

สรุปแนวข้อสอบ OS 59

มีทั้งหมด 11 ข้อ คะแนนเต็ม 135 คะแนน

1. Main Memory -> จัดสรร Process เข้าใช้หน่วยความจำ ตาม Algorithm มี 3 ข้อย่อย 3 แบบ

First-Fit, Best-Fit, Worst-Fit ข้อละ 5 รวม 15 คะแนน วิธีทำตามนี้เลย By เทพดีหิด

15 คะแนน

MAIN MEMORY

Dynamic Storage - Allocation Problem

มี 3 แผน 1. First-Fit จัดสรร hole แรกที่ใหญ่เพียงพอ ถัดไป partition แรกที่เจอแล้วขนาดในหน่วยที่จะเดินขบวนไม่ได้แล้ว

2. Best-fit จัดสรร hole ที่พอดีที่สุด ถัดไป จะใช้ partition ที่มันน้อยตามจำนวนที่ขบวน

3. Worst-fit จัดสรร hole ใหญ่ที่สุด ถัดไป จะใช้ partition ที่มันน้อยตามจำนวนที่ขบวน

ตัวอย่าง
 ส่วนหน่วยตามจำ 6 partition คือ 300 KB, 600 KB, 350 KB, 200 KB, 750 KB และ 125 KB

ทำการจัดสรรใช้พื้นที่ 115 KB, 500 KB, 358 KB, 200 KB, 375 KB จาก process size

P1 P2 P3 P4 P5

First-fit

first-fit มีพื้นที่ partition ที่ยังไม่ถูกใช้งาน 325 KB และพื้นที่เหลือจากการใช้งาน 452 KB

Best-fit

best-fit มีพื้นที่ partition ที่ยังไม่ถูกใช้งาน 650 KB และพื้นที่เหลือจากการใช้งาน 127 KB

Worst-fit

ไม่สามารถจัดสรร hole ใหญ่เกิน process P5 ได้ เนื่องจาก พื้นที่ของแต่ละ partition ไม่เพียงพอต่อ

worst-fit มีพื้นที่ partition ที่ยังไม่ถูกใช้งาน 625 KB และพื้นที่เหลือจากการใช้งาน 527 KB

2. Main Memory -> คำนวณ หาจำนวน bit ที่ Logical และ Physical Memory ใช้ทั้งหมด

Logical นั้นจะแบ่งเป็นชั้นๆ เรียกว่า Page เช่นเดียวกัน Physical แบ่งเป็นชั้นๆเหมือนกัน แต่จะเรียกว่า Frame และในแต่ละ Page, Frame ก็จะมีขนาดของมัน ซึ่งขนาดจะเท่ากัน แต่ตำแหน่งอาจไม่ใช่ตำแหน่งเดียวกัน ซึ่งวิธีหาจำนวน bit ที่ใช้ทั้งหมดใน Logical ก็คือ เอา bit ที่ใช้ทำ Page + bit ที่ขนาดของ Page ใช้ และเช่นกันกับ Physical คือ เอา bit ที่ใช้ทำ Frame + bit ที่ขนาดของ Frame ใช้ ตัวอย่างตามการบ้านเลย ข้อละ 3 รวม 6 คะแนน

Consider a logical address space of 256 pages with a 4-KB page size, mapped onto a physical memory of 64 frames.

- How many bits are required in the logical address?
- How many bits are required in the physical address?

a. จากโจทย์บอกว่าที่ logical address มีทั้งหมด 256 pages และในแต่ละ pages มี 4 KB

4 KB = $4 \times 1024 = 4096$ byte แสดงว่าจะใช้จำนวน bit = $\log_2(4096) = 12$ bit

และ 256 pages จะใช้จำนวน bit = $\log_2(256) = 8$ bit

ดังนั้น จำนวน bit ที่ logical address ต้องใช้ทั้งหมดคือ $12+8 = 20$ bit

B. จากโจทย์บอกว่าที่ physical address มีทั้งหมด 64 frames และในแต่ละ frames มี 4 KB

4 KB = $4 \times 1024 = 4096$ byte แสดงว่าจะใช้จำนวน bit = $\log_2(4096) = 12$ bit

และ 64 frame จะใช้จำนวน bit = $\log_2(64) = 6$ bit

ดังนั้น จำนวน bit ที่ ~~logical~~ **Physical** address ต้องใช้ทั้งหมดคือ $12+6 = 18$ bit

แต่เราต้องถอด \log_2 ซึ่งใช้เครื่องคิดเลขไม่ได้ ก็ต้องเอาเลขที่ต้องการถอด มาหาร 2 เอา
หารจนเหลือ 1 แล้วก็ค่อยนับเอาว่าหาร 2 ไปกี่ครั้ง ก็จะได้คำตอบ หรือจำแบบนี้ไปเลย
เราจะรู้ว่า 1-KB = 1024 คือ 2^{10} แล้วจะไล่ขึ้นไปทีละ คูณ 2 KB จะได้ดังนี้

2-KB = $2^{11} = 11$ bit , 4-KB = $2^{12} = 12$ bit , 8-KB = $2^{13} = 13$ bit

16-KB = $2^{14} = 14$ bit , 32-KB = $2^{15} = 15$ bit , 64-KB = $2^{16} = 16$ bit

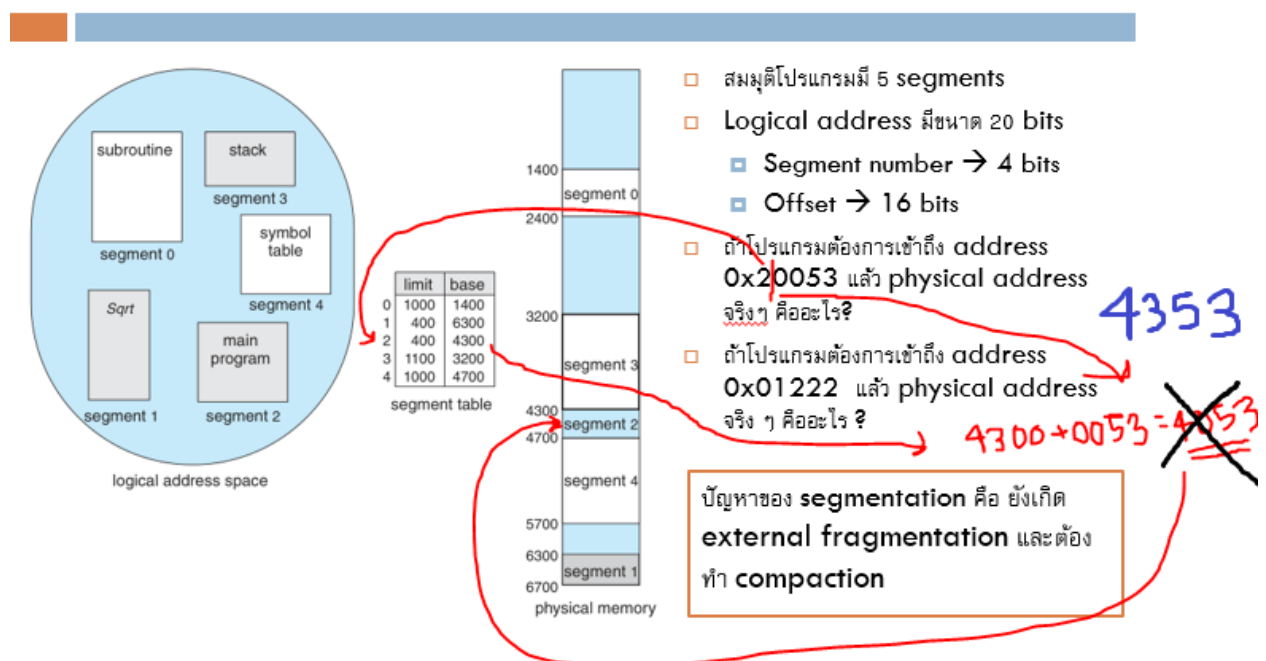
128-KB = $2^{17} = 17$ bit , 256-KB = $2^{18} = 18$ bit , 512-KB = $2^{19} = 19$ bit

