ความรู้พื้นฐานเกี่ยวกับระบบปฏิบัติการ (Introduction to Operating System)

ผู้จัดการล่องหนแห่งโลกคอมพิวเตอร์ (The Invisible Manager of Your Computer)

ระบบปฏิบัติการ (OS) คืออะไร?

ผู้ประสานงานอัจฉริยะ (The Ultimate Coordinator)

หน้าที่หลัก:
โปรแกรมที่เป็นผู้ประสานงานระหว่างผู้ใช้คอมพิวเตอร์ (User) และฮาร์ดแวร์ (Hardware)

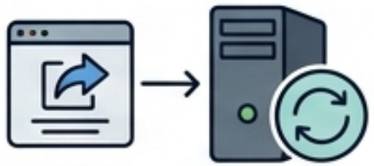
เป้าหมาย:
จัดเตรียมสิ่งอำนวยความสะดวกเพื่อให้ผู้ใช้ทำงานได้ง่ายที่สุด โดยไม่ต้องรู้กลไกที่ซับซ้อนของเครื่อง

ผลลัพธ์:
การใช้งานทรัพยากรอย่างมีประสิทธิภาพสูงสุด

3 รูปแบบการสร้างระบบปฏิบัติการ

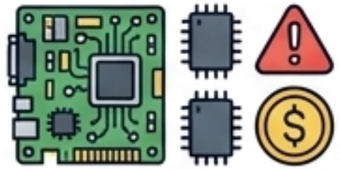
ซอฟต์แวร์ OS (Software)

- ทั่วไปนิยมใช้ที่สุด
- **ข้อดี:** ปรับปรุงแก้ไขข้อบกพร่องได้ง่าย
- **ข้อเสีย:** ทำงานช้ากว่าฮาร์ดแวร์



ฮาร์ดแวร์ OS (Hardware)

- สร้างจากอุปกรณ์อิเล็กทรอนิกส์
- **ข้อดี:** ทำงานรวดเร็วที่สุด
- **ข้อเสีย:** แก้ไขยากมากและมีราคาแพง การเปลี่ยน OS คือการสร้างเครื่องใหม่



เฟิร์มแวร์ OS (Firmware)

- เขียนด้วยคำสั่งไมโครโปรแกรม (เช่น BIOS)
- **จุดเด่น:** ความเร็วอยู่ระดับกลาง แก้ไขได้แต่มีค่าใช้จ่าย คุมการทำงานของ CPU ในระดับต่ำสุด

