polling

- 1.) ລັບ busy-bit ຈາກ status register ຈົນ 0
- 2.) Host ຖະ set i) := read/write
- 3.) Host ^{set} ready-bit
- 4.) Controller ^{sets} busy-bit ພົມ= execute transfer
- 5.) controller ^{clear} busy-bit, error-bit, command-ready bit
ເມື່ອ transfer - done

ក្នុងការទិន្នន័យ I/O និង Interrupts

- Polling - សារពីការកែចម្លាយ 3 instruction cycle

→ ចូល status-bit

- CPU Interrupt-request line ← ទីន I/O

- Interrupt Handler (ទិន្នន័យ Interrupt)

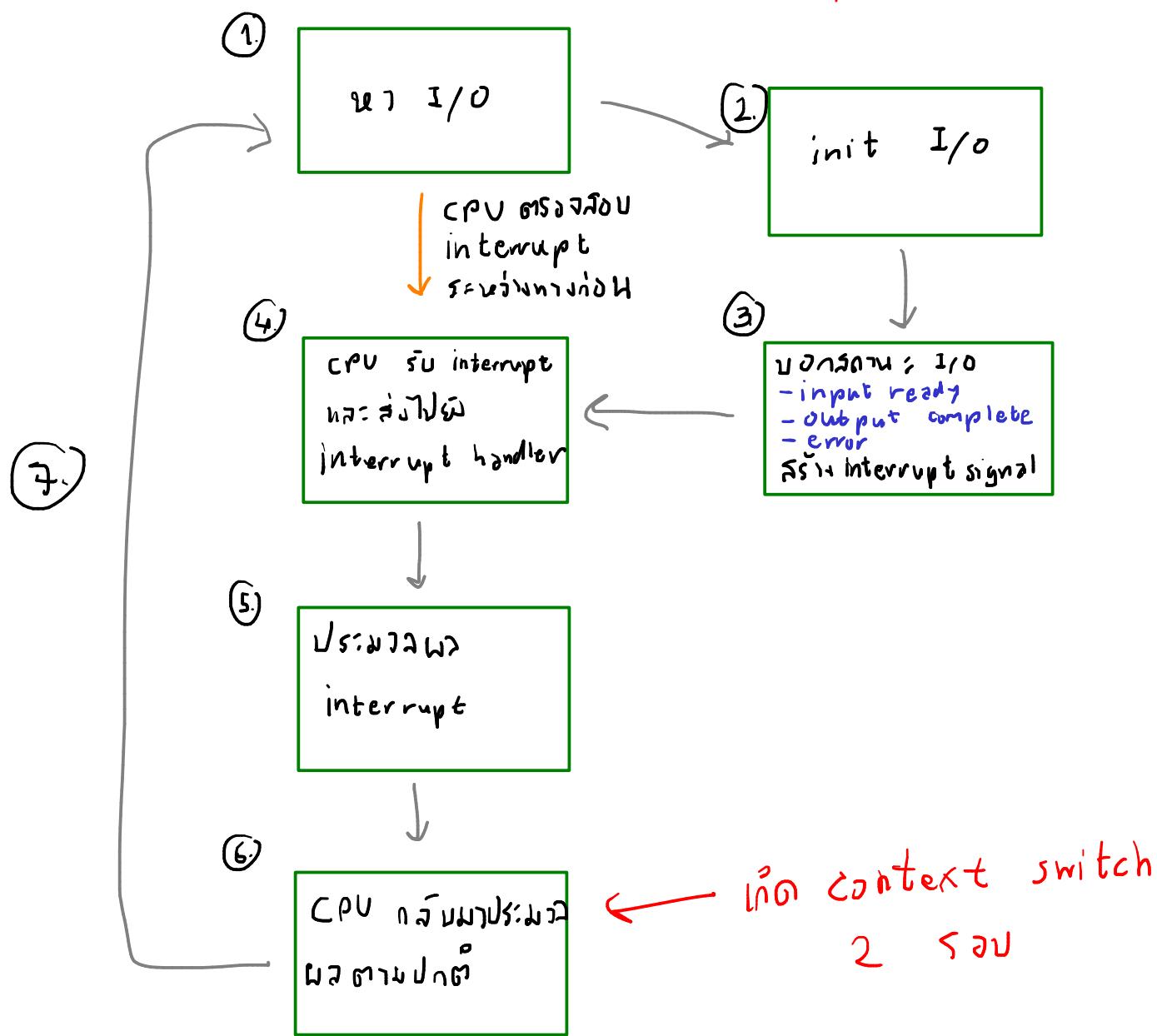
- Interrupt Vector (ទិន្នន័យដែលអាចបញ្ជូន handler រួចរាល់បាន)

- context switch
- priority

Interrupt - Driven I/O cycle

CPU

I/O controller



- การที่ Interrupt ยังไงก็รับ exception

- Page Fault

- trap

- Multi-CPU สามารถรับ interrupt พร้อมกันได้ (ด้วย OS ดูแล)

- ต้องเร็ว

โดยปกติ OS จะ desktop จะเกิด interrupt หลังร้อน / รีบ
Server จะเกิด interrupt หลังพ้น / รีบ

Direct Memory Access

- ต้องมี hardware controller

- ใช้ส่วนรับ I/O $\xleftrightarrow{\text{ถ่ายกับ}}$ memory (RAM) โดยตรง

- โดย OS จะใช้ Command ต่อไปนี้ memory

- src \rightarrow dest address

- R / W mode

- หนึ่งเรื่องนวน bytes

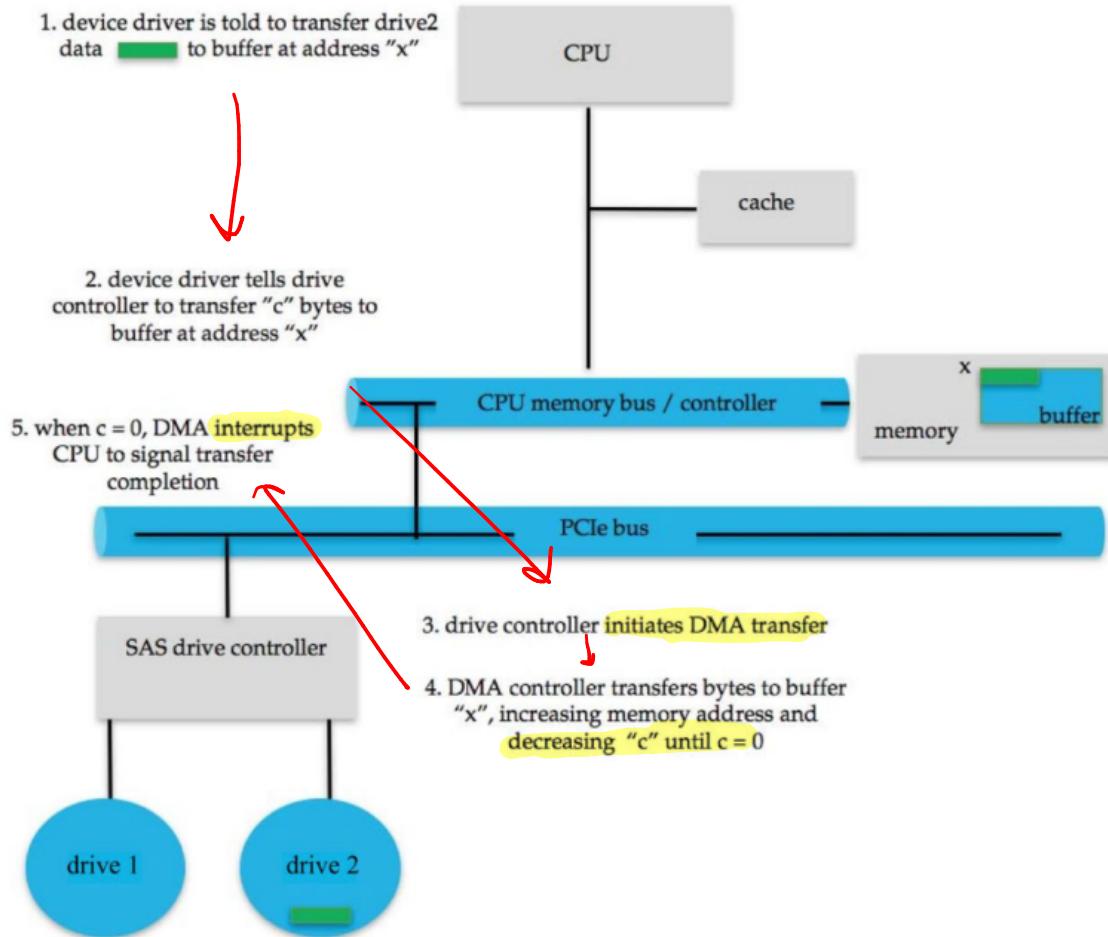
- ตำแหน่งที่ปัจจุบัน DMA อยู่ใน DMA controller