และ 1-MB ก็คือ $2^{20} = 20$ bit

3. Main Memory -> ข้อนี้จะให้ตำแหน่งที่ต้องการเข้าถึง แล้วให้ไปหาว่า อยู่ตรงไหนของ Physical Memory มีอยู่ 2 แบบ น่าจะออกแบบแรก แต่ก็ดูแบบสองเพื่อไว้ 4 คะแนน

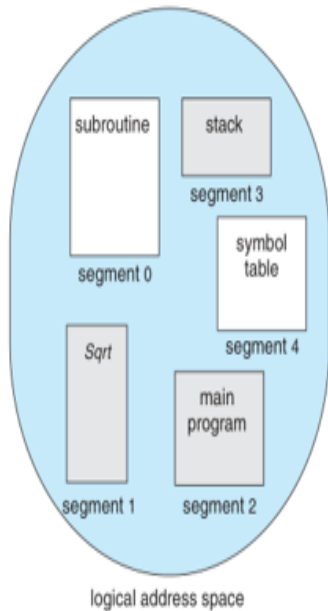
3.1 แบบ Segment แบบนี้จะแบ่ง Logical เป็น Segment โดยจะมีตารางบอก base และ limit ซึ่ง base คือ ตำแหน่งเริ่มต้น Segment และ limit คือ ความยาวของ Segment แบบนี้โจทย์จะให้ตำแหน่งที่ต้องการเข้าถึงมา ตามตัวอย่างนี้เลย

ตัวอย่าง : Segmentation



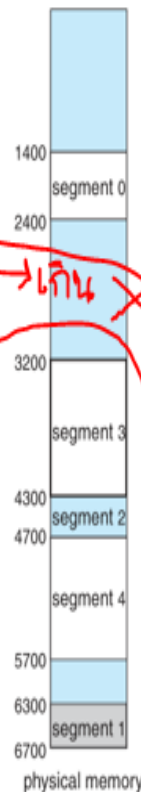
รูปนี้โจทย์ให้ $0x20053$ มา โฟกัสแค่ 20053 บิตแรกจะบอกลำดับของ Segment ที่ต้องการเข้าถึง คือ 2 แล้วบิตที่เหลือ 0053 ให้เขาไปบวกกับ base ในตาราง Segment ลำดับที่ 2 จะได้ 4353 ตามภาพ เราก็จะได้ตำแหน่งใน Physical ที่เข้ามาจาก $0x20053$ คือ 4353

ตัวอย่าง : Segmentation



	limit	base
0	1000	1400
1	400	6300
2	400	4300
3	1100	3200
4	1000	4700

segment table



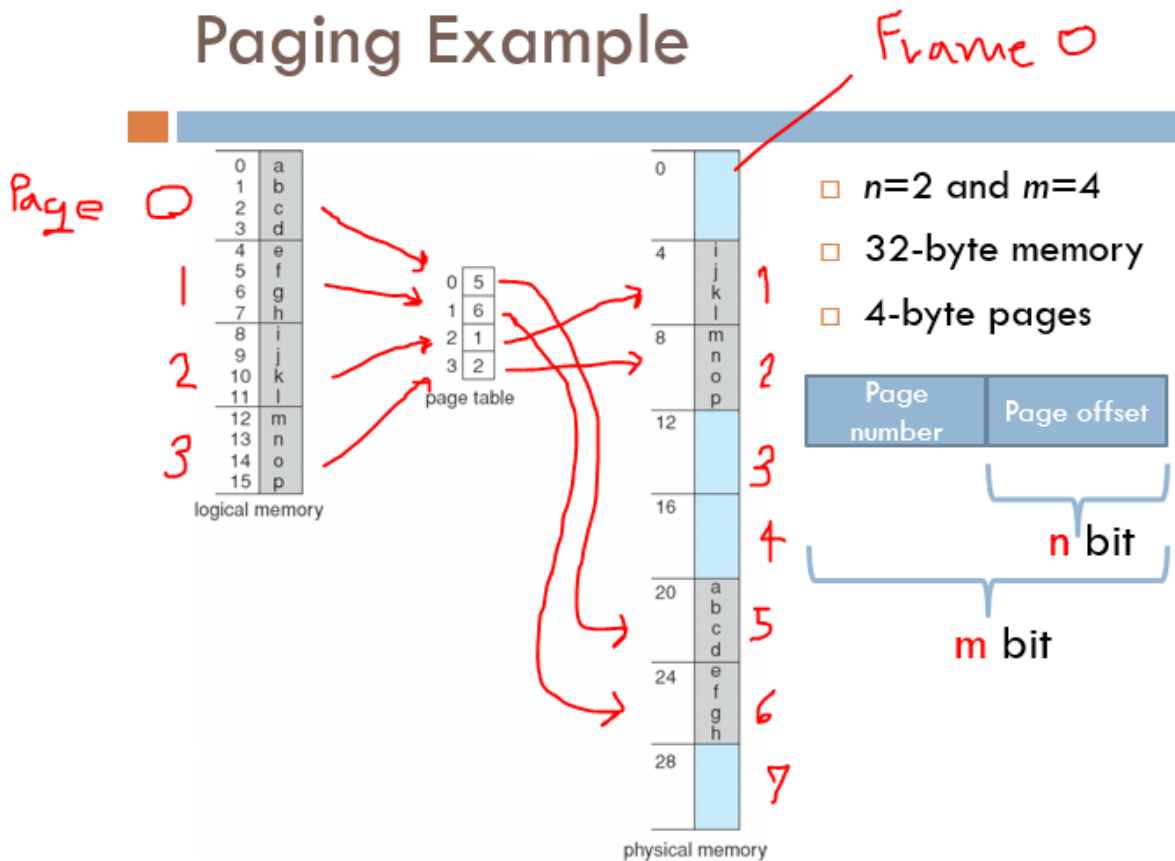
- สมมติโปรแกรมมี 5 segments
- Logical address มีขนาด 20 bits
 - Segment number → 4 bits
 - Offset → 16 bits
- ถ้าโปรแกรมต้องการเข้าถึง address 0x20053 แล้ว physical address จริงๆ คืออะไร?
- ถ้าโปรแกรมต้องการเข้าถึง address 0x01222 แล้ว physical address จริงๆ คืออะไร?

