

Virtual Memory

Virtual Memory คือ หน่วยความจำที่ถูกสร้างขึ้นมาในกรณีที่ RAM ไม่พอใช้ ซึ่งโดยส่วนมากมักเอาพื้นที่ในฮาร์ดดิสก์บางส่วนมาใช้แทน เนื่องจากหน่วยความจำของระบบมีจำกัดและมีราคาสูง เราจึงเรียกว่าความจำเสมือน

Valid and Invalid

- Valid แสดงว่า page อยู่ใน logical address ของ process นั้น
- Invalid แสดงว่า page ไม่อยู่ใน logical address space ของ process นั้น

Page Fault

โปรแกรมอ้างถึง page ที่ไม่มีอยู่ใน physical memory

Page fault exception ให้กับ OS

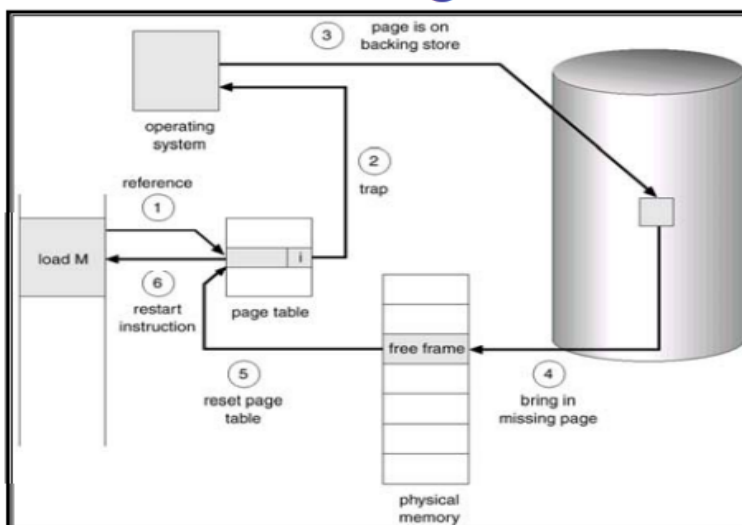
OS ตรวจสอบว่า page ที่ต้องการอยู่ใน address space หรือไม่?

- NO หยุดการทำงานของโปรแกรม
- YES load page ที่ต้องการลง memory

ขั้นตอนในการจัดการ Page Fault

1. ตรวจสอบข้อมูลใน PCB ของโปรเซส ว่าหน้าที่ ต้องการอ้างอิงถูกต้อง หรือไม่?
2. ถ้าการอ้างอิงผิดพลาดให้ยกเลิกโปรเซส ถ้าถูกต้องแต่ page ที่ต้องการนี้ไม่ได้อยู่ main memory ให้ดำเนินการขั้นตอนต่อไป
3. หาที่ว่างในหน่วยความจำ (Frame) 1 หน้า เช่น เลือกจากรายการ ชื่อ หน้าว่าง (Free frame list)
4. จัดคำร้องขอส่งไปยังงานบันทึก เพื่อให้อ่าน page ที่ต้องการ มาไว้ในเนื้อที่ว่างที่เลือกไว้
5. เมื่องานบันทึกนำ page เข้ามาเสร็จแล้ว ระบบจะแก้ไขบิตสถานะในตารางเลขหน้าเป็น valid
6. จัดโปรเซสทำงานต่อจากจุดที่เกิด page Fault

ขั้นตอนจัดการ Page Fault



Fetch Policy: กำหนดว่า page ไหน load ลง memory

Demand paging – load เฉพาะ page ที่โปรแกรมอ้างถึงลง memory

- เร็วเพราะ load แค่ page เดียว
- แต่เกิด page fault บ่อย
- ใช้ swap space หรือ Backing store

Prepaging - load page ที่อยู่ต่อเนื่องมาด้วย

- เสียต่อ page ที่โหลดเพิ่มจะไม่ถูกใช้
- เกิด page fault น้อยลง

ประสิทธิภาพของ Demand Paging

Page Fault Rate ($0 \leq p \leq 1$)

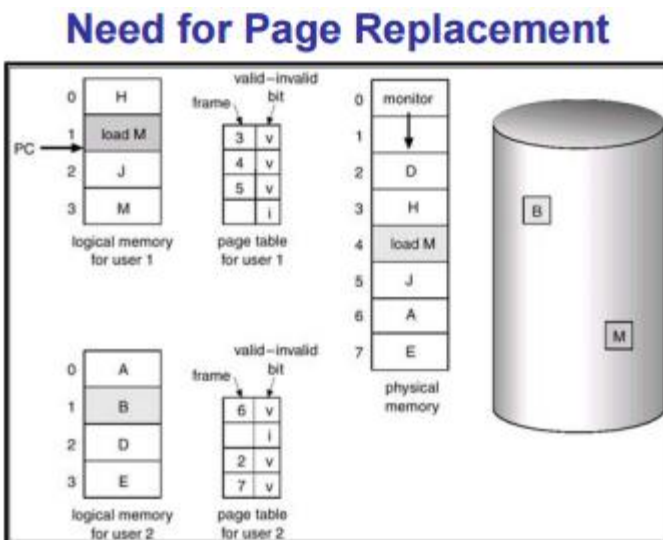
- if $p = 0$ no page faults
- if $p = 1$, every reference is a fault

