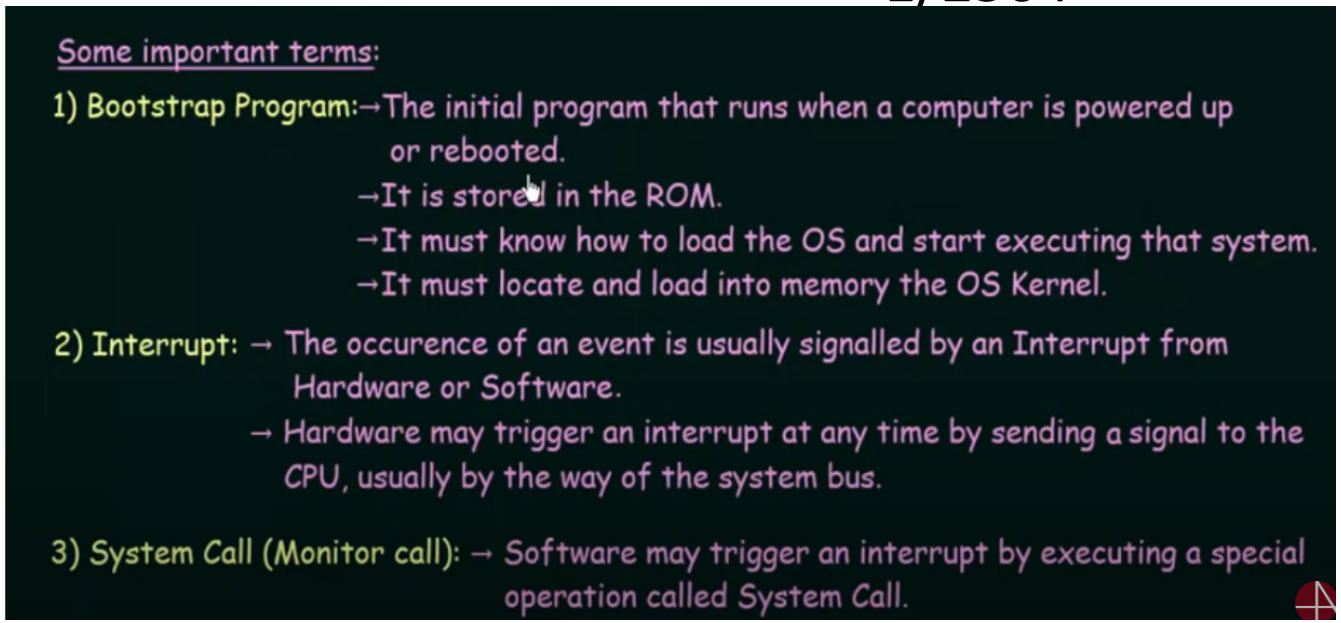
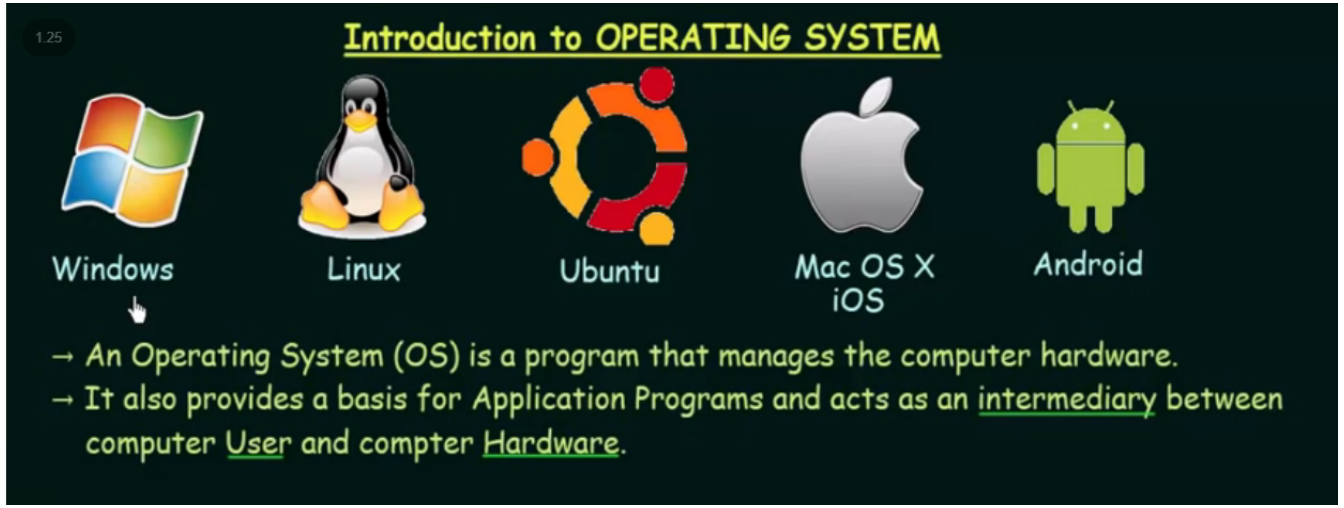
Introduction

