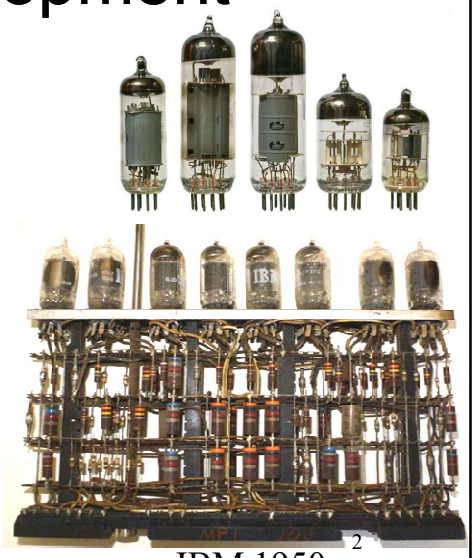
Principles of Operating Systems

Overview

1

History of OS development

- 45-55 period
  - Using vacuum tubes to process information and programming via plug board
  - Pioneered by H.Aiken, J.V.Neuman, J.P Eckert, W. Mauchley, K.Zuse ...
  - Programming language: not available
  - No OS yet exists



IBM 1950

2

... History of OS development

- 55-65 period
  - Using semiconductor technology (ex: IBM 1401)
  - Programs written on punch cards, magnetic tapes
  - Batch processing OS
  - Programming languages: assembly, fortran...

Punch card

Punch card reader

IBM1401

Printer

... History of OS development

- 65-80 period
  - Using integrated circuit (IBM 360)
  - Multi-task OS
  - The birth of Unix in '70

IBM 360 in Vietnam before 1975

5

... History of OS development

- Unix
  - Multics (Multiplexed Information & Computing Services)
  - Unics (Uniplexed) => Unix
- Unix is written in C and assembly language
  - C is the improvement of B language (BCPL)

Dennis Ritchie, Ken Thompson at PDP-11

6

... History of OS development

- 80-00 period
  - Using (Very) Large-scale Integration cuircuit (LSI, VLSI)
  - Personal PCs and OSs
  - Mid 80s: network OS (NOS) sharing common resources between computers

7

OS development trend

- Before ’80 – centralization trend
  - Grosch law: System performance is proportional to the square of cost
- ’80s – personal computation
  - Resources are dispersed on personal computers connected via networks
- Merging trend
  - Centralization via clouds

8

Components of a computer system

Application
Compiler – Editor
Operating System
Machine language
Micro program
Physical devices

9

What is an Operating System

- Definition: is system software that assists users to manage and control computer resources.
- An operating system consists of 2 independent components
  - The shells and
  - The kernel

10

Shells

- A Shell is a system program that interprets user's commands into system calls to the kernel and returns the outcome to the user
- Mulptiple shells can exist and run in parallel

11

The Kernel

- Contains the core code of an OS, that directly interacts with system resources. The kernel receives requests from the shell, executes the task and returns the outcome back to the shell
- The kernel is designed as an independent layer from the shells
- Only 1 kernel can take control in an OS

12

Processes

- Process: is a software program running on the computer with resources allocated by the operating system. Each process is managed by the operating system via an ID number (PID)
- A process may be suspended and another process will be running in turn
- A process may generate child processes
- Processes communicate with each other via signals (~ software interrupts). OS maintains a process table to keep current process states

13

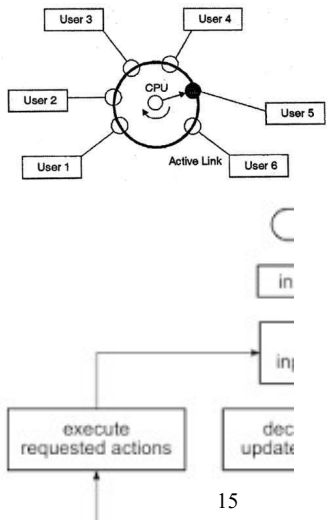
Single and multi-tasking

- Single task OS: only one process is allowed. One has to terminate before another can run.

14

... Single and multi-tasking

- Multi tasking OS: multiple processes are allowed by the OS
  - Preemptive (time sharing): processes are waiting for CPU in a round queue. A process is given a certain amount of CPU time for each round. When CPU time for a process runs out, CPU will be allocated to the next process in the round queue.
  - Non-preemtive: processes are waiting for CPU in a round queue. CPU will be allocated to each process in turn. A process is responsible to release CPU to the next process after finishing an event handling.
- Multiuser OS: multiple processes belong to multiple users are allowed