Effective Access Time (EAT)

$$EAT = (1 - p) * \text{memory access} + p * \text{Page fault time}$$

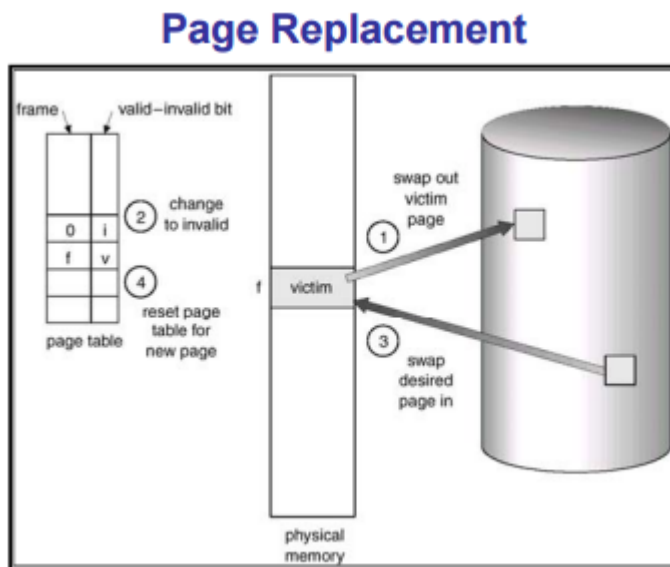
Page Replacement

- ไม่มี frame ของ physical memory เหลือให้ load page ที่ต้องการ
- OS หา page ที่อยู่ใน memory ที่ไม่ถูกใช้ ณ ขณะนั้น
- หาก page นั้นไม่ถูกแก้ไขระหว่างทำงาน จะ load ทับ
- หาก page นั้นถูกแก้ไข จะ swap page นั้นออกไปเก็บไว้ใน disk ก่อน



ขั้นตอนการทำ Page Replacement

- 1 หาดำแหน่งของ **page** ที่ต้องการบนจานบันทึก
- 2 หาพื้นที่ว่างในหน่วยความจำ
 - ถ้ามีที่ว่าง ให้ใช้หน้านั้นเลย
 - ถ้าไม่มีที่ว่าง ให้ใช้ขั้นตอนในการสับเปลี่ยนหน้า เลือกหน้าที่จะ **swap** ออกไป
- 3 อ่านหน้าที่ต้องการเข้ามาเก็บไว้ในเนื้อหาที่ว่างที่เลือกไว้ แก้ไขตารางข้อมูลให้ถูกต้อง
- 4 ให้กระบวนการที่หยุดรอทำงานต่อ



Swapping Area

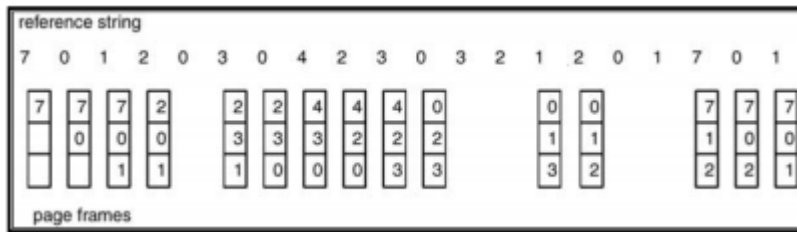
- พื้นที่ในDisk ที่ใช้เก็บ **page** ของโปรแกรมที่ถูก **replace** โดย **page** อื่นๆ
- Windows จะเป็น Page file
- Unix, Linux จะเป็น Swapping file

Page Replacement Algorithms จะมี

- FIFO
- Optimum
- LRU

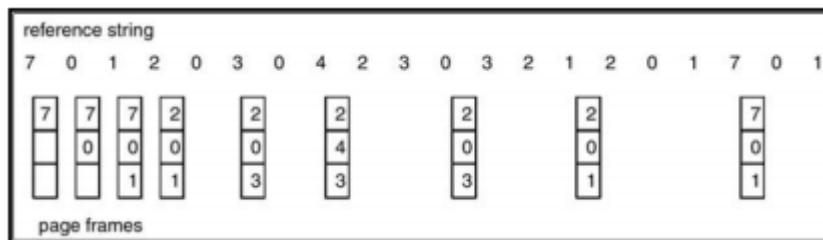
First-In-First-Out (FIFO) Algorithm

Page ที่เข้ามาก่อนจะถูก Replace ก่อน First-in-Firstout (แบบเข้าก่อน-ออกก่อน)



Optimal Algorithm

Page ที่จะไม่ถูกใช้นานที่สุดจะถูกreplace (จะต้องรู้ว่าPageไหนถูกใช้เมื่อไหร่)



Least Recently Used (LRU) Algorithm

Page ที่ถูกใช้งานน้อยสุดจะถูก Replace