1/2564



โจทย์

- ในความคิดเห็นของนักศึกษา ระบบปฏิบัติการมีบทบาทหน้าที่หลักอะไรบ้าง



Roles of the Operating System

ทำหน้าที่เป็น Referee , Illusionist , Glue

Referee (กรรมการ) : ผู้ชี้ขาดว่าใครได้ทรัพยากรเท่าไหร่ แล้วจัดคิวว่าใครใช้ก่อนใช้หลัง

- **Resource allocation** among users, applications
- **Isolation** of different users, applications from each other
- **Communication** between users, applications

Illusionist (นักเลนกล) : หลอก app วามีตัวมันเพียงคนเดียวที่ใช้ทรัพยากรทั้งเครื่องอยู่

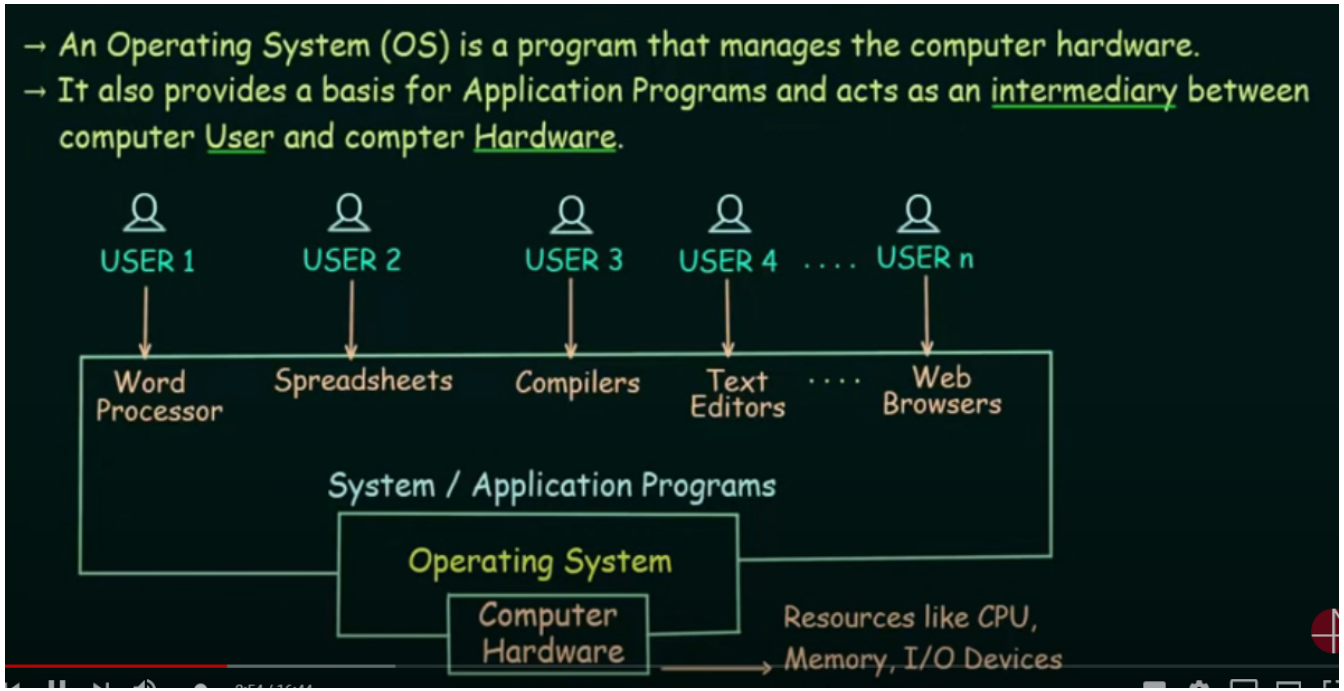
- Each application appears to have the entire machine to itself
- Infinite number of processors, (near) infinite amount of memory, reliable storage, reliable network transport