- Bus Mastering (steal CPU cycle)

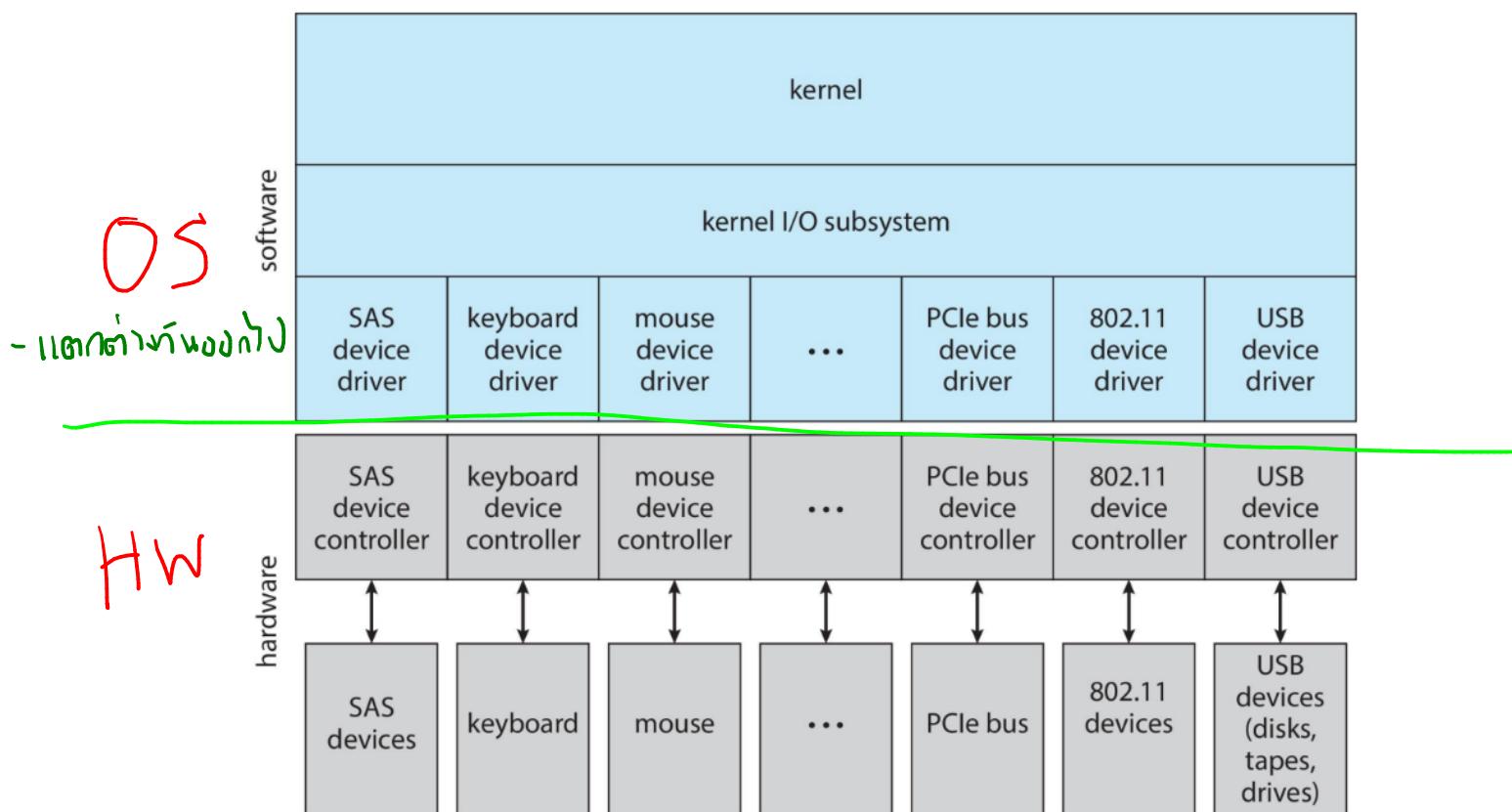
- เนื่องจากสิ่งใด DMA ทำงานเสร็จสิ้น \rightarrow ส่ง interrupt

* ถ้าเป็น virtual address จะเรียกว่า DVMA

የኢትዮጵያውያን የሰነድና ስርዓት በግልጽ



Application I/O Interface



ລົກສອນຂອງ I/O devices

Aspect	Variation	Ex.
- data - transfer mode	char, block	terminal, disk
- access method	seq, rand	modem, CD
- transfer schedule	sync, async	tape, keyboard
- sharing	dedicated, sharable	tape, keyboard
- device speed	latency, seek time	HDD, SSD
- I/O direction	R, W, R/W	CPU, GPU, disk

OS ໄ້ Group I/O ໃນ 4 ຈຳກັນ

- Block I/O
 - disk drive
 - read, write, seek
 - DMA
- Character I/O
 - keyboard, mouse
 - get(), put()
- Memory-Mapped File access
- Network Socket
 - socket interface
 - select()
 - ລົມາຮັກ ສ່ວນໄອບວຍແບບ

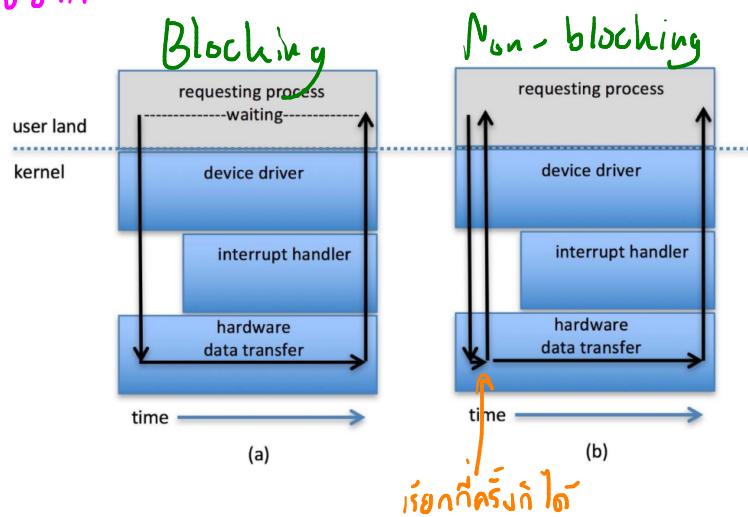
Clock & Timers

- current time
- elapsed time
- timer

ioctl()

< pipes, FIFOs, streams, queues, mailbox

- Blocking I/O → process មិនអាចធ្វើការទៅ I/O ឡើយ
- Non-Blocking I/O → I/O ត្រូវបានដោះស្រាយឡើង
- multi-thread
- Asynchronous → process នឹងអាចធ្វើការ I/O បានដោយពេលវេលា
- ទីនៅ

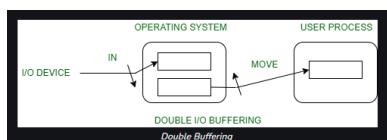


Vectorized I/O

- ដែលអាចប្រើបានបន្ទាយ I/O បានបាន
- ដែលអាចប្រើបាន system call បានបាន
- ដែលអាចប្រើបាន context switching បានបាន
- ដែលអាចប្រើបាន system call បានបាន