$1400 + 1222 = 2622$

ปัญหาของ segmentation คือ ยังเกิด external fragmentation และต้องทำ compaction

รูปนี้จะเข้าไม่ถึง address ใน physical เพราะตำแหน่งที่โจทย์ให้มาคือ 01222 ซึ่ง Segment ที่ 0 base คือ 1400 เอาไปบวกกับ 1222 จะได้ 2622 ซึ่งตำแหน่งนี้จะเกิน limit ที่ Segment 0 อยู่ เพราะตำแหน่งที่เข้าถึงได้สูงสุดคือ $base + limit = 1400 + 1000 = 2400$ ซึ่ง 2622 เกิน 2400 หรือดูง่ายๆ คือเอาตำแหน่งที่จะเข้าถึง 1222 ไปเทียบกับ limit เลย ถ้ามากกว่า จะเกิด external fragmentation นั่นคือ mem มีพื้นที่พอก็จริง แต่ไม่ได้อยู่ติดกันเลยไม่สามารถให้บริการได้

3.2 แบบ page ถ้าออกแบบนี้ห้วนเลย



จากรูปเราแค่ไปดูในตาราง page แล้วดูว่า address ของ page แต่ละตัวอยู่ใน frame ไหน เช่น address แรก คือ page 0 โดยตารางระบุเลข 5 นั้นหมายความว่า มันชี้ไปที่ frame ที่ 5 ของ physical โดยอยู่ address ที่ 20 ใน physical นั่นเอง

4. Virtual Memory -> คำนวณ Page Fault ใส่ข้อมูลลงตาราง ตาม Algorithm มีทั้งหมด 3 แบบ
FIFO , LRU และ Optimal ข้อละ 7 คะแนน รวม 21 คะแนน ออกตามการบ้าน

กำหนดให้ Reference String คือ

1, 2, 3, 4, 2, 1, 5, 6, 2, 1, 2, 3, 7, 6, 3, 2, 1, 2, 3, 6

จงเขียนวิธีทำในการคำนวณจำนวน page faults ที่เกิดขึ้น จาก replacement

5.FIFO replacement 5 page

1	1	1	1			1	6		6	6	6	6							
	2	2	2			2	2		1	1	1	1							
		3	3			3	3		3	2	2	2							
			4			4	4		4	4	3	3							
						5	5		5	5	5	7							

เกิด page fault ทั้งหมด 10 ครั้ง

10.LRU replacement 5 page

1	1	1	1			1	1			1	1								
	2	2	2			2	2			2	2								
		3	3			3	6			6	6								
			4			4	4			3	3								
						5	5			5	7								

เกิด page fault ทั้งหมด 8 ครั้ง

15.Optimal replacement 5 page

1	1	1	1			1	1			1									
	2	2	2			2	2			2									
		3	3			3	3			3									
			4			4	6			6									
						5	5			7									

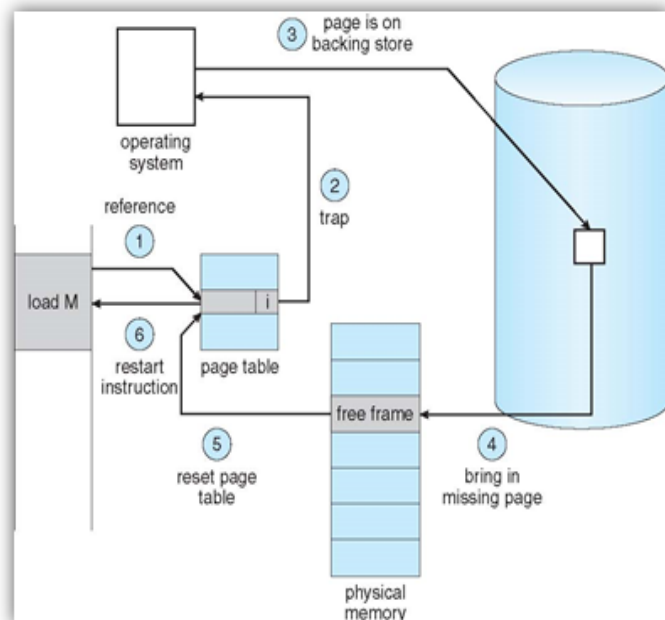
เกิด page fault ทั้งหมด 7 ครั้ง

จะเห็นว่า FIFO เกิด page fault เยอะสุด เพราะใช้หลักการเข้าก่อนออกก่อนไม่ค่อยดี รองมา LRU จะมองย้อนหลังว่าตัวไหนใกล้สุดก็จะเอาออก และ Optimal ดีที่สุด โดยจะมองไปข้างหน้าตัวไหนใกล้สุดเอาออก แต่ในปัจจุบัน ดีที่สุดคือ LRU เพราะ Optimal ยังทำไม่ได้ เพราะเราไม่รู้อนาคต

5. Virtual Memory -> อธิบายในซีท บอกมาแค่หัวข้อ ไม่ได้บอกว่าให้อธิบายตรงไหน คิดว่าน่าจะออกภาพนี้เลย 5 คะแนน โดยให้ภาพมา แล้วให้อธิบายแต่ละขั้นตอน ไม่งั้นอาจจะถามง่ายๆ 2 คำถาม 2 กับ 3 คะแนน รวม 5 คะแนน

Page Fault

- ถ้ามีการอ้างอิงไปถึง **page**, การอ้างอิงครั้งแรกจะถูกดักโดย **OS** เรียกว่า **page fault**
- 1. เมื่อชุดคำสั่งมีการอ้างอิง **page** ตรวจสอบ **internal table** ปกติอยู่ใน **PCB**
- 2. **Operating system** จะดักและดูใน **table** เพื่อตัดสินใจ:
 - ถ้าการอ้างอิง **Invalid** \Rightarrow **abort**
- 3. ถ้าไม่ได้อยู่ในหน่วยความจำไปหาใน **disk** และหา **free frame** ในหน่วยความจำ
- 4. **Swap page** ไปยัง **frame** ด้วย **scheduled disk operation**
- 5. เปลี่ยนข้อมูลใน **tables** เพื่อระบุว่าตอนนี้ **page** ได้อยู่ในหน่วยความจำแล้ว : ตั้งค่า **validation bit = v**
- 6. เริ่มต้นคำสั่งที่ทำให้ **page fault** ใหม่อีกครั้ง



ภาพนี้เป็นขั้นตอนการเข้าถึง page แต่ page ไม่ได้อยู่ใน memory จึงเกิด page fault และเมื่อเกิด OS ก็จะไปดึงใน disk โหลดเข้าสู่ memory ตามกระบวนการดังนี้

1. Reference อ้างถึง page ที่ต้องการเข้าถึง แล้วตรวจสอบในตาราง page
2. Trap ตัดสินใจจากการตรวจสอบ ในตาราง page จะมีสถานะของ page คือ v และ i ซึ่ง V คืออยู่ใน mem แล้ว i ไม่ได้อยู่ใน mem โดยจะ invalid ถ้าสถานะเป็น i แล้วจะ abort คำสั่งเลย และคำสั่งที่เฟลไปนี้ ก็คือ page fault

3. Page is on backing store ไปหาใน disk เมื่อเจอแล้ว จะหา free frame คือหาว่าเฟรมไหนที่ว่างอยู่
4. Bring in missing page โหลด page จาก disk เข้าสู่ frame ที่อยู่ใน mem
5. Reset page table เปลี่ยนสถานะ page ที่อ้างอิงเป็น v
6. Restart instruction เริ่มคำสั่งอ้างอิง page นี้อีกครั้ง เพราะครั้งแรกเฟลไปแล้ว ทีนี้ก็จะไม่เกิด page fault แล้ว เพราะโหลดมาเรียบร้อยแล้ว

แต่ถ้าไม่ใช่ภาพนี้ ก็น่าจะถามคำถามนี้แน่นอน ทำไมต้องมี page ?