3 บทบาททางเทคนิคของ OS

1. Resource Allocator (ผู้จัดสรรทรัพยากร)

บริหาร Memory, Hard Disk และ Printer ให้เกิดประโยชน์สูงสุด

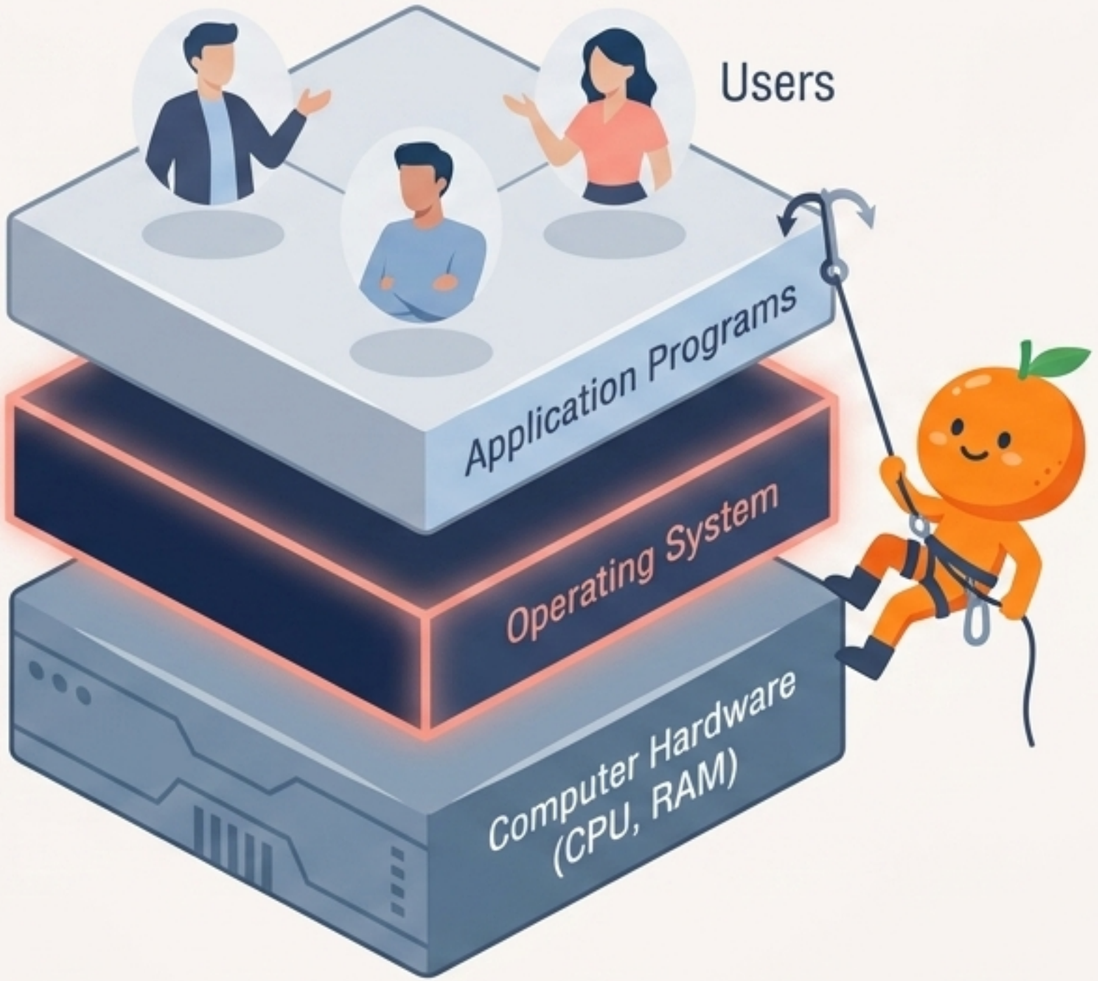
2. Control Program (ผู้ควบคุม)

ควบคุมการ Execute โปรแกรมของผู้ใช้และการทำงานของอุปกรณ์ I/O

3. Kernel (แก่นแท้)

โปรแกรมหลักที่ทำงานอยู่ตลอดเวลาบนคอมพิวเตอร์ในระดับฮาร์ดแวร์

สถาปัตยกรรมระบบคอมพิวเตอร์: OS อาศัยอยู่ตรงไหน?



OS คือชั้น (Layer) ที่สำคัญที่สุดที่คอยปิดบังความซับซ้อนของ Hardware เอาไว้ ทำให้ Application และ User สามารถสั่งงานเครื่องได้โดยไม่ต้องเขียนโค้ดคุยกับฮาร์ดแวร์โดยตรง

3 หน้าที่หลักที่ OS ทำงานอยู่เบื้องหลัง

1. ติดต่อกับผู้ใช้ (User Interface)

รับคำสั่งหรือสัญลักษณ์จากผู้ใช้
(เช่น การสั่ง Copy)
ผ่าน System Call

2. ควบคุมอุปกรณ์ (Device Control)

ดูแลอุปกรณ์ต่างๆ ให้ทำงาน
อย่างถูกต้องและสอดคล้องกัน
โดยผู้ใช้ไม่ต้องรู้หลักการทำงาน

3. จัดสรรทรัพยากร (Resource Allocation)

เนื่องจาก CPU, Memory, I/O มีจำกัด
OS ต้องแจกจ่ายทรัพยากรให้ทุกโปร
แกรมอย่างเหมาะสมและไม่แย่งกัน