Kernel I/O Subsystem

- Scheduling
- Buffering - រាយការណ៍ data នៃ memory ឱ្យបានបង្ហាញបានស្ថិត
- cope speed / transfer size និងទីតាំង
- Double Buffering



- Caching - រាយការណ៍ copy [ខ្លះ]
 - Spooling - រាយការណ៍ hold output ex. printer
 - Device Reservation - រាយការណ៍ឱ្យកីត្តិថត device
- ↓
ក្នុងក្នុង deadlock ឡើ

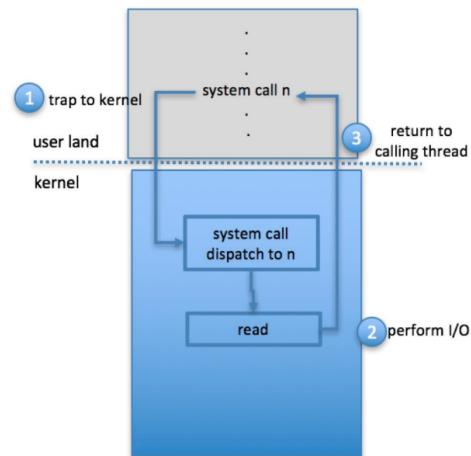


Error Handling

- OS ສາມາດ Recover จากຄວາມຝຶກຂາດບາງ I/O ໄດ້
 - ເອີ້ນ -Retry
 - Return Error Code
 - ປັບທຶກລົງ Log

I/O protection

- I/O ຕັ້ງເຮັດກ່າວ System calls ໃຫ້ໜີ້
(User ຖືສາມາດກ່າວ)



Kernel Data Structures

- ໄກສົງ state ຂອງ I/O
- ນິຍາກຂາຍຮູ່ໃນກາຕົກສົງ
- ex. Windows ອະນຸ message passing

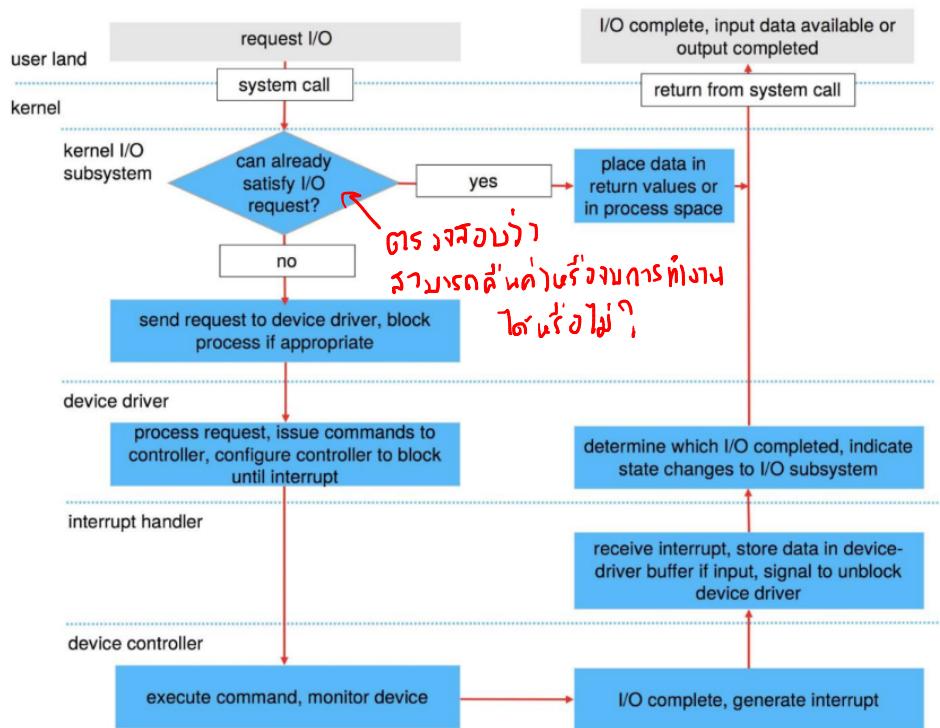
Power Management

- ອະນຸ cooling ອົບມືອດງານຮົບນ
- ອະນຸ mobile device ເປັນ OS First-class aspect

ex. Android

- ສຽງ tree
- ວັດໄຟ້ນຳມື້ງດີ ອະນຸ → ປິດ
- ດັກໃນ tree ນິກໍລື ນັ້ນມີແລ້ວ → ປິດນັດ
- ພັຈັນເປົ້າ ACPI (Advance Configuration and power Interface)

Life Cycle of An I/O Request



Performance

- I/O ผ่านสื่อสารและก่อผลด้านพื้นฐาน
- CPU execute driver, kernel I/O code
- จำนวน context switching
- Data Copying
- ภายนอกไฟฟ้า Network traffic

Performance Improving

- ลดจำนวน context switches
- ลดการ copy data
- ลด interrupts ระหว่าง transfer ไฟล์บน磁碟, หน่วยความจำ controller
- ใช้ DMA
- ใช้ HW Queue
- Balance Spec ให้ Total throughput น้อยที่สุด
- ให้ User-mode process ทำงาน kernel threads / daemons

File - System (Interface)

File



ມະນາຄາດຕັ້ງ

- Name
- Identifier
- Type
- Location
- Size
- Protection
- Datetime และ user identification
- ຂອບພລິໄຟຟ້າ

ກົມ່ວ່ານຫຸ້ນ
ໂທ Directory
Structure



ເປັນ node ເກີບ
ສ່ວນກາໃນກາງໄວ້

File

Operation

- Create
- Write
- Read
- Reposition
- Delete
- Truncate
- Open (F_i) ໂອງ content ຢູ່ການ disk ຫຼື memory
- Close (F_i) ຢູ່ການນັ້ນ disk

ອຸນການປິໄວ້ ຈະມີການຈົດກາດດັ່ງນີ້

ຜົນປິໄວ້ File

- open file table
- file pointer
- file open count

File Locking

- Shared Lock → ອະວິນ
- Exclusive lock → ອະເໜີຍນຸ່ມ
- Jessinosis lock
- mandatory
 - ບັນດີນ
- Advisory
 - ໄລະນີ້ນ

File Structure

- none ← word, bytes
- simple record structure ← CSV, \n \r
- complex structure ← XML, JSON

Access Method

- Sequential Access (ອຳນວຍຈຳນວນ)

- read next
- write next
- reset ← ໄກສາໃໝ່

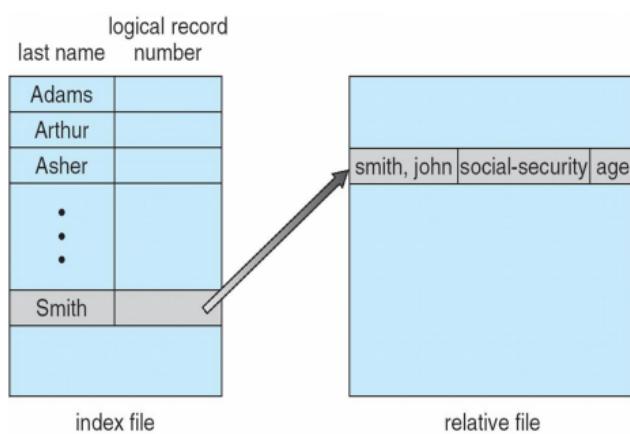
- Direct Access

- read n
- write n
- position to n
 - read next
 - write next
 - rewrite n

* n = relative block number.

ຮັບອ່ານໄຟລ໌ໄຟລ໌

- Index and Relative file



(Database?)