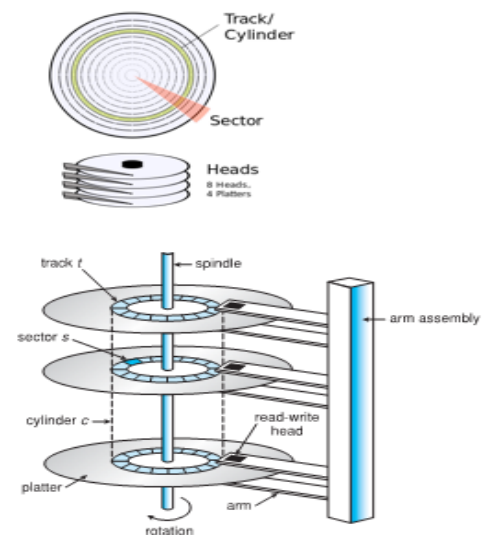
เพราะเราหาพื้นที่เยอะๆ ให้ process ไม่ได้ เลยต้องซอยๆ เป็น page เพื่อรองรับให้ใช้พื้นที่ได้โอเค

และอีกข้อก็คงไม่หนีไปจากในซีท

6. Mass-Storage System -> ข้อนี้แค่ 3 คะแนน อธิบาย ไม่บอกถามตรงไหน คิดว่าถามภาพนี้

Overview of Mass Storage Structure

- **Magnetic disks** เป็นตัวให้บริการ secondary storage ขนาดใหญ่สำหรับ modern computers
 - Drives จะหมุนด้วยความเร็วประมาณ 60 - 250 รอบ ต่อวินาที
 - **Transfer rate** คืออัตราที่ data เคลื่อนย้ายระหว่าง drive และ computer
 - **Positioning time (random-access time)** คือเวลาที่เลื่อนแขนตัวอ่านของ disk ไปยัง track/cylinder ที่เก็บข้อมูล (**seek time**) และรอเวลาที่ sector หมุนมาถึงหัวอ่าน (**rotational latency**)
 - **Head crash** คือผลลัพธ์ที่เกิดจากหัวอ่าน disk สัมผัสกับพื้นผิว disk -- แย่
- Disks สามารถเคลื่อนย้ายได้
- Drive เชื่อมต่อกับ computer ผ่านทาง **I/O bus**
 - Bus มีหลายประเภท เช่น **EIDE, ATA, SATA, USB, Fibre Channel, SCSI, SAS, Firewire**
 - **Host controller** ใน computer ใช้ bus เพื่อติดต่อสื่อสารกับ **disk controller** ที่ถูกสร้างอยู่ภายใน disk หรือ storage array



อาจจะถามพวก transfer rate , seek time , rotational latency หรือให้วาดภาพโครงสร้างคร่าวๆ ตามนี้ แต่ถ้าไม่ใช่ ก็อย่าลืมอ่านในซีทเสริมไปด้วยได้

7. Mass-Storage System -> คำนวณระยะทาง จาก Cylinder ทั้งหมด ที่หัวอ่านขยับไปมา โดยการเคลื่อนของหัวอ่าน ไปหา Cylinder นั้นจะมี Algorithm ในการเคลื่อนที่ทั้งหมด 6 แบบ 1. FCFS จะไปตามลำดับใน queue 2. SSTF จะไปหาตัวที่อยู่ใกล้ที่สุด 3. SCAN จะไปตามทิศของ Cylinder โดยจะไปจนสุด Cylinder แล้วย้อนกลับมาโดยตอนที่ย้อนกลับจะ คิดตัวที่อยู่ระหว่างทาง ด้วย 4. LOOK คล้าย SCAN แต่จะไม่ไปจนสุด Cylinder จะไปถึงแค่ตัวที่ร้องขอตัวสุดท้าย 5. C-SCAN จะไปตามทิศของ Cylinder โดยจะไปจนสุด Cylinder แล้วย้อนกลับมา Cylinder แรก โดยตอนที่ย้อนกลับจะ ไม่คิดตัวที่อยู่ระหว่างทาง 6. C-LOOK คล้าย C-SCAN แต่จะไม่ไปจนสุด Cylinder จะไปถึงแค่ตัวที่ร้องขอตัวสุดท้าย ข้อละ 4 คะแนน รวม 24 คะแนน เป็นข้อที่คะแนนเยอะที่สุด พอไล่ตามการบ้าน และเขียนวิธีทำแบบนี้เลย โดยการบ้านนี้ ทิศของ Cylinder จะไปจากซ้ายไปขวา ดูจากโจทย์ที่บอกว่า ขณะนี้อยู่ 210 ซึ่งมาจาก 180

กำหนดให้ disk drive มี 500 cylinder คือหมายเลข 0 จนถึง 499 ถ้าในขณะนี้หัวอ่านกำลังอยู่ที่ cylinder หมายเลข 210 จากก่อนหน้านี้อยู่ที่ cylinder หมายเลข 180

และมี queue ของการร้องขอไปยัง cylinder ต่างๆ ตามลำดับดังนี้
200, 120, 300, 50, 160, 100, 30, 450, 350, 150

จงหาระยะทาง (จำนวน cylinder) ที่ทั้งหมดที่หัวอ่านต้องขยับ ถ้าใช้ disk scheduling แบบ

- 1) FCFS
- 2) SSTF
- 3) SCAN
- 4) LOOK
- 5) C-SCAN
- 6) C-LOOK

1. FCFS : $|210 - 200| + |200 - 120| + |120 - 300| + |300 - 50| + |50 - 160| + |160 - 100| + |100 - 30| + |30 - 450| + |450 - 350| + |350 - 150| = 1480$

2. SSTF : $|210 - 200| + |200 - 160| + |160 - 150| + |150 - 120| + |120 - 100| + |100 - 50| + |50 - 30| + |30 - 300| + |300 - 350| + |350 - 450| = 600$

3. SCAN : $|210 - 300| + |300 - 350| + |350 - 450| + |450 - 499| + |499 - 200| + |200 - 160| + |160 - 150| + |150 - 120| + |120 - 100| + |100 - 50| + |50 - 30| = 758$

4. LOOK : $|210 - 300| + |300 - 350| + |350 - 450| + |450 - 200| + |200 - 160| + |160 - 150| + |150 - 120| + |120 - 100| + |100 - 50| + |50 - 30| = 660$

5. C-SCAN : $|210 - 300| + |300 - 350| + |350 - 450| + |450 - 499| + |499 - 0| + |0 - 30| + |30 - 50| + |50 - 100| + |100 - 120| + |120 - 150| + |150 - 160| + |160 - 200| = 988$

