

Advanced Operating System and Virtualization

Introduction
Hiroaki Fukuda

Contents

- Purpose
 - Understanding Operating System based on exercise
 - Understanding Virtualization based on exercise
- What to do in this course
 - Implementing disassembler
 - Understanding binary
 - Implementing Interpreter (Virtual Machine)
 - Understanding operator of CPU
 - Understanding System Call
 - Understanding essential part of Virtualization

Syllabus

- | | |
|----------------------------------|------------------------|
| 1. Introduction | 8. System Call |
| 2. Brainf*ck | 9. Execute 1.c and 2.c |
| 3. Setup Environment read Binary | 10. Open/Read/Close |
| 4. Disassemble1 | 11. Creat/brk/lseek |
| 5. Disassemble2 | 12. Execute printf |
| 6. Disassemble3 | 13. Execute 3 ~ 7.c |
| 7. Introduction of Interpreter | 14. Execute nm |

Required Skill and knowledge

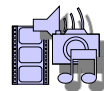
- Skill
 - Enough experience using any programming language (e.g., C and Java)
- Knowledge
 - Basic understanding Computer Architecture
 - Memory, CPU, Address, Byte, Bit, Endian...
- Environment
 - Linux or Mac (for executing reference interpreter)

Overview

- What is Operating System
 - Roll of Operating System
 - History

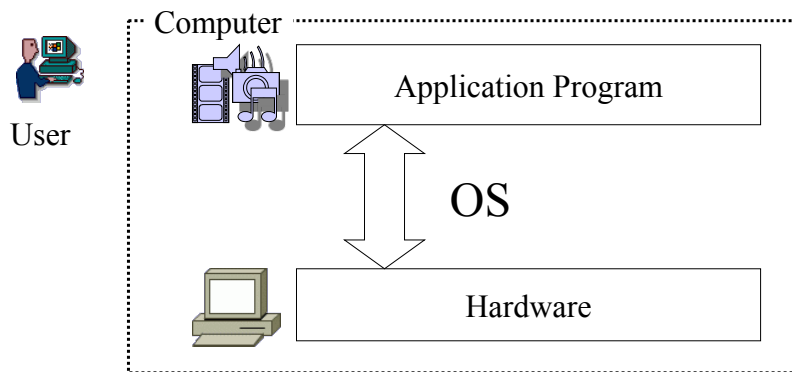
Components of Computer Systems

- Hardware
 - Provide basic resources
 - CPU, Memory, IO devices
- Application Program
 - Solve any problems using system resources
 - Compiler, DataBase, Games
- User
 - Solve its own problem using computer
 - People, another computer



Operating System

- Software system which bridges application program and hardware



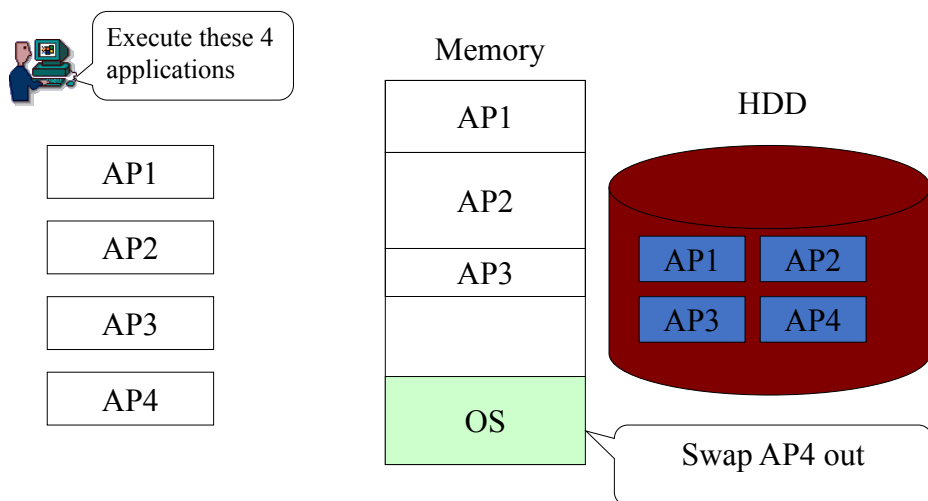
Roll of Operating System

- Resource Management
- Control Program
- Resource Virtualization

Resource Management

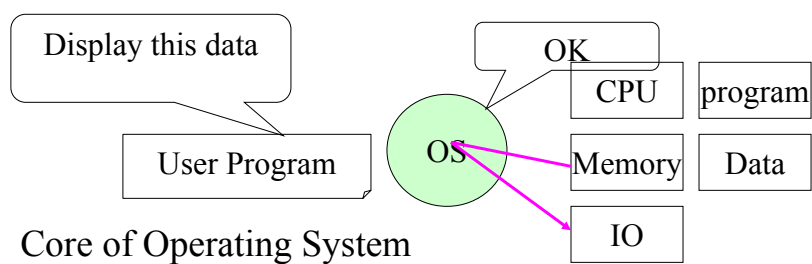
- Resource
 - Hardware: CPU, Memory, IO
 - Software: Program, Data
- Purpose
 - Control shared resources
 - Adequate access from each application