Disk Structure

- Disk ត្រូវបាន partitions ទៅ
- RAID
- Disk នាម FS ឬទៅក្នុងផែនក្នុងទៅ

ផ្សេងៗនៃ File System

Solaris

- tmpfs – memory-based volatile FS for fast, temporary I/O
- objfs – interface into kernel memory to get kernel symbols for debugging
- ctfs – contract file system for managing daemons
- lofs – loopback file system allows one FS to be accessed in place of another
- procfs – kernel interface to process structures
- ufs, zfs – general purpose file systems

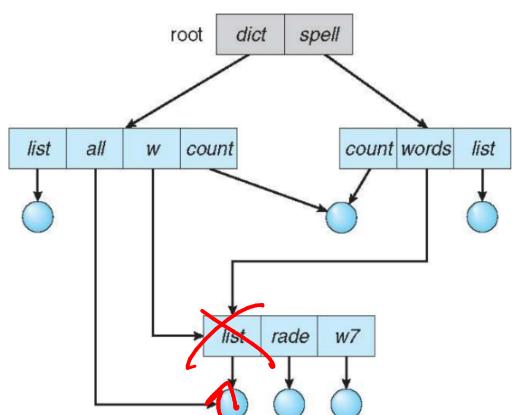
ការសំរែកការណ៍ឱ្យឲ្យនូវ directory

- រក File (find)
- តម្លៃង (create)
- លើ (delete)
- list
- rename
- traverse

ការចំនួន Directory

- single-level directory → នៅ user ឬ directory តិច
→ ដំឡើង Naming, Grouping
- Two-level directory → 1 user 1 directory
→ search ឬចុះឱ្យ
- មិថត name
- ឬ យោងឬ grouping
- Tree-Structure directory
- Acyclic-Graph Directories → ឬ file ឬជា subdirectory
ឬ shared ទំនាក់ទំនង

Acyclic - Graph Directories (เพิ่มเติม)



- ไม่ aliasing (dict/w/list
spell/words/list)

ถ้าลง list สองปัจจัย ex. ลง w/list
จะทำให้เกิด dangling pointer
แบบก่อตัวๆ กัน Back pointer
แล้วล็อ้งลง pointers 9 หน่วย

→ เก็บ directory entry type 9 หน่วย

→ Link ← ใช้ลง Shortcut ไปยังไฟล์ที่ผูก

[Resolve the Link ใช้หาน各 follow pointer
ไปยังไฟล์]

General Graph Directory

→ จะรู้ได้ยังไงว่าไม่มี cycle ใน Graph ไม่?

Allow → Link File ใจดีๆ ใจดีๆ

→ Garbage Collection

→ ผู้ดูแล link จะรัน cycle detection algo.

Current Directory ทำอะไรได้บ้าง



→ pwd

→ create / delete 2 file
touch rm

Protection

→ R W X
↑ ↑ Execute
Read Write

→ Append, delete, list

binary
3 bit

Owner

rwx
4+2+1
7

Group

r-x
4+0+1
5

Other

r-x
4+0+1
5

มาตรฐาน 777

File - System (Implementation)

• File System Structure

‣ ຖាំង secondary storage

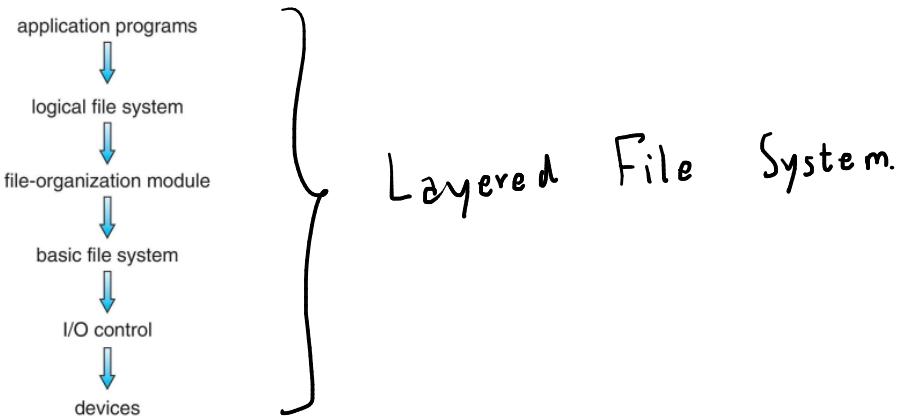
L I/O จะ perform នៃ blocks of sectors

‣ ទាយ mapping នៃវត្ថុ logical → physical

‣ ចំណាំការបង្កើតឱ្យផ្តល់ data , ការកិន និងការគ្រប់គ្រង

‣ Driver ជីថាបញ្ជូន physical device.

‣ File System មែនទទួលខាន់ខ្លួនជាលែកបង្ហាញ layer



File System Layer

‣ Device Driver. - ចំណាំការ I/O នៃ I/O control layer

‣ Basic File System - រួមការងារ: លិចស្សុករុងការណ៍ដែលត្រូវការកិនការណ៍ driver

- នគរាងនូយែងទៅការ caches, buffers

‣ File Organize Module - រួមការការងារ file , logical address
និង physical block

- រួមការងារ logical block #

↓
physical block #

- ចំណាំ free space , disk allocation

‣ Logical File System - រួមការងារ metadata ឬវិវាទ

- ឯកសារឯកសារ → file number, File handle, location
ឯកសារ control block

- ឯកសារ directory

- នរបៈ ឯកសារ (protection)

នៃប្រព័ន្ធនា File System ដែលមាន layer ចំណេះតារាងង់បច្ចនា ហេតក្នុងការងារនេះ

* Logical Layer តាមរបាយ implement តាមរាជធានីក្នុងខាងក្រោម

In OS នៃ File System នេះ

- CD-ROM នឹង ISO 9660
- UNIX នឹង UFS
- Windows នឹង NTFS
- Linux ដែលអាចទាន់បាន ext 3

File System Operation

- Boot Control Block
 - ទីតាំង volume នៃ OS ពេល boot រួចរាល់
- Volume Control Block
 - ទីតាំង volume detail → នៃ រឿងនូវ blocks , free blocks
block size , free block pointers
ឬ array.

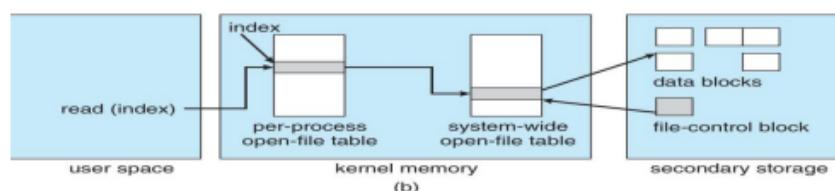
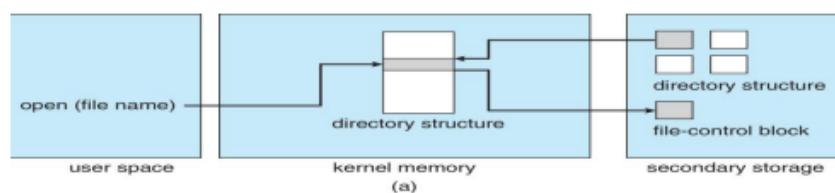
File Control Block (FCB)

- ទីតាំងរាយរបៀបឈើឯក (FCB per file)

In-Memory File System Structures

- Mount Table - ទីតាំង file system mounts, mount points, file system types.
- System-wide open-file Table
- Per-process open-file Table

- Figure 12-3(a) refers to opening a file
- Figure 12-3(b) refers to reading a file



Directory Implementation

- ▷ Linear List - รายการหนานาน (linear search)
 - ▷ Hash Table - มีการ collision เมื่อค่ามาซ้ำกัน

Allocation Method

- ▷ Contiguous (ជាកំណើនក្នុងខ្លួនគ្នា) displacement : R

- ទូរសព្ទ
 - កែបង្រាប
 - ចំណាំពុល

→ - តួនាទីដែលអាចរាយការណ៍ file នេះ

 - ចំណាំទូរសព្ទ

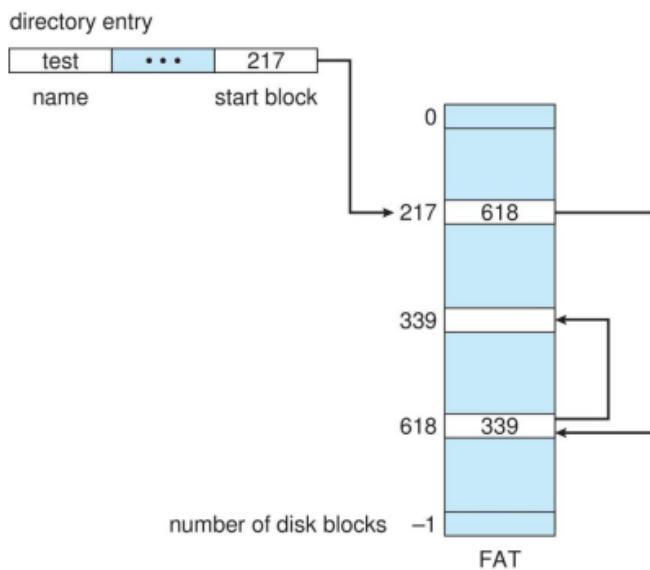
- กีด external fragmentation
ก าน 05 เอกพาร์ค รัชดา Extent-Based Systems