6. C-LOOK : $|210 - 300| + |300 - 350| + |350 - 450| + |450 - 30| + |30 - 50| + |50 - 100| + |100 - 120| + |120 - 150| + |150 - 160| + |160 - 200| = 830|$

8. Mass-Storage System -> ต่อยอดจาก com arc อีกแล้ว ข้อนี้ออก RAID ถาม-ตอบ ประมาณว่า จะมี HDD อยู่เท่านั้น แล้วถ้าทำ RAID นั้นนี้ แล้วจะสามารถกู้ข้อมูลได้ที่ลูก ทั้งหมด 12 คะแนน

RAID 0 ใช้ 2 ตัวขึ้นไป แบ่งข้อมูลไปใน HDD แต่ละตัวช่วยกันเก็บ แน่นอนว่าจะอ่านและเขียนข้อมูลเร็วขึ้น เพราะมี 2 ตัวช่วยกันอ่าน/เขียน เก็บข้อมูลได้เท่าทุน เช่น ใช้ HDD 5 ตัว ตัวละ 500 GB ทำ RAID 0 ก็จะได้เก็บข้อมูลได้ 2.5 TB และอ่าน/เขียนได้เร็วขึ้น 5 เท่า แต่ข้อเสียคือ ไม่สามารถกู้ข้อมูลได้ เสียก็เสียเลย

RAID 1 ใช้ 2 ตัวขึ้นไป จะสำรองข้อมูลไว้ HDD อีกตัว เขียนความเร็วเท่าเดิม เพราะต้องเขียนทีละตัวอยู่ดี แต่จะอ่านเร็วขึ้น เพราะอ่านแค่ตัวใดตัวหนึ่งก็ได้ข้อมูลเหมือนกัน ส่วนอีกตัวก็ทำคำสั่งอื่นไป เก็บข้อมูลได้เหมือนเก็บตัวเดียว เช่น ใช้ HDD 5 ตัว ตัวละ 500 GB ทำ RAID 1 ก็จะได้เก็บข้อมูลได้ 500 GB แต่อ่านได้เร็วขึ้น 5 เท่า ข้อดีคือสามารถกู้ข้อมูลได้

RAID 5 ใช้ 3 ตัวขึ้นไป โดยใช้จะมีการกระจาย parity ไปใน HDD แต่ละตัว ตัวละ 1 ที่ ข้อดีที่ของการกระจาย parity จะสามารถกู้ข้อมูลได้ 1 ตัว และการอ่าน/เขียน จะเร็วขึ้นเป็น n เท่าอีกด้วย เพราะแบ่งข้อมูล เหมือน RAID 0 แต่จะต้องยอมเสียพื้นที่ HDD 1 ตัวในการทำ เช่น ใช้ HDD 5 ตัว ตัวละ 500 GB ทำ RAID 5 ก็จะได้เก็บข้อมูลได้ 2 TB แต่อ่าน/เขียนได้เร็วขึ้น 5 เท่า

RAID 6 มีหลักการคล้าย RAID 5 ทุกอย่าง แต่จะเก็บ parity 2 ที่ ดังนั้นจะต้องเสียพื้นที่ HDD ไป 2 ตัว แต่ข้อดีคือ HDD สามารถเสียได้ 2 ตัว แต่ก็ยังสามารถกู้ข้อมูลได้ทั้ง 2 ตัว

RAID 10 มาจาก RAID 0 + RAID 1 คือการสำรองข้อมูลใน HDD แบบ RAID 1 ก่อน แล้วมาแบ่งข้อมูลกับเก็บข้อมูลใน HDD อีกที แบบ RAID 0 จะต้องใช้ 4 ตัวขึ้นไปในการทำ โดยจะได้ข้อดีของทั้ง 2 วิธีมา คือสามารถอ่าน/เขียนได้เร็วขึ้น แต่ไม่เร็วขึ้น n เท่า จะเร็วขึ้น $n/2$ เท่า แต่ข้อดีที่สำคัญคือ สามารถกู้ข้อมูล $n/2$ ลูก แต่จะเก็บข้อมูลลดลงไปครึ่งหนึ่ง เช่น ใช้ HDD 4 ตัว ตัวละ 500 GB ทำ RAID 10 ก็จะได้เก็บข้อมูลได้ 1 TB ซึ่งความจริงควรได้ 2 TB แต่จะสามารถกู้ข้อมูลได้ 2 ลูกเลยทีเดียว

9 & 10 เรื่อง File ทั้งหมด

-> อธิบายความหมาย และหาตำแหน่งของไฟล์ ข้ออธิบายตามที่อาจารย์บอกมาจะมีอธิบาย Directory แต่ละแบบมีทั้งหมด 3 Level แล้วก็ File-System Structure และหาตำแหน่งหรืออาจให้อธิบายว่ามันทำงานยังไง คือ Allocation Methods มีทั้งหมด 3 Methods โดยข้อ 9 จะมี 4 ข้อย่อย ข้อละ 3 คะแนน รวม 12 คะแนน และข้อ 10 ไม่บอกว่ากี่ข้อ แต่ทั้งหมด 15 คะแนน

มาเอาข้อ 9 กันก่อน น่าจะออก Directory แน่ๆ แต่มันมีแค่ 3 แบบ ก็ได้แค่ 3 ข้อ แต่อาจารย์บอก 4 ข้อ อีกข้อ น่าจะอธิบายง่ายๆ ประมาณพวก file concept คืออะไร เสริมมาอีกข้อ จะอธิบายความหมายพวกที่น่าจะออกก่อนนะ และตามด้วย Directory มาเริ่มโหลดตัวหนังสือเข้าสมองกัน

มาเคลียร์คำให้เข้าใจตรงกันก่อน File : แฟ้มข้อมูล ก็คือไฟล์ทั่วไปในคอมของเรา เช่น รูปภาพ ไฟล์ notepad ไฟล์ zip ไฟล์เพลง ต่างๆนาๆ และน่าจะรวมไปถึง หน้าที่ y ด้วย อีๆ และเราก็จะชอบเก็บมันไว้ใน Folder ซึ่งในเรื่องนี้ เราจะใช้คำว่า Directory ซึ่งมันก็คือไฟล์เดอร์ที่เราคุ้นเคย ในการใช้เก็บหน้า y ของเรา 55 โอเคไรสาระไปแล้ว ไปหาสาระกันดีกว่า

File Concept : concept ในการเก็บไฟล์คือ ไฟล์จะต้องเก็บเรียงกัน

File Types : คือนามสกุลของไฟล์ ก็คืออักษรย่อ ที่อยู่หลังจุดในชื่อไฟล์ เช่น abc.txt นามสกุลของไฟล์ abc คือ txt ซึ่งจริงๆไม่จำเป็นต้องมีก็ได้ แต่ที่มีเพราะมันจะเปิดโปรแกรมของนามสกุลนั้นๆ ให้อัตโนมัติ