วิวัฒนาการยุคที่ 1: จากเครื่องเปล่าสู่การลดเวลาคอย

ระบบเครื่องเปล่า (Non-OS):
ผู้ใช้ต้องป้อนข้อมูลและควบคุมเครื่องด้วยมือทุกขั้นตอนผ่าน Console

ระบบงานแบ็ตช์ (Batch System):
รวบรวมงานที่คล้ายกันเป็นกลุ่ม (Job Pool) แล้วส่งให้เครื่องประมวลผลรวดเดียว

ระบบบัฟเฟอร์ (Buffering):
สร้างพื้นที่พักข้อมูล (Buffer) ในหน่วยความจำ เพื่อให้หน่วยรับ-แสดงผล ทำงานไปพร้อมกับการประมวลผลของ CPU ได้

วิวัฒนาการยุคที่ 2: รีดประสิทธิภาพ CPU ให้ถึงขีดสุด

ระบบสพูลลิ่ง (Spooling):

ระบบสพูลลิ่ง (Spooling): Simultaneous Peripheral Operating On-Line. แยกการประมวลผล CPU ออกจากการรับ-ส่งข้อมูล (เช่น พิมพ์เอกสารพร้อมกับรันงานอื่น) โดยมี Job Pool คอยจัดลำดับความสำคัญ (Priority)

ระบบมัลติโปรแกรมมิ่ง (Multiprogramming):

โหลดหลายโปรแกรมไว้ในหน่วยความจำหลัก เมื่อโปรแกรมหนึ่งหยุดรอ (Wait) ใบหยุดรอ (Wait) OS จะดึงโปรแกรมถัดไปเข้า CPU ทันที ทำให้ CPU ไม่เคยว่างงาน

วิวัฒนาการยุคที่ 3: สู่ยุคแห่งการเชื่อมต่อไร้ขีดจำกัด

Time-Sharing: สับเปลี่ยนงานของหลายคนเข้าสู่ CPU ด้วยความเร็วสูงมาก จนทุกคนรู้สึกเหมือนครอบครอง CPU ไว้คนเดียว

Personal Computer (PC): เครื่องราคาถูกลง มุ่งเน้นความบันเทิงและธุรกิจส่วนตัว (Desktop, Notebook, PDA)

Multiprocessor: ใช้ CPU หลายตัวช่วยกันทำงาน (สมมาตร/ไม่สมมาตร) เพื่อเพิ่มพลังประมวลผล

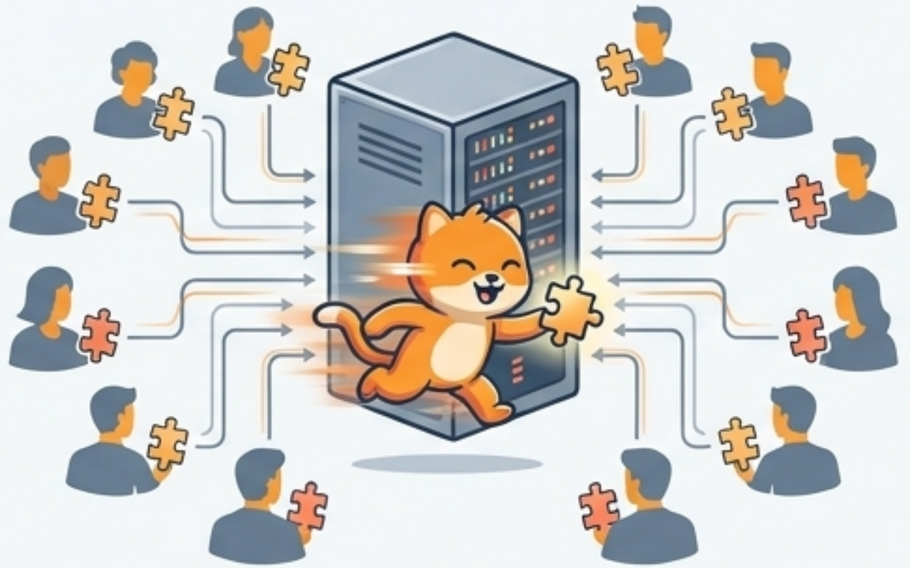
Distributed System: ระบบเครือข่าย (TCP/IP) ที่กระจายหน้าที่และแชร์ทรัพยากรร่วมกันทั่วโลก