Resource Management

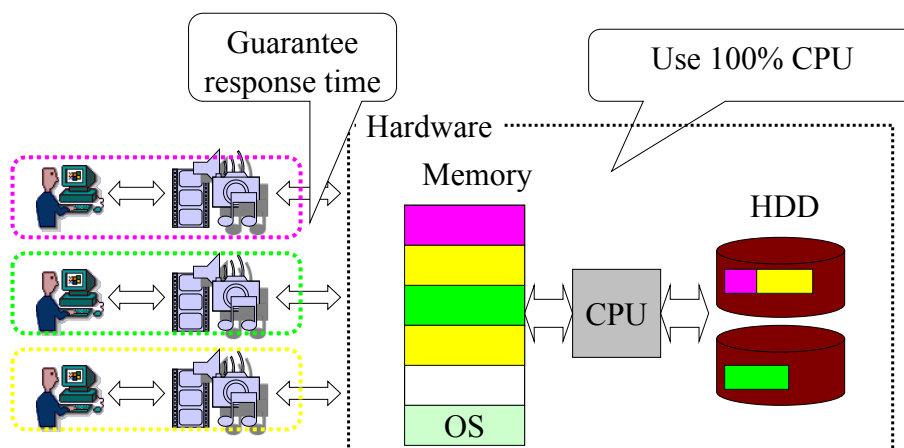


Control Program

- Management User Program
- Manage hardware resources
 - Access control to hardware
 - Guarantee performance and reliability