Directory : เป็นการจัดโครงสร้างแบบ logical ในระบบไฟล์ โดยคำนึงถึง

Efficiency : หาไฟล์ได้อย่างรวดเร็ว

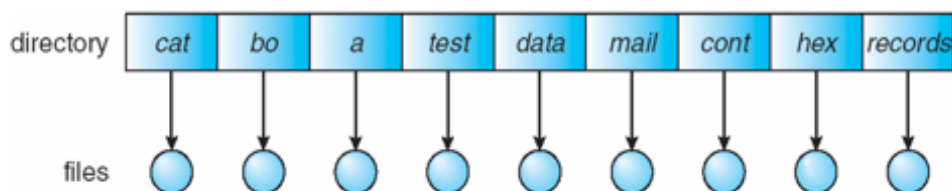
Naming : ผู้ใช้ 2 คนสามารถเก็บไฟล์ชื่อเดียวกันได้

Grouping : ต้องสามารถสร้าง Directory ใน Directory ได้หลายๆอัน คือในไฟล์เดอร์ ต้องสร้างไฟล์เดอร์ย่อยๆได้อีก เหมือนที่เราใช้กันในปัจจุบัน

Operation ที่ทำกับ Directory ได้ : สร้าง,ลบ,ค้นหา,เปลี่ยนชื่อ,list ข้อมูล,เดินทางตามระบบไฟล์

7 อันนี้ต้องออกสักอันแหละ มาดูอันที่ออกเน้นๆกันดีกว่า จำให้ขึ้นสมองเลย

Single-Level Directory

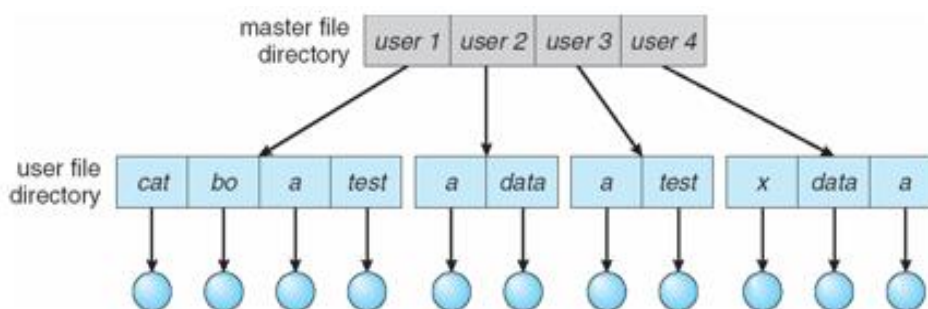


□ **Directory** แบบชั้นเดียว ใช้งานร่วมกันทุก **users**

- ▣ Naming problem
- ▣ Grouping problem

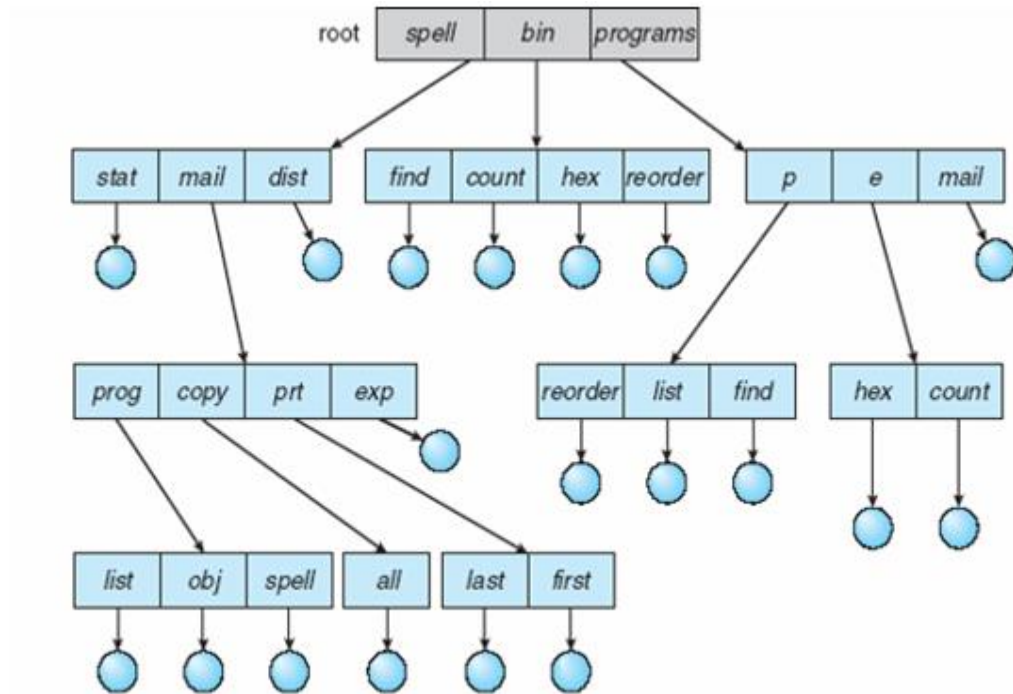
เป็น Directory แบบชั้นเดียว เช่นเข้าไปไดร์ C แล้วเจอไฟล์เลย แนนอนว่า เกิดปัญหา ทั้ง naming และ grouping เพราะถ้าชื่อไฟล์ซ้ำ ก็เขียนทับเลย และยังจะสร้าง directory ย่อยไปอีกไม่ได้ เพราะมันมีแค่ชั้นเดียว

Two-Level Directory



เป็น Directory ที่ ผู้ใช้แต่ละคนสามารถสร้าง Directory ใน Directory ได้ 1 ระดับเท่านั้น ซึ่งมันดีขึ้นกว่าเดิม คือแต่ละผู้ใช้ มีชื่อไฟล์เดียวกันได้ แต่ปัญหา grouping ก็ยังแก้ไม่ได้ เพราะถ้าจะจัดกลุ่มไฟล์เดอร์ ยังไม่ได้ เพราะสร้างได้แค่ระดับเดียว

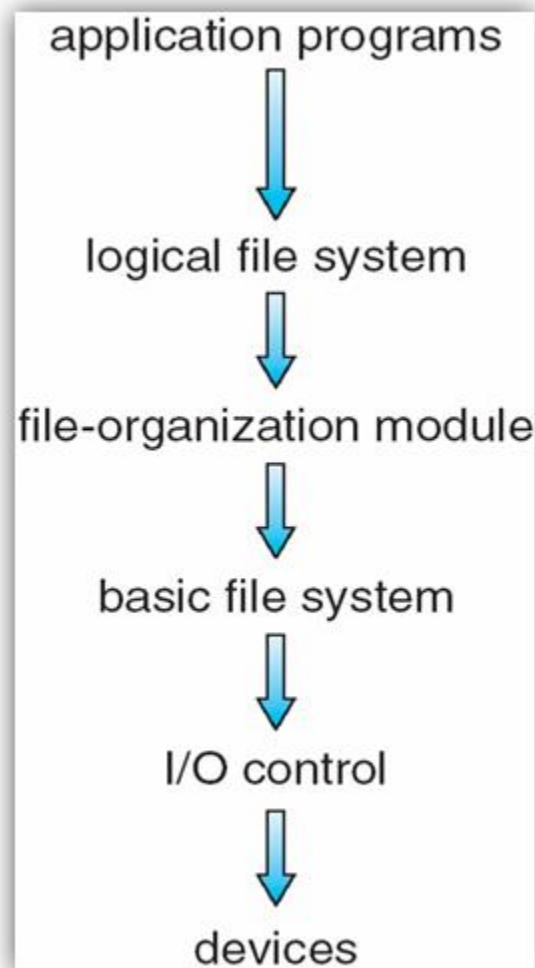
Tree-Level Directory



เป็น Directory ที่สร้าง Directory ย่อยไปได้เรื่อยๆ เลยทำให้แก้ปัญหาใน Level 1 และ 2 w ได้หมดเลย และนี่ก็คือ Directory ที่ใช้กันในปัจจุบัน โดยเราจะสร้างไฟล์เดอร์ในไฟล์เดอร์ที่อื่นก็ได้ เหมือนที่เราซ่อนหน้า y กันอะ สร้างซ้อนเข้าไปลึกๆ อธิ