- ក្នុងការទូរសព្ទ នប់អំពីទូរសព្ទទាំងអស់ និងទូរសព្ទទាំងអស់ និងទូរសព្ទទាំងអស់

- Linked displacement : R + 1

- เก็บกรากษาด้วย linked-list 1/2 blocks
 - ไม่มี external fragmentation (มี internal fragmentation)
 - ไม่ต้อง block ที่มี pointer ไปยัง block อื่นๆ
 - การค้นหาทำได้เร็วมากยิ่งขึ้น (I/O, disk seek)
 - มีปัจจัยการันตี Reliability

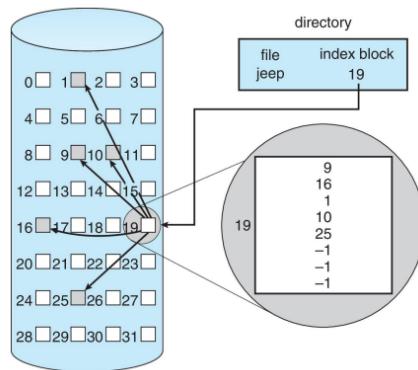
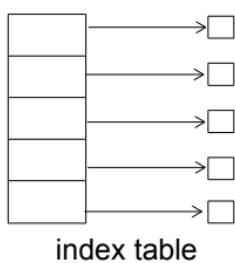
- ## → File Allocation Table (FAT)



▷ Indexed Allocation Method

- ແຕ່ລະໄຟລ໌ຈະມີ index block of pointers ທີ່ໄດ້ຈຳ data blocks

- Logical view



Performance

- Contiguous เหมาะສິນຮັບ sequential, random
- Linked เหมาะສິນຮັບ sequential
- Indexed ຈະຫຼັບຫຼວດກ່າວເພຣະວ່າ
 - ການເຫັດຖິງ block ຈະຕົວໄວ 2 index block
ເພວຍ, 1 block ສໍານົບເຄີຍ index ຂັບຂອງມີ
 - clustering ຕ່າຍສະ CPU overhead
- In NVM ຕັງເລີ່ມໃຊ້ algorithm ອີ່
- ຄ້າໃຊ້ algorithm ເກົ່າງຈາກໃໝ່ປ່ອງ CPU

Free-Space Management

- ອົງ file system ຈະມີເກີນ free-space list ອີ່ສໍາເລັບຕື່ມຕາມ
ນັ້ນທີ່ກໍ່ເນັດວ່າ
 - ↳ implement ອີ່ວ່າ bit-vector ມີວ່າ bit-map

Block Number Calculation

(number of bits per word) * (number of 0-value words) + offset of first 1 bit

example :

block size = 4KB = 2^{12} bytes

disk size = 2^{40} bytes (1 terabyte)

$n = 2^{40}/2^{12} = 2^{28}$ bits (or 32MB)

if clusters of 4 blocks -> 8MB of memory

ມີການ

contiguous file

Linked free Space List on Disk

- นาพื้นที่เปลือกต่อเนื่องๆ กัน
- ไม่ต้อง traverse ห้าม list (ถ้านั้นก็รบกวน blocks ที่ free อีก)

Free-Space Management

- Grouping - เก็บ จํานวนช่องที่ว่างต่อเนื่อง
- Counting
- Space Maps (ZFS)
 - แบบปุ๊ມ metaslab
 - ↳ record as log

Effciency and Performance

ประสิทธิภาพขึ้นอยู่กับ

- Disk allocation and directory algorithm
- Types of data
- pre-allocation
- Fixed-Size / Vary-Size data structure

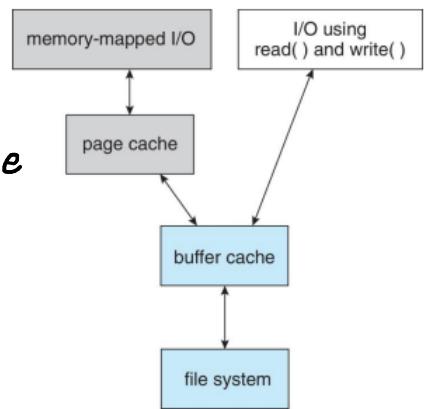
Performance

- I/O data รวม metadata ใหญ่ลงกว่า
- Buffer Cache หัก block ที่ใช้งานบ่อยๆ
- Synchronous - ขึ้นกับ CPU / OS ต้องการ , ไม่มี buffer (ใช้ disk โหลดเร็ว)
- Asynchronous - ขึ้นกับ - ร่องรอย
 - หัก buffer
- Free-behind / Read-ahead * Read มากกว่า Write

Page Cache

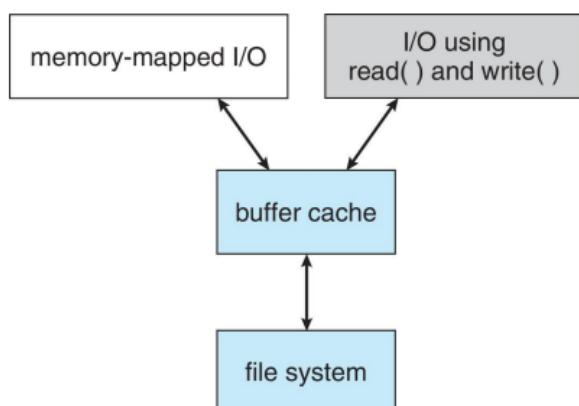
- คือ cache page ของ虚拟 memory
- memory-mapped I/O ใช้ร่วมกับ page cache
- Routine I/O ผ่าน File system หัก buffer disk cache

I/O Without a Unified Buffer Cache



Unified Buffer Cache

- វិញ្ញាន page cache នៅ = buffer cache
ដើម្បី ស្ថិតិយាជ្ញា ចូលទៅ (មេដកលីបង ឬ double buffering)



Recovery

- Consistency checking - ពួរឈាម data នៃ directory structure ក្នុង data blocks នៃ disk និង fix inconsistencies
- រាយចារ នឹងបងក្រុងកំណើង

- ធ្វើប្រព័ន្ធគ្នុង Backup នៃ restore ដើម្បី ក្លាយជាយករាយ

Log Structure File Systems (Journaling)

- ក្រុងការការងារ log

- បងក្រុង log រាយការណ៍ ឱ្យកិច្ចការក្នុងការណ៍

- នៅឯណានៃ file system រាយការណ៍ត្រូវត្រូវត្រូវ update

- ទិន្នន័យ asynchronous

- តិច file system សម្រាប់ កិច្ចការណ៍ transaction ដែលត្រូវការក្នុងមេដក

- recover ទៅ ex. WAFL, APFS, HFS+

File System Internals

File System

- Computer តាមរយៈ storage
- រូបនីទំនាក់ទំនង partitions $\xrightarrow{\text{hold}}$ volume
- volume រាយការយោង partitions ក្នុង
- នៅលើ volume ក្នុង file system និងបាន

Partitions and Mounting

- partition
 - $\xleftarrow{\text{raw}}$ = មិនជា file system
 - $\xleftarrow{\text{cooked}}$ = ជា file system តាម
- boot block \longrightarrow boot volume
 \longrightarrow boot loader
 - នឹងវិញ boot-manager ដួងរាយ OS
- Root Partition
 - នេះ OS ធ្វើការ នៅ partition នេះ
 - នេះ OS នៅក្នុង ឱ្យការការពារ
 - mounted នៅលើ boot
 - partition នៅក្នុង mount auto ឬ manual ក្នុង
- នូវចំណាំការកែវិញ file system
 - ពារាសុំ metadata រៀបចំបង្ហាញ នៃវិញ
 - ឯកសារ \rightarrow អ៊ូដ \rightarrow ឱ្យការការពារ
 - ឯកសារចំណាំ \rightarrow ឱ្យការការពារ mount table
 - ឯកសារ \rightarrow allow access