ขีดจำกัดผู้ใช้: Single-User vs Multi-User

Single-Tasking:

Single-Tasking: ยอมให้มีผู้ใช้เพียงคนเดียว และทำงานได้เพียงอย่างเดียวในเวลาหนึ่ง (เช่น แปลโปรแกรมอยู่ จะพิมพ์ข้อความไม่ได้) จัดการทรัพยากรไม่ซับซ้อน

Multi-User (Time-Sharing):



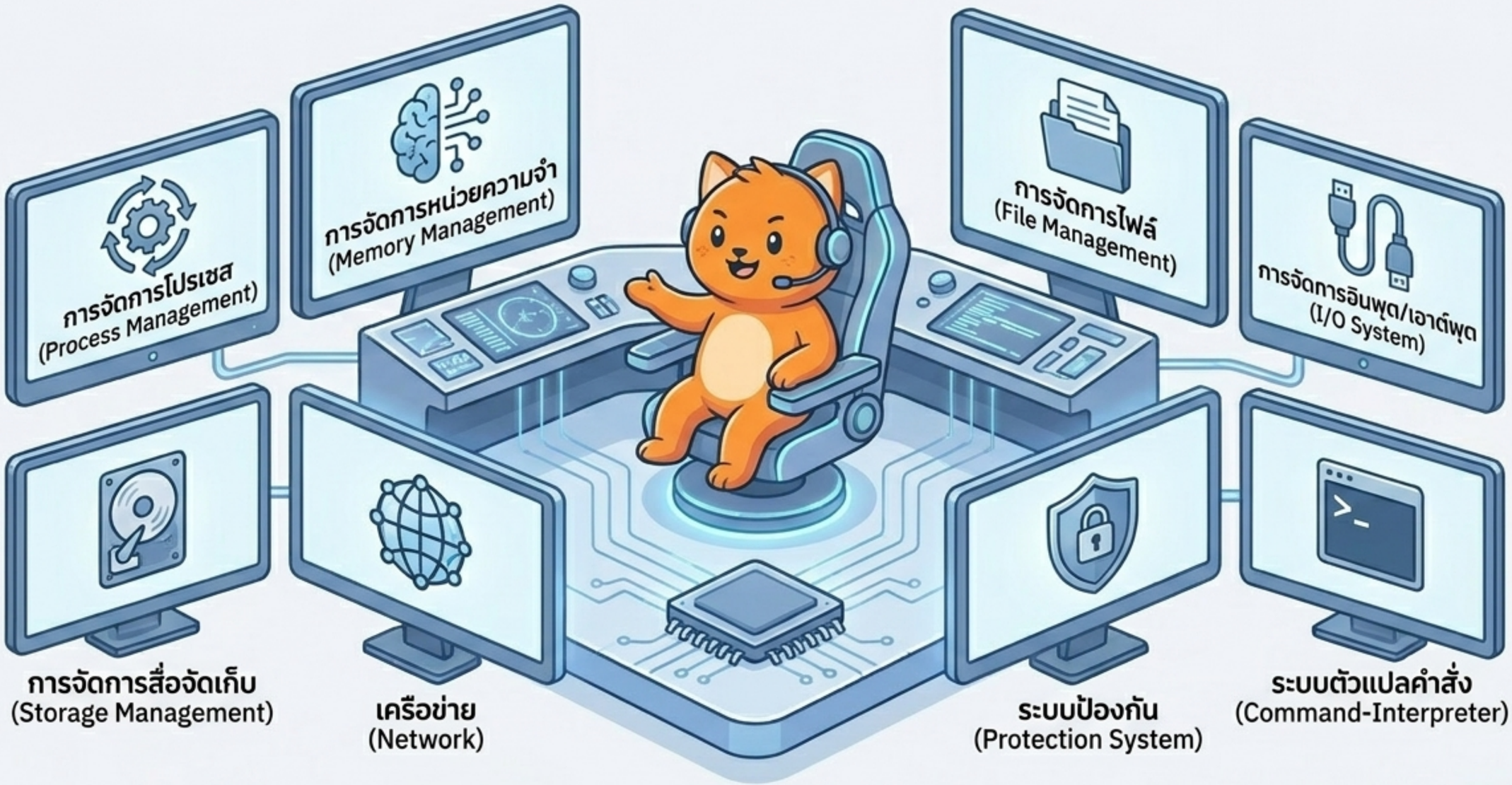
Multi-User (Time-Sharing): ยอมให้โหลดหลายโปรแกรม แบ่งช่วงเวลา (Time Slice) สั้นๆ สลับสับเปลี่ยนให้แต่ละโปรแกรมเข้าใช้ CPU อย่างรวดเร็ว ผู้ใช้แต่ละคนจะรู้สึกว่าตนได้ครอบครองคอมพิวเตอร์แต่เพียงผู้เดียว (เช่น UNIX, VMS)