ต่อไปข้อ 10 ออก 15 คะแนน โดยจารย์บอกมา 2 หัวข้อ คือ File System กับ Allocation Method น่าจะแบ่งเป็น File System 6 คะแนน แล้ว Allocation มันมี 3 Method ก็ Method ละ 3 คะแนน เป็น 15 คะแนนพอดี ฟันโอออ

File System เป็น User interface ให้ผู้ใช้เข้าใช้งาน storage เพื่อสามารถจัดเก็บและเข้าถึงข้อมูลใน disk ได้อย่างสะดวกและมีประสิทธิภาพ ในแต่ละ os มีหลายระบบไฟล์ มีรูปแบบเป็นของตัวเอง เช่น Windows จะมี FAT16, FAT32, NTFS แบ่งโครงสร้างออกเป็น layers ทั้งหมด 6 ชั้น ดังนี้



Application program : โปรแกรมที่เราใช้กันปกติ เช่น word

Logical file system : มีหน้าที่เชื่อมต่อผู้ใช้กับ Os โดยผู้ใช้สามารถจัดการกับ directory ได้

File – organization module : ทำหน้าที่เชื่อม logical block กับ physical block

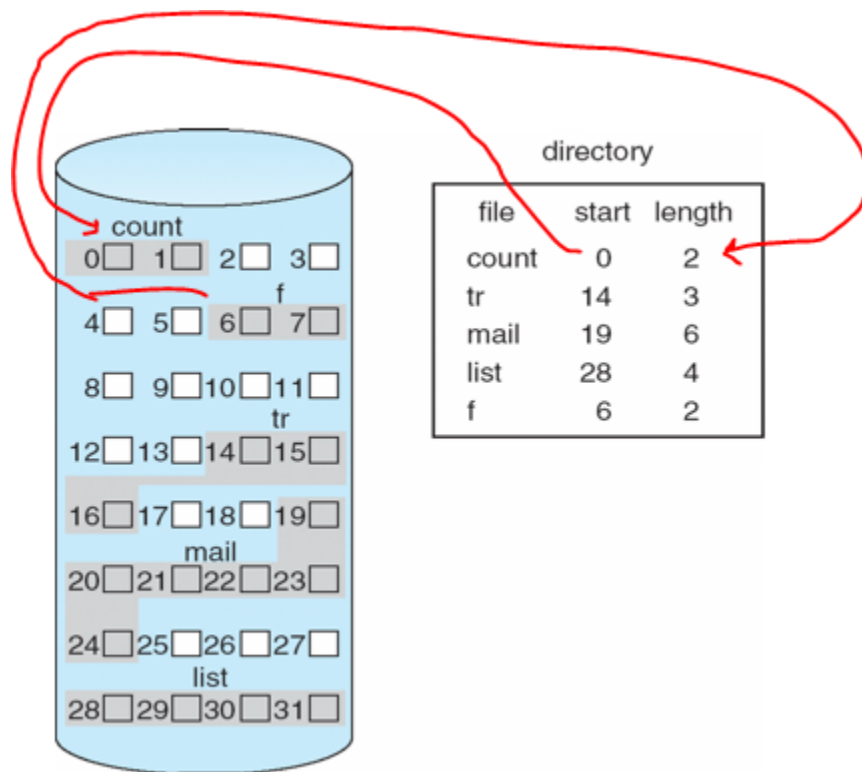
Basic file system : ทำหน้าที่เข้าไปอ่าน และเขียนในส่วนของ physical

i/o control : จัดการ i/o ประกอบด้วย device driver ควบคุมอุปกรณ์ให้ใช้งานได้

devices : อุปกรณ์ที่เก็บไฟล์ ถูกควบคุมโดย device driver

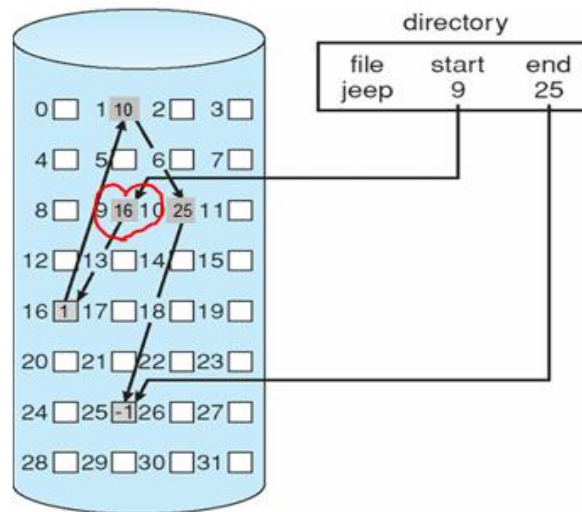
ต่อไปคือ Allocation Method คือการจัดสรร disk block ให้กับ ไฟล์ มีวิธีจัดสรรอยู่ 3 Algorithm ดังนี้

Contiguous block



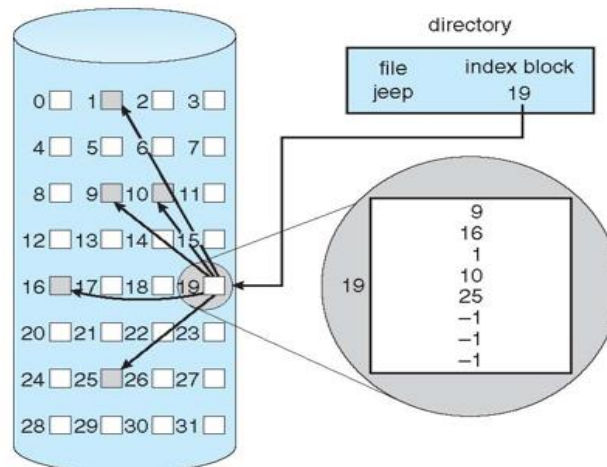
โดยแต่ละไฟล์ จะมีหมายเลขบล็อกที่ใช้ และความยาวจำนวนบล็อกที่ใช้บอกไว้ ตามรูป
ไฟล์ชื่อ count จะเริ่มต้นที่ block 0 แล้วใช้ 2 บล็อก ก็คือ 0 และ 1 นั่นเอง ประสิทธิภาพดีที่สุดใน
การเข้าถึงข้อมูล ทั้งแบบ Sequential และ random แต่จะยังมีปัญหา external fragmentation อยู่