File Sharing

- រូបនីទំនាក់ទំនង user/system ដែលឱ្យ file ត្រូវការកែវិញ
- permission ត្រូវការកែវិញដោយរាយ
 - OS សំណង់ group, owner member
 - ដែលវិនិច្ឆ័យ apply នៅលើបុរីបុណ្ណោះ

Virtual File Systems

- ពីរបៀវឌីជនក្នុង file-system និងសម្រាប់ Object-Oriented
- មាន API តែងតាំងនៃ file system និងសម្រាប់ក្នុង

នៃ Linux មាន 2 object types

- inode, file, superblock, dentry

VFS មាន ក្នុង operation ពីរដែល implement

Remote File Systems

- ពីរបៀនាសម្រាប់ file នៃ network
 - FTP
 - NFS Distributed File System
 - HTTP (google drive)

Client-Server Model

- ពីរបៀនាសម្រាប់ server និង client ដើម្បី allow ប្រើប្រាស់ network protocol ដើម្បីសម្រាប់ remote file system
- ដែលបានពាយតាមពីរបៀនាដូច network ID និងពីរបៀនាទិន្នន័យ

Distributed Information Systems

- ពីរបៀនាទិន្នន័យ remote computing
aka. distributed naming services

- DNS → domain និង ip-address

- Network Information Service
 - provide
 - Username
 - password
 - user ID
 - group information

- common internet file system
 - ផ្តល់ព័ត៌មាន network info + auth
 - ពិនិត្យព័ត៌មាន network log in

- Active Directory distributed-naming service

- Kerberos - derived network auth protocol

- Lightweight directory-access protocol

Consistency Semantics

- เป็น criteria สำหรับการ evaluating file ใน sharing-FS
- เจาะเจง user หลาย คน ว่ามีการเข้าถึง shared-file อย่างไร
 - ถ้า modify ข้อมูล user อื่น ก็จะไม่สามารถ
 - ผู้ที่เข้าไปในไฟล์ atomic
- จัดเรียง process ตาม session
 - open/close

UNIX

- อนุญาตให้มีการเขียนไฟล์ที่เปิดอยู่
- share pointer ณ I/O location ของ file
- physical image ร่วมกัน access ไม่ exclusive
แต่ทำให้เกิด process delay

Session Semantic (Open AFS)

- บนเซิร์ฟเวอร์จะมีบันทึก session ของผู้ใช้งาน
- ตรวจสอบ copy ได้

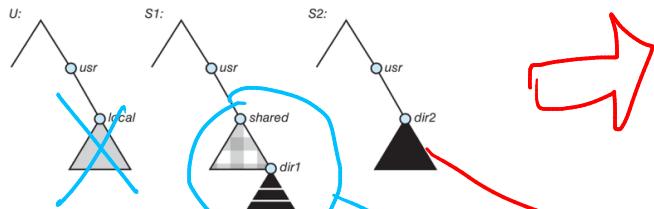
Sun Network File System (NFS)

- เป็น file system คำสั่งเข้าถึง file ผ่าน LAN ผ่านโปรโตคอล TCP, UDP

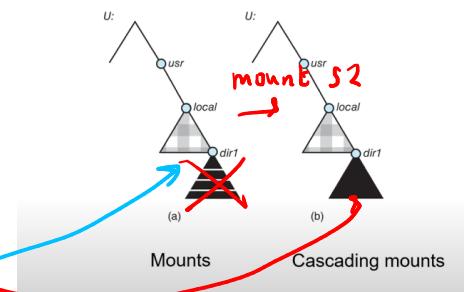
- remote directory mount ที่ local directory
- รูป non-transparent ต้องมี hostname ก่อน
ถึงจะเข้าถึงแบบ transparent ได้

NFS - สามารถ bridge ระหว่างที่ตั้งกันได้

▪ Three independent file systems



▪ Mounts and cascading mounts



NFS mount protocol

requirement: - ต้อง directory service server-client

- mount request เป็น RPC

- เมื่อ request แล้ว server จะนำ key ที่มีมาบันทึก

- server ไม่เปลี่ยนแปลง

ใน NFS protocol จะมี คำสั่งมาในตานี้

- search

- read

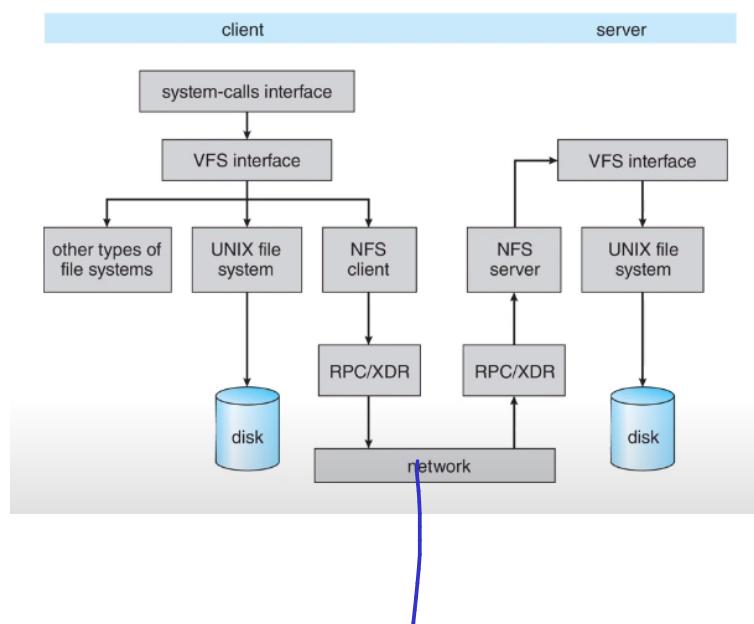
- link

- ตรวจสอบ file attributes

- R/W

* คือ stateless ต้องใช้ argument ทุกๆ ครั้ง

(คือ request ทุกครั้ง)



มีการทำ path translation

Security

信息安全 Violation (สิ่งที่ hacker ทำมุ่งเป้าก็จะนี้)

- Breach of confidentiality - unauthorized read
- Breach of integrity - unauthorized modify
- Breach of availability - unauthorized delete
- Theft of service - unauthorized use of resources
- Denial of service - prevention of legitimate use

วิธีการโจมตี

- Masquerading - ปลอมตัวเป็น user เพื่อเข้าถึงระบบ
- Replay Attack - การดักเร็บว่างานเพื่อเอาข้อมูล
- Man-in-the-middle - ปลอมเป็นผู้ร่วมกลาง
- session hijacking - ขโมย session ที่เข้าสู่ระบบแล้วไม่ต้อง login
ex. ใช้ cookie สำหรับ login
- privilege escalation - ยกระดับ permission ให้สูงขึ้นเรื่อยๆ

Security Measure Levels

attack	Application	prevention
bug, design flaws, ข้อบกพร่อง	patch code	sandboxing, การจำกัดสิทธิ์
insecure defaults, ช่องโหว่	Operating System	patches, reconfig, hardening
sniffing, spoofing, masquerading	Network	encryption, auth, filtering
console access, HW-based attack	Physical	guard, vaults, device data encryption

Program Threats

- example : Trojan Horse - code ที่อยู่ใน environment ไม่ควร
 - ปลอมตัวเป็น user แล้ว run program ใดๆ
 - spyware, pop-up browser windows, convert channels
 - 80% มากกว่า คนที่ติด spyware นั้นหลวม
- Trap Door - ซ่อนใน compiler
 - เก็บใน user เฉพาะ
- Malware - โปรแกรมที่ เปิดช่องโหว่, ปิด หรือสร้างความเสียหาย
 ต่อ computer
- Spyware - program ที่จะติดลงกับ software ปกติ
 เพื่อแสดง ads หรือ เก็บข้อมูลผู้ใช้
- Ransomware - lock ข้อมูล แล้วเรียกเงินเพื่อปลด lock