Glue (คนเชื่อม) : ประสานส่วนต่าง ๆเพื่อใช้งาน

- Libraries, user interface widgets, …

โจทย์

- ระบบปฏิบัติการคืออะไร?

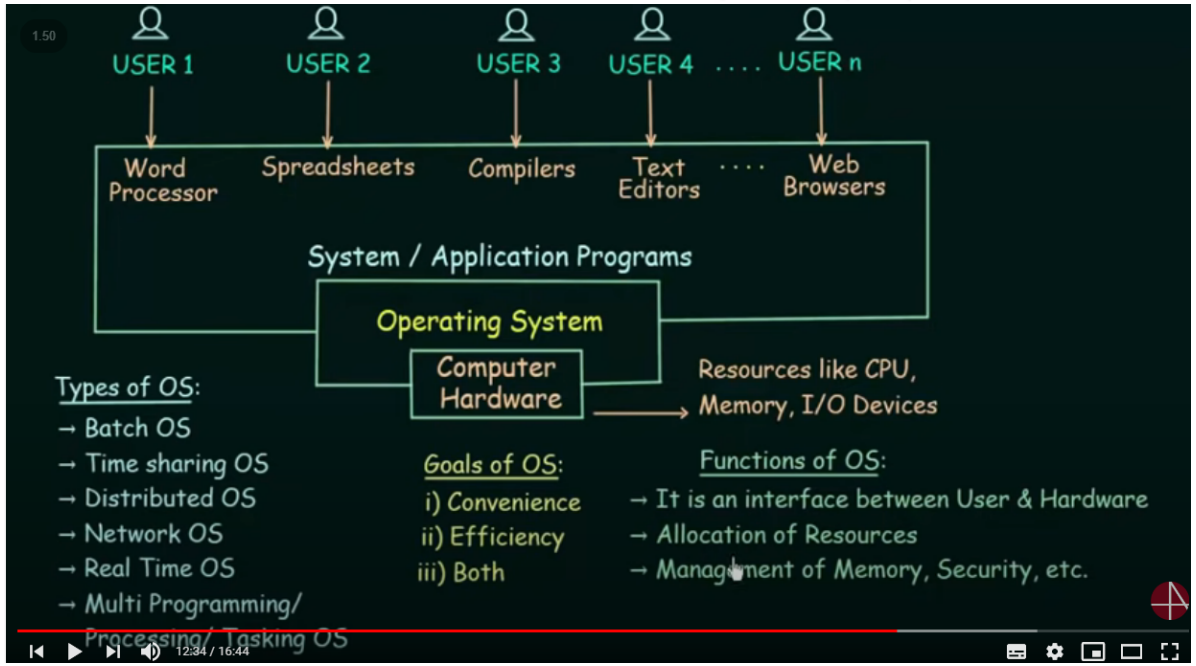


What is an Operating System?