15

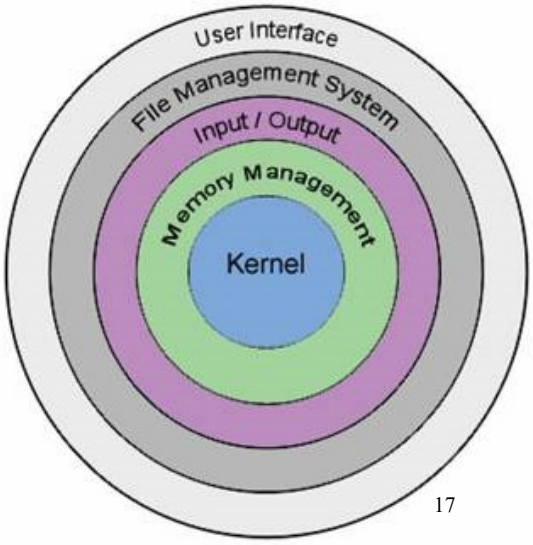
OS structure

- Monolithic approach
  - The kernel is a single large program providing services to applications via system calls
  - CPU runs on either user or kernel mode.
- Execution steps:
  - User applications make system calls which set parameters into registers and trap into kernel mode
  - The kernel checks the parameters and calls relevant services
  - The outcome is returned to user applications; CPU is switched back to user mode

16

Layered approach

- OS is designed into layers:
  - Layer 1: CPU allocation, multitasking management
  - Layer 2: Memory management
  - Layer 3: I/O; communication between processes
  - Layer 4: File management
  - Layer 5: User interface

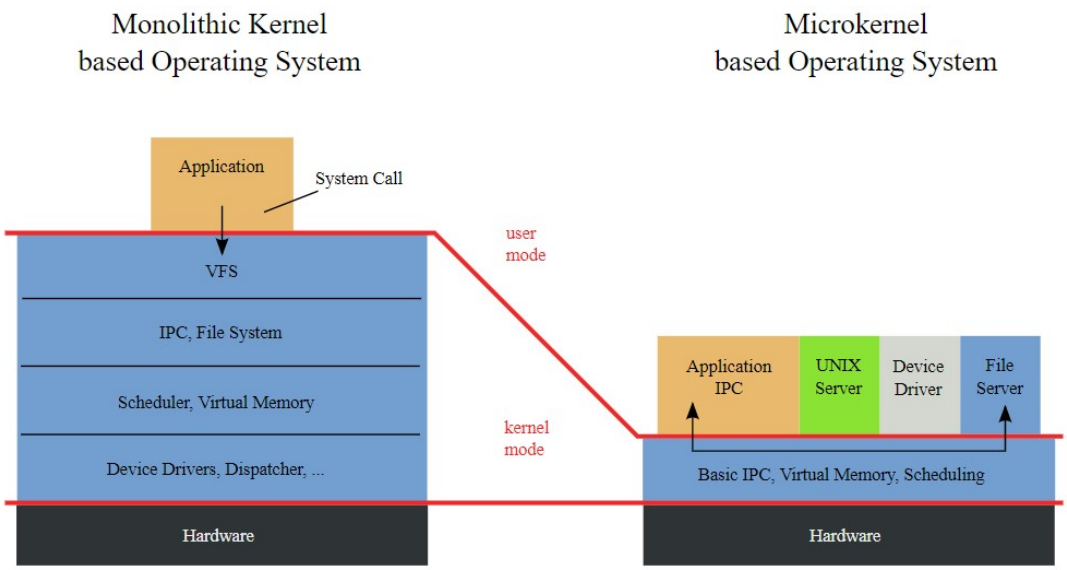


17

Monolithic vs Micro kernel

- Monolithic kernel
  - All 5 layers are designed into one single OS kernel and run in kernel mode
- Micro kernel
  - OS kernel only provides basic process and memory management services
  - Other extension services such as file management, process communication, device driver etc. are implemented as client-server services and run in user mode
  - Is appropriate for distributed system as it is easier to share workload between machines

18



19

Principles of Operating Systems

Course outline and references

20

Course outline

1. Processes
2. Memory management
3. File system
4. Input/Output
5. Deadlocks
6. A quick overview of distributed systems

21

Course materials

- Link: Qrcode
- Google classroom

  A4: gz2doix A7: llcn4jq

  A6: sxuhsuq A3: r4litle
- References: Modern Operating System, A. Tanenbaum 4th Edition

Lab tutorial

- Language: c / Linux
- Environment options:
  - Windows 10: install Ubuntu app, then
    - sudo apt-get update
    - sudo apt-get install gcc
  - Install vmware and then run ubuntu on the virtual machine
  - Install Cygwin terminal

23

…Lab tutorial

- Concurrent programming with processes and threads
- Using semaphores/monitors to handle resource race condition
- Memory management: allocation/deallocation, virtual memory
- File system: implement a file system (FAT, i-nodes)
- Deadlock: Banker's algorithm

24