เครื่องคอมพิวเตอร์เสมือน (Virtual Machine)

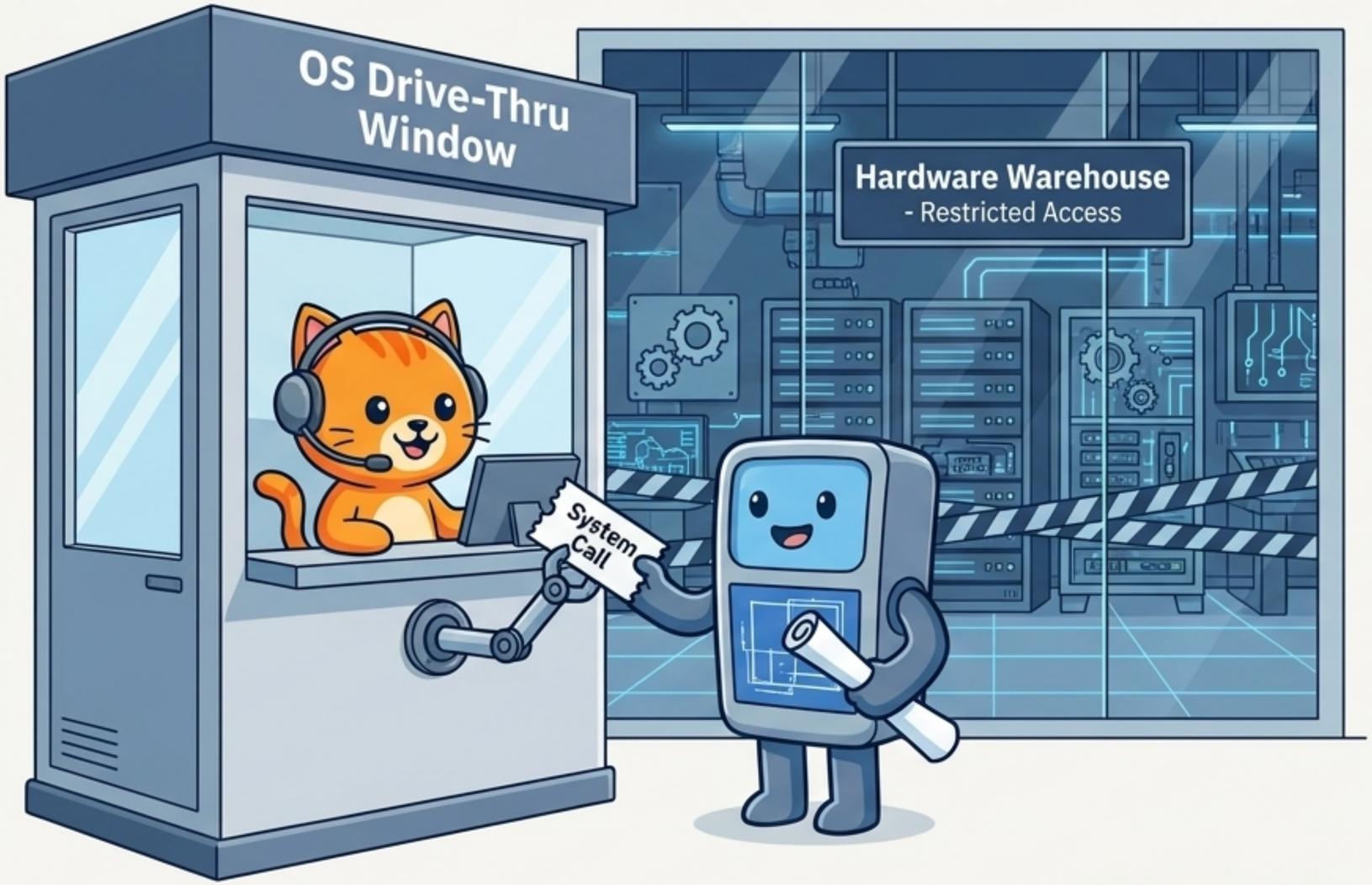
แนวคิดหลัก:

- **การแปลง**เครื่องคอมพิวเตอร์ 1 เครื่องให้กลายเป็นคอมพิวเตอร์หลายเครื่อง (เสมือน)
- **การทำงาน:** OS ตัวแรกทำหน้าที่เป็นฐานและจำลองสภาพแวดล้อมให้สามารถติดตั้ง OS ตัวที่ 2, 3 เข้าไปทำงานซ้อนทับได้
- **ประโยชน์:** ผู้ใช้หลายคนสามารถรันโปรเซสและระบบปฏิบัติการที่ต่างกันบนฮาร์ดแวร์จริงเพียงชุดเดียวได้อย่างอิสระและปลอดภัย

ห้องเครื่องยนต์: โครงสร้างของระบบปฏิบัติการ (OS Structure)



System Call: ช่องทางบริการสำหรับโปรแกรม



System Call คืออะไร?

ส่วนที่ OS จัดไว้ให้ผู้ใช้/โปรแกรม
เรียกใช้งานฮาร์ดแวร์ได้อย่างสะดวก
โดยไม่จำเป็นต้องรู้กลไกเชิงลึกของเครื่อง

ประเภทบริการหลัก:

- **ควบคุมโปรเซส**
  (Process Management)
- **จัดการไฟล์และอุปกรณ์**
  (File & Device Management)
- **บำรุงรักษาข้อมูล**
  (Data Maintenance)
- **ติดต่อสื่อสาร**
  (Communication)

บทสรุป: ผู้จัดการล่องหนที่ทำให้คอมพิวเตอร์มีชีวิต

หัวใจของระบบ (The Core):

ระบบปฏิบัติการคือสะพานเชื่อมที่ขาดไม่ได้ เปลี่ยนฮาร์ดแวร์ที่ซับซ้อนให้กลายเป็นเครื่องมือที่ใช้งานง่าย

นักจัดการเวลาและทรัพยากร (The Master of Efficiency):

วิวัฒนาการจากยุคที่ต้องทำงานทีละคำสั่ง สู่การสับเปลี่ยนงานระดับเสี้ยววินาที (Multitasking) จนเราแทบไม่รู้สึก

ผู้คุ้มครองเบื้องหลัง (The Invisible Shield):

คอยจัดสรรทรัพยากรและป้องกันข้อผิดพลาดผ่าน System Call เพื่อให้ผู้ใช้โฟกัสแค่จินตนาการและการสร้างสรรค์ผลงาน