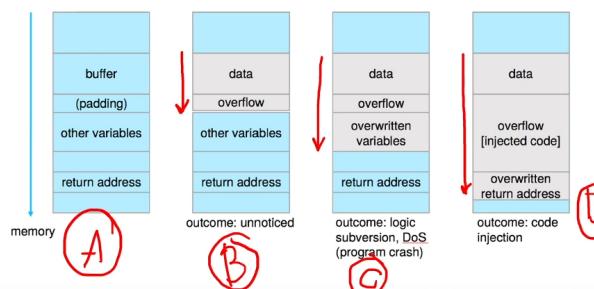
* ทุก program threats จะพยายาม violate ด้วย Least Privilege

Goal : ยุติการเข้ารหัสผ่านไวไฟ Remote Access Tool เพื่อโจมตีซึ่ง

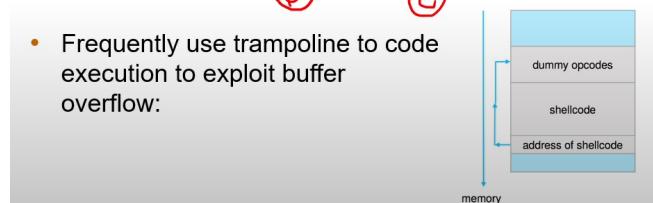
Code Injection

- พนักงาน C, C++ ที่ buffer overflow attack
 เพื่อใส่ code ที่ต้องการลงไว

- Outcomes from code injection



- Frequently use trampoline to code execution to exploit buffer overflow:



- การเขียนโปรแกรมที่ดีก็เป็นส่วนหนึ่ง 例
 - script kiddies ← เก็บ script ที่สามารถไปดูของคนอื่น
 - attack ชุด อาจเข้าถึง shell หรือเป็น network port
delete file
หรือ download files. etc.

- buffer overflow สามารถแก้ไขเอง - ปิด stack execution
หรือ เพิ่ม bit ใหม่ page-table
เพื่อเพิ่มส่วนที่ non-executable
(msi, NX, DEP)

ตัวอย่าง Program Threat (เพิ่มเติม)

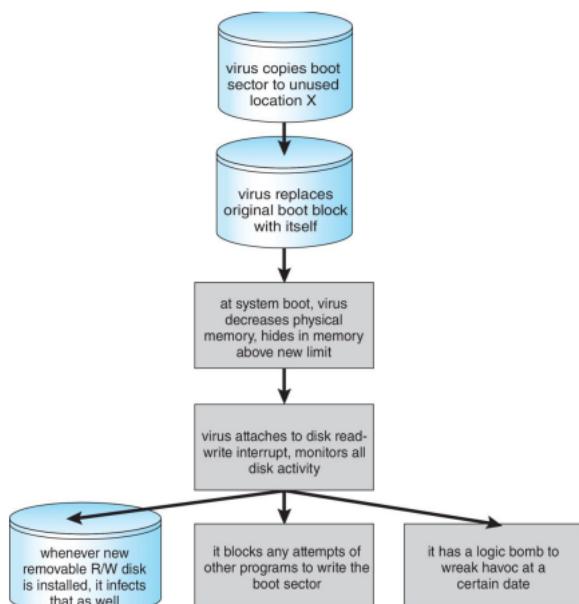
- Viruses

- มากกว่ากัน โปรแกรมปกติ
- สามารถ แพร่กระจายต่อๆ กัน
- ผู้ใช้งานส่วนใหญ่ program, OS, หรือ application
- มักมากับ email หรือเป็น macro
ex. VBS

- Virus Dropper

- ตั้งปลุย virus ลงในระบบ (อาจเป็นนักพน)

ex. Boot-Sector computer virus



System and Network Threats

- worms → វិធានការប្លង់

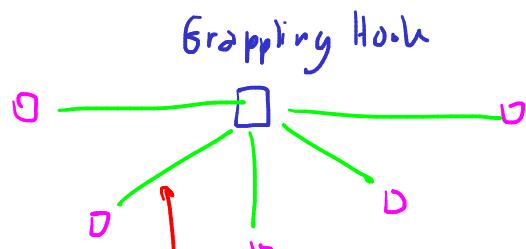
ex. internet worm

- វិធានការប្លង់ Unix remote access

នេះជា bug finger ឬ sendmail program

- វិធានការប្លង់ rsh ដែលការសម្រាប់គ្រប់គ្រងព័ត៌មាន

- Grapping Hook រួមជាមុន ផលកខែងក្រុងកិច្ចការណ៍



ក្នុងការពេងចាយនៃបានស្តាប់ password guessing

- Port Scanning

- ការពេងចាយព័ត៌មាន ឬ IP ឬ ឬ IP អីដែលមាន IP

- វិធានការសម្រេច protocol នូវ client ទៅវិញ

- ព្រមទាំង OS ផ្លូវ

ex. nmap ← វិធានការពេងចាយ របៀប

nessus ← ឱ្យសំរាប់ bug ឬកើតឡើង

- មកការប្រើប្រាស់ zombie systems.

- Denial of Service

Distributed

- វិធានការប្រាក់បង្ហាញ ឬការបង្ហាញ

- ឯកសារការប្រាក់បង្ហាញ ឬការបង្ហាញ (DDoS)

Cryptography

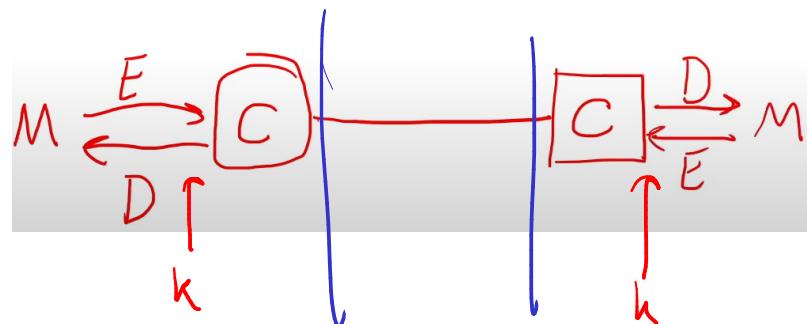
- เป็นการเข้ารหัสระหว่าง senders \longleftrightarrow receivers ของข้อมูล
- ใช้ keys
 - ↓ นำไปรู้
 - ▷ 用于 confirm ความถูกต้อง
 - ▷ เพื่อความน่าเชื่อถือของ Sender และ Receiver

Encryption

- เข้ารหัส แปลง plainText \rightarrow ciphertext

$$\begin{aligned} E : K &\rightarrow (M \rightarrow C) \\ D : K &\rightarrow (C \rightarrow M) \end{aligned}$$