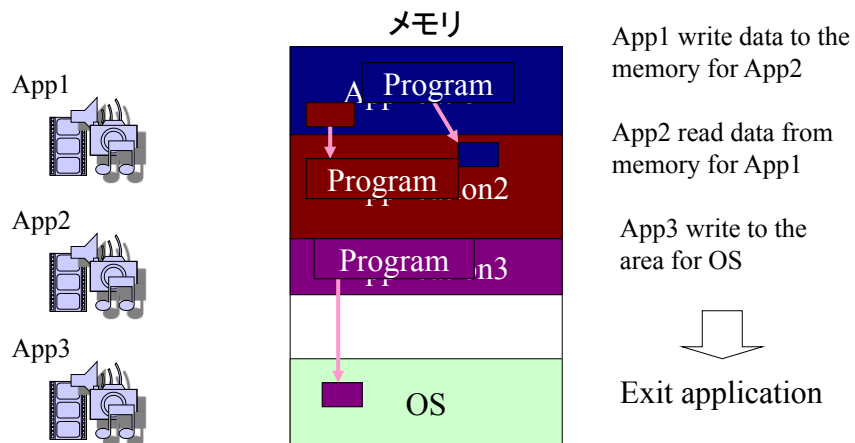


Control Program: Guarantee Performance



Control Program: Guarantee Reliability

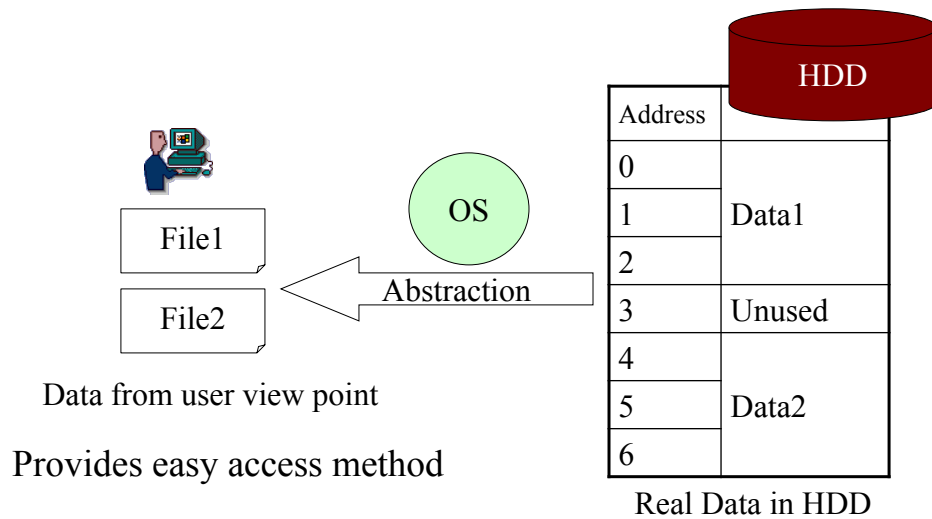
例 : Memory Protection



Resource Virtualization

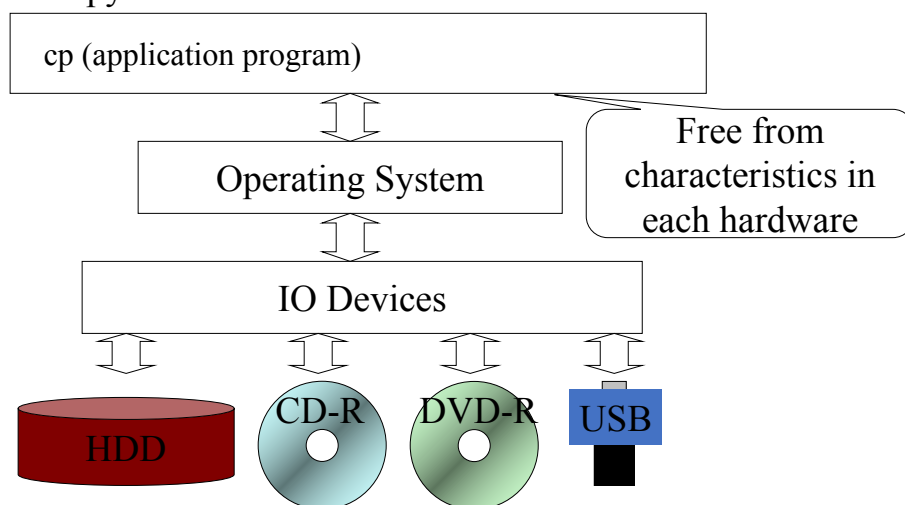
- Virtualize Hardware
- Provide access to the abstract concept
 - File, Directory..
- Programming Environment
 - Compiler, libc

Virtualization: Virtualize Hardware



Virtualization: Virtualize Hardware

ex : data copy



Provide access to the abstract concept

Hardware	Concept
Memory	Virtual Memory
Disk device	File
IO devices	
Network	Process
CPU	