- A set of software that manage computer's resources for its users and their applications

มาจากหลายๆ software

- May visible or invisible to the user
- 2 major kinds
  - General purpose OS
  - Specific purpose OS



โจทย์

- หากนักศึกษาต้องประเมินระบบปฏิบัติการหนึ่ง นักศึกษาจะประเมินด้านใดบ้าง และแต่ละด้านจะวัดอย่างไร

Operating System Evaluation

ความน่าเชื่อถือและความพร้อมในการให้บริการที่ตรงตามเป้าหมาย

- Reliability and Availability

ความปลอดภัย

- Security

ใช้กับอุปกรณ์ได้หลากหลาย

- Portability
  - AVM, API, HAL

In the case of a virtual machine, you can have a program that run directly on the hardware, or runs under the control of a virtual machine that hides (and protect) somewhat the hardware. Running on a virtual machine will slow the program, and the variation in running time is the overhead (extra time cost) caused by the virtual machine. You just measure the time taken by the same program running in both situations, and you determine the percentage of additional time required when running in a virtual machine. It may depend on the type of application you use to test the overhead of the virtual machine.

Overhead is a measure of extra cost. It applies to devices or programs or procedure that cause an extra cost (but usually provide an extra servive).

Application ไม่ติดต่อ HW โดยตรง เลยสร้าง AVM มาจำลอง Function การใช้งาน

Abstract Virtual Machine , API (เป็นตัวกลางระหว่าง SW ซ้าย และ HW ขวา) , Hardware Abstract Layer

- Performance
  - Overhead, efficiency
  - Fairness, response time, throughput
  - Performance predictability

The meaning of overhead is usually an additional expense occurring in addition to normal cost (whatever normal cost may be).

ความเป็นที่ยอมรับ

- Adoption

มี Driver

Design Tradeoffs

ได้อย่างต้องเสียอย่าง

- Must balance between the 5s
  - Examples
    - Preserves legacy API → Portability ↑, reliable ↓, secure ↓
    - Breaking an abstraction → Performance ↑, Portability ↓, Reliability ↓

Computer Performance Over Time

	1981	1997	2014	Factor (2014/1981)
Uniprocessor speed (MIPS)	1	200	2500	2.5K
CPUs per computer	1	1	10+	10+
Processor MIPS/$	$100K	$25	$0.20	500K
DRAM Capacity (MiB)/$	0.002	2	1K	500K
Disk Capacity (GiB)/$	0.003	7	25K	10M
Home Internet	300 bps	256 Kbps	20 Mbps	100K
Machine room network	10 Mbps (shared)	100 Mbps (switched)	10 Gbps (switched)	1000
Ratio of users to computers	100:1	1:1	1:several	100+

From Thomas Anderson and Michael Dahlin, Operating Systems Principles & Practice Volume I, 2nd edition, Recursive Books, 2015