Linked Allocation



โดยแต่ละไฟล์ จะมี linked list ไปหา blocks ต่างๆที่ใช้ โดยแต่ละ block จะเก็บตำแหน่งของ block ถัดไปไว้ ดีสำหรับการเข้าถึงแบบ Sequential แต่ช้าสำหรับ random แต่จะไม่เกิด external fragmentation เพราะสามารถชี้ไปหาพื้นที่บล็อกที่ไว้ได้ แต่ข้อเสียคือ ถ้าบล็อก 9 พังก็พังหมดเลย กู้ข้อมูลลำบาก และต้องใช้ i/o และ disk seek เยอะเพราะวิ่งอ่านข้อมูลเยอะ

Indexed Allocation



โดยแต่ละไฟล์ จะมี index block ที่เป็น pointer ไปยังแต่ละ block ของตัวเอง โดยวิธีนี้จะซับซ้อนมากขึ้น เพราะต้องอ่านข้อมูล 2 ครั้ง ทั้ง index และ data

11. Protection and Security -> ถาม-ตอบ คำศัพท์ โดยจะมีคีย์เวิร์ดเกี่ยวกับการโจมตีต่างๆมา แล้วให้เราอธิบาย 2 บรรทัดสั้นๆให้ได้ใจความ ทั้งหมด 18 คะแนน น่าจะมีสัก 9 ข้อย่อย หรือ 6 ข้อย่อย ข้อละ 2 หรือ 3 คะแนน ไปลุยกันนน **By โอมมีชัยปลาเค็ม :D**

CIA ประกอบด้วย 3 แบบนี้

Breach of confidentiality (C) ป้องกันการเข้าไปอ่านข้อมูลที่ไม่ได้รับอนุญาต

Breach of Integrity (I) ป้องกัน ถ้าเจาะข้อมูลได้แล้วก็ถูกโมดิฟายและรวมถึงการส่งข้อมูลผิดเพี้ยน

Breach of availability ป้องกันการทำลายข้อมูลทิ้ง

Theft of service ใช้ทรัพยากรของระบบโดยไม่ได้รับอนุญาต

denial of service (DOS) โจมตีการบริการจนไม่สามารถให้บริการไม่ได้

Masquerading ปลอมเป็นผู้ส่งข้อมูล

Replay attack เปลี่ยนข้อความในการส่งข้อมูล

Man-in-the-middle attack หลอกผู้ส่งว่าเราเป็นผู้รับ หลอกผู้รับว่าเราเป็นผู้ส่ง

Session hijacking เปิด session ทิ้งไว้ ทำให้ถูก ขโมยไปได้

Security จะจำต้องป้องกัน 4 ทาง ดังนี้

physical -> ป้องกันการดูข้อมูล จากภายนอก

Human -> เกิดขึ้นจากคนชอบแปะ postit ไว้ โดนแสบจากคน

Operating System ป้องกันโดยระบบปฏิบัติการ การดีบัค

Network -> ป้องกันจากช่องทางการสื่อสารแบบออนไลน์

การโจมตีรูปแบบต่างๆ ออกพวกนี้แน่นอนไม่ต่ำกว่า 4-5 ข้อ

Trojan Horse เปิดประตูหลังให้คนอื่นเข้ามารันเครื่องเราได้ โดยการตั้งใจลง program เอง แต่มี code ที่ไม่ประสงค์ดี เรียกอีกชื่อว่า spam

Trap Door โปรแกรมที่ Developer ฝังโค้ดไว้ เช่น ตอน login ให้ user เป็นอะไรก็ได้ แต่ถ้า pass เป็น EnET เข้าระบบได้หมดเลย ตรวจสอบยาก

Logic Bombs คือ โปรแกรมที่ตั้งเวลาทำงานเฉพาะสถานการณ์บางอย่าง โดยทำงานพร้อมกัน เช่น ฟรังก์ถ้า Os ยาก เรานัดกันยิง ping ไปที่เว็บบอร์ดให้ล่มไปเลย อธิ

Stack and Buffer Overflow โปรแกรมที่เขียนไม่ค่อยดี ทำให้ผู้ใช้ใส่ input มากเกินขอบเขต แล้วทำให้มีช่องโหว่ และอาจทำให้ระบบพังได้

Script kiddies คือแฮกเกอร์ (Hacker) หรือ แฮคกิง (Hacking) ประเภทหนึ่งมีจำนวนมาก ประมาณ 95 % ของแฮคกิง (Hacking) ทั้งหมด ซึ่งยังไม่ค่อยมีความชำนาญ ไม่สามารถเขียนโปรแกรมในการเจาะระบบได้เอง เช่น โปรแกรมสำเร็จรูป อาศัยให้เรา Download จากอินเทอร์เน็ต

Viruses ก่อตัวขึ้นได้เอง ต่างกับ trojan คือ viruses สามารถติดเครื่องอื่นได้

Monoculture วัฒนธรรมเดียวกัน ใช้ System เดียวกันทั้งองค์กร จะถูกเจาะพร้อมกัน

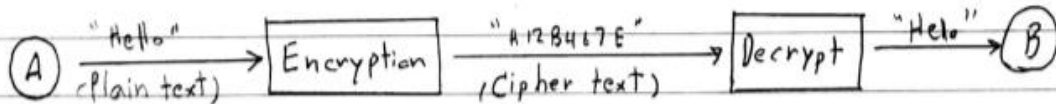
Worm เครื่องที่ติด จะส่งบริการที่ไม่มีการอัปเดต จากเครื่องอื่นๆ แล้วโจมตี ถ้าเจาะเข้าไปได้ จะส่งให้เครื่องนั้น โหลด worm จากเครื่องที่ติด แล้วก็ทำต่อไปเรื่อยๆ

Post scanning สแกนหาพอร์ตปลายทาง มีพอร์ตไหนเปิด มีช่องโหว่ทิ้งไว้ ก็เจาะเข้าไป

หลักการเข้าและถอดรหัส มี 2 แบบ 2 key ง่ายๆ จำไปมีคะแนนแน่นอน By จัมเปอร์เบอเริ่มเต็ม

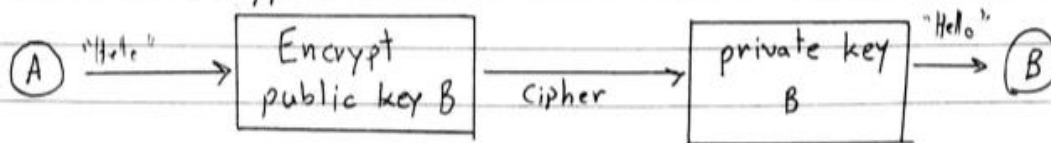
Cryptography, Encryption

Symmetric Encryption (สมมาตร ผู้รับ-ส่ง ใช้ key เดียวกัน)



- ผู้รับและผู้ส่งจะใช้ key k เดียวกัน ตอนส่งผู้ส่งจะ Encrypt ด้วย key k
ผู้รับก็จะ Decrypt ด้วย key k และ รับข้อความมา

Asymmetric Encryption



- จะมีสองคีย์ key ก็นั่น ก่อนส่งจะมีการแลกเปลี่ยน public key กับ ทักท้ออย่าง
ผู้ส่ง Encrypt ด้วย Public key ของผู้รับ แล้วผู้รับจะ Decrypt ด้วย Private key เหนือ

โชคเอนะ

ขอให้ได้อย่างที่หวังตามที่ตั้งใจไว้ทุกคน