Prevent direct access to hardware

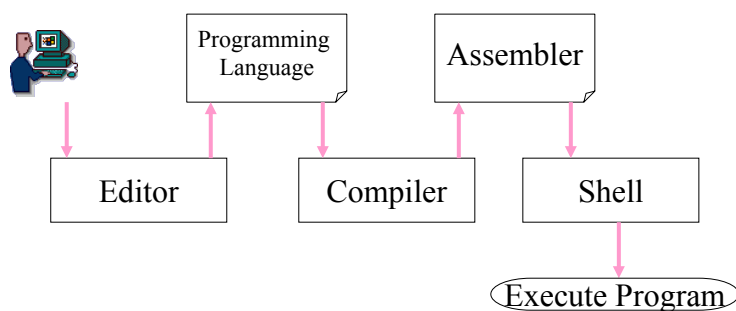


Avoid invalid access

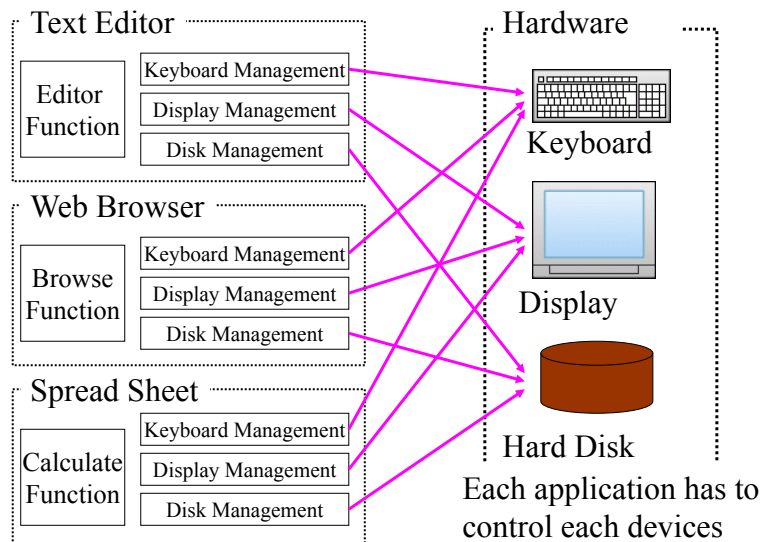
Easy operation

Programming Environment

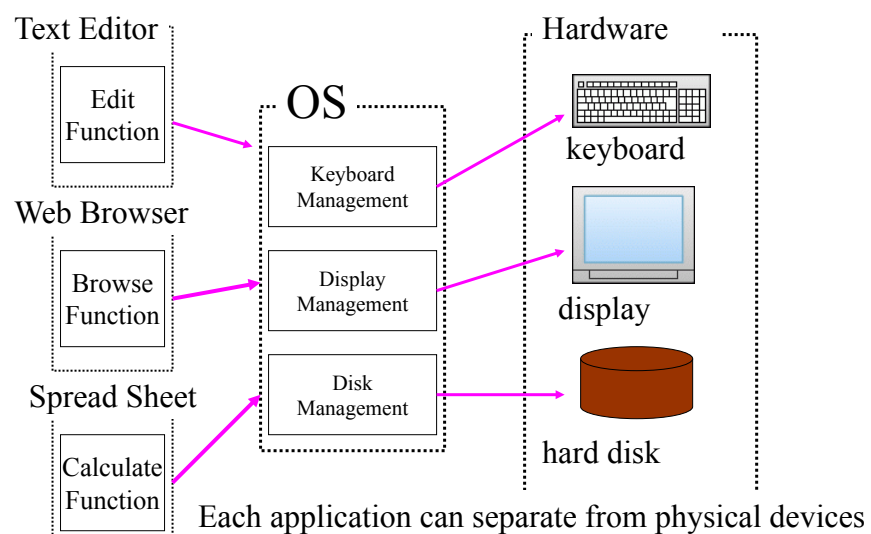
- Compiler, Editor, Shell



If OS does not exist...



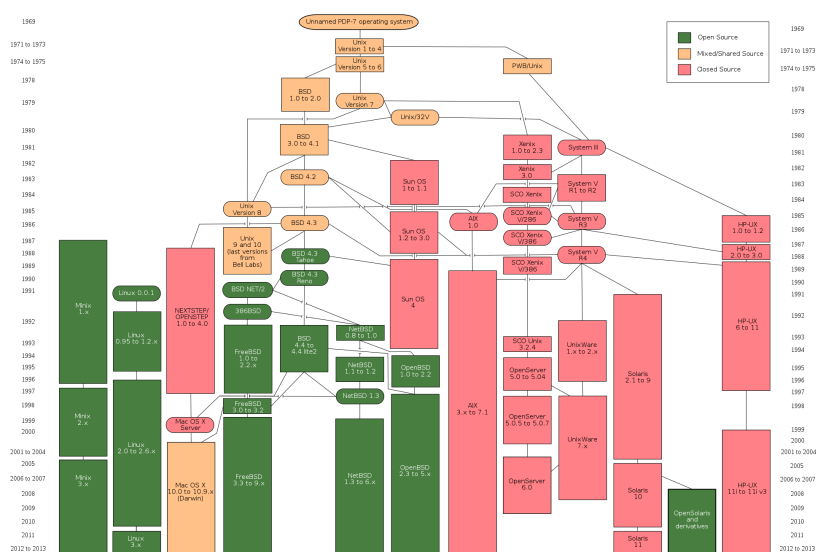
If OS exists..



Roll of Operating System

- Resource Management
 - Hardware/Software resource management
 - Assign each resources to users
- Control Program
 - Manage user program and hardware
 - Guarantee performance and reliability
- Resource Virtualization
 - Provides programming environment
 - Make it easy

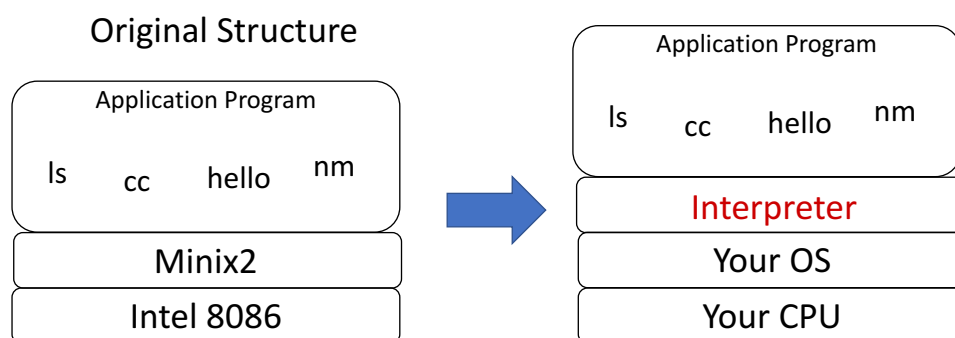
History of Operating System



Target : Minix2

- Developed for education by Andrew S. Tanenbaum
- Adopt micro Kernel architecture
- Open Source
- Running on 8086 CPU
- <https://minix1.woodhull.com/>

What we should do



Brief specification of Interpreter

- Read Minix Binary and execute it on your own environment
 - Emulate 8086 based operators
 - Trap system call and emulate their behavior

Deeply understanding...

- What is system call
- What is the virtualization
- Roll of Operating System
- Binary Specification