Early Operating Systems: Computers Very Expensive

รันทีละโปรแกรม

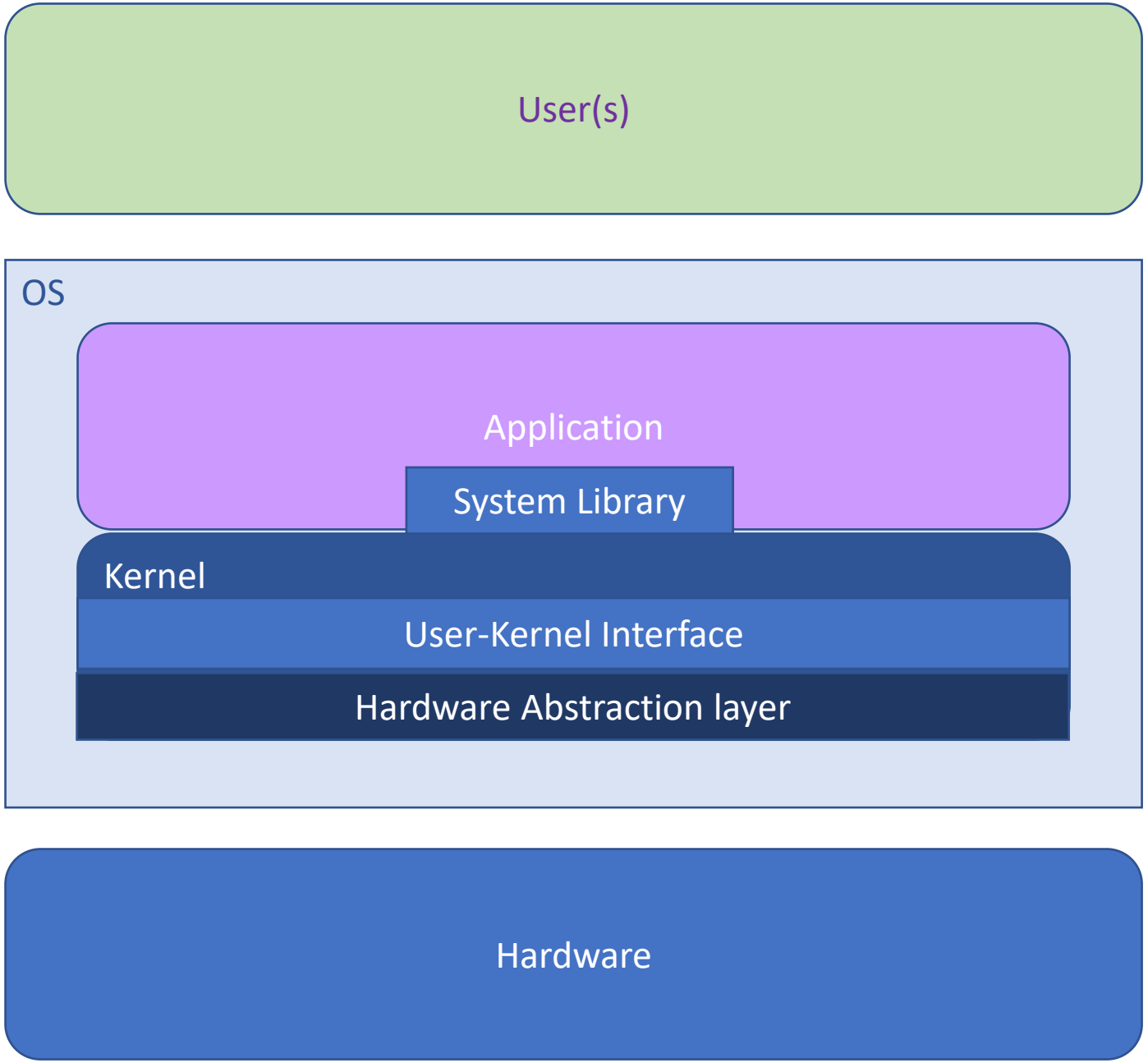
- One application at a time
  - Had complete control of hardware
  - OS was runtime library
  - Users would stand in line to use the computer
- Batch systems
  - Keep CPU busy by having a queue of jobs
  - OS would load next job while current one runs
  - Users would submit jobs, and wait, and wait, and

Time-Sharing Operating Systems: Computers and People Expensive

- Multiple users on computer at same time
  - Multiprogramming: run multiple programs at same time
  - Interactive performance: try to complete everyone's tasks quickly
  - As computers became cheaper, more important to optimize for user time, not computer time

Today's Operating Systems: Computers Cheap

- Smartphones
- Embedded systems
- Laptops
- Tablets
- Virtual machines
- Data center servers


User(s)
OS
Application
System Library
Kernel
User-Kernel Interface
Hardware Abstraction layer
Hardware

Tomorrow’s Operating Systems

- Giant-scale data centers
- Increasing numbers of processors per computer
- Increasing numbers of computers per user
- Very large scale storage