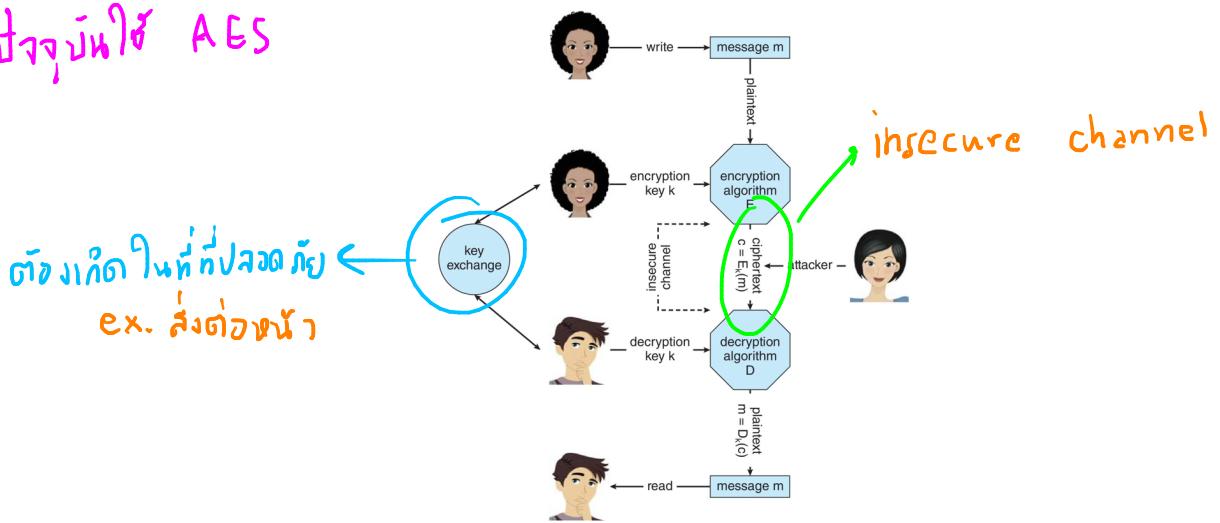
เป็น function ที่มีประสิทธิภาพ



Symmetric Encryption

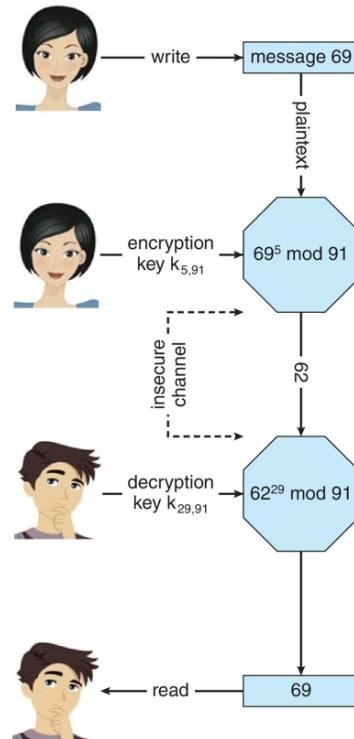
- ใช้ key เดียวกันทั้ง 2 ฝ่าย เช่น RSA

- ปัจจุบันใช้ AES



Asymmetric Encryption

public key - ດາວໂຫຼນເກີດ
private key - ດາວໂຫຼນສ່ວນເກີດ ← ລັບອົງກະຕິ
ex. RSA



Authentication

$S : k \rightarrow (M \rightarrow A)$ ກົດລົງໃຫຍ່ generate message auth.

$V : k \rightarrow (M \times A \rightarrow \{\text{true}, \text{false}\})$ ກົດຍືນຍຸ້ນ

> ກໍານົດລົງໄດ້ message ທີ່ສົມນາ ມີ k ອີ່ມ key ຖໍ່
ແລະ ດີວີ່ key ດາວໂຫຼນ ດີວີ່ກຳລັງກຳນົດ

Hash Function

- basic of auth.

m ຍາມໄດ້ໄວ້ $\text{hash}(m)$ fixed size

$m \rightarrow m'$ $\text{hash}(m') \neq \text{hash}(m)$ *ສະຫຼິບ collision *

ex. SHA-1 md5

MAC (Message-authentication code)

q.v Symmetric

cryptographic

generate

checksum in secret-key

Digital Signature

q.v Asymmetric

Authentication > Encryption

- ធ្វើនាំរុញការចិត្តមួយ
- ធ្វើនាំរុញការអនុវត្ត commitment យ៉ាវ

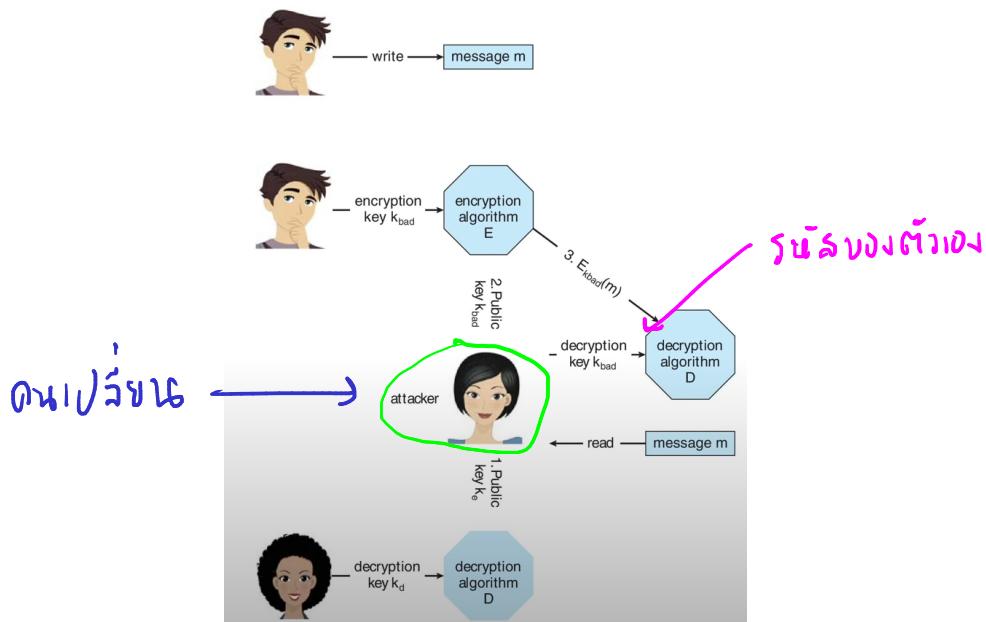
កែវគ្នាក់ key

- Symmetric key - សំណូល out-of band
- Asymmetric key - សំណូល key-ring

Digital Certificate

- រូបិយាយរាល់ទៅជាបីទី public key និងឈ្មោះអ្នក

Certificate Authority → included in browser



Man in a middle attack on
symmetric cryptography

ms Implement Cryptography

- ห้อง layer OS I 6 (Transport Layer)

ex. TLS

User Authentication

- password ที่ดี
 - เปลี่ยนบ่อยๆ
 - เครื่องไว้
- จะ encrypt  /etc/shadow
บังคับมีการใส่ salt
ทำให้ hash ต่างกัน

- One-time password

ex. TOTP

- biometric

- multifactor Authentication

- หลายชั้น ex Password + USB

Implement Security Defenses

- Defense in depth ex. multiple layers of security

- Security Policy ← Policy ของผู้ดูแล

- Intrusion detection (IDS)

- signature-based

- Anomaly detection → zero-day

- False-positive, False-negative

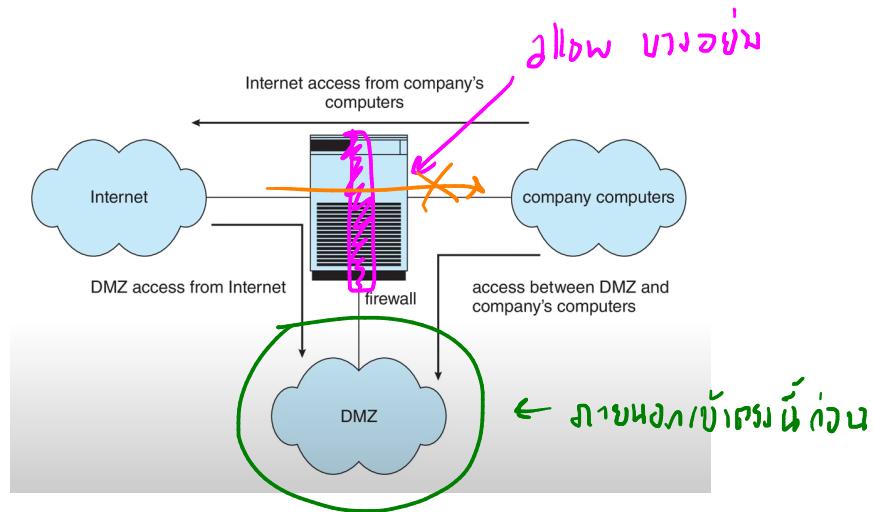
- Virus Problem → จำลอง Sandboxing

- Auditing, Accounting, Logging

- Safe Computing

Firewall

- firewall
- tunneling
- personal firewall
- Application proxy firewall
- System-call firewall



Security Defense

- not educate → safe computing
 - prevent phising
 - block service ที่ไม่ต้องการ
 - block surface attack