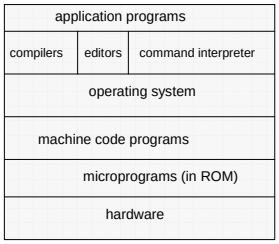
Operating Systems

An Overwiew

Operating System

- software for using the hardware
- computer resources
  - controls
  - shares
- program development environment
- kernel = operating system

Computer System



Operating System

- direct access to hardware not allowed
- user mode x kernel mode
- hides difficulties of using hardware
- interface between user and hardware
  - system calls

System Calls

- for user programs to
  - interact with operating system
  - get operating system to perform a task for them
- a library routine for every system call
- user program uses library routine

Operating System Responsibilities

- resource sharing
- virtual machine

Resource Sharing

- sharing among users
- security
  - isolate users
- shared resources:
  - CPU
  - memory
  - I/O components
  - data

Resource Sharing

- objectives:
  - to maximize resource utilization
  - to maximize resource availability

Resource Sharing

- provided services:
  - define user interface
    - system calls
  - sharing and usage control of resources in multi-use systems
    - prevent race for resources
    - mutual exclusion
  - allow users to share data (shared memory)
  - resource scheduling
  - I/O scheduling
  - error handling

Resource Sharing

- example:
  - users cannot share printer
  - possible to share screen

Virtual Machine

- as if single user
  - resource sharing transparent to user
- virtual machine may be different from actual physical machine:
  - I/O
  - memory
  - file system
  - protection and error handling
  - program interaction
  - program control

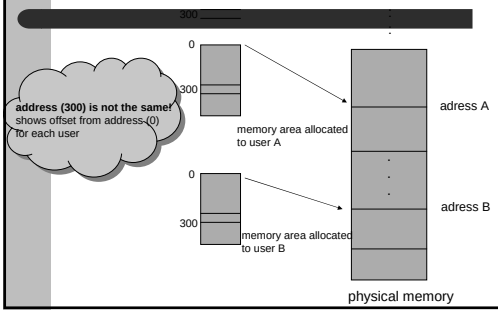
Virtual Machine

- I/O
  - requires hardware dependent programming
  - device drivers

Virtual Machine

- memory
  - different memory capacity in virtual machine
    - use disk as secondary memory
    - share among users
      - each user sees part of memory allocated to her

Virtual Machine



Virtual Machine

- file system
  - for longterm storage of program and data
  - on disk
  - use symbols to acces info instead of physical addresses
    - naming
      - all accessed as files in UNIX

Virtual Machine

- protection and error handling
  - isolate users in multi-user systems

Virtual Machine

- program interaction
  - in runtime
    - for example one program may use output of another program as input

Virtual Machine

- program control
  - provide user with high-level command set
    - shell commands
      - shell: command interpreter
      - shell not part of operating system
      - but shell uses system calls heavily

Types of Operating Systems

- mainframe operating systems
- server operating systems
- multi-processor operating systems
- PC operating systems
- real-time operating systems
- embedded operating systems
- smartcard operating systems

Mainframe Operating Systems

- for heavily I/O bound tasks
- three main services:
  - batch mode
    - non-interactive, routine tasks
    - e.g. preparing employee paychecks
  - transaction processing
    - e.g. airline reservation systems
  - time-sharing
    - multiple remote users running tasks
      - e.g. database
  - e.g.: OS/390

Server Operating Systems

- on servers
  - PCs with high resource capacities
  - workstations
  - mainframe systems
- services for multi-users over a network
  - hardware and software sharing
  - e.g: printer services, file sharing, web access
- e.g.: UNIX, Windows 2000

Multi-Processor Operating Systems

- for multi-processor systems
- to increase computing power
- based on interconnection between processors:
  - parallel systems
  - networked computers
  - multi-processor computers
- special operating system features required
  - design objectives similar to server operating systems
  - extra features for interconnection and communication between processors

PC Operating Systems

- efficient and easy to use interface
- office applicaitions
- e.g.:
  - Windows 98, 2000, XP
  - Macintosh
  - Linux

Real-Time Operating Systems

- time constraints important
- industrial control systems
  - feedback
- two types:
  - hard real-time
    - time constraints compulsory
    - e.g. robots in car production line
  - soft-real-time
    - possible not to obey some constraints
    - e.g. multimedia systems
- örnek: VxWorks ve QNX

Embedded Operating Systems

- palm computers and embedded systems
- limited operation
- special purpose
- e.g.: TV, microwave oven, cell phones, ...
- in some systems, size, memory and power consumption constraints
- e.g.: PalmOS, Windows CE

Smart-Card Operating Systems

- smallest operating system
- on credit card sized cards with processor
- strict memory and CPU constraints
- some are dedicated e.g. elektronic payments
- some may have several functionalities
- usually special purpose operating systems developed by card companies
- some Java based
  - possible to load and execute small JAVA programs (applet)
  - some may execute more than one applet
    - multi-programming, scheduling, resource sharing and protection

Main Kernel Architectures

- monolithic
- layered
- virtual machine
- exo-kernel
- server-client model
- modular

Monolithic

- no general structure
- all services and functionalities included in operating system
- all functional procedures
  - at the same level
  - may interact with each other
- large

Modular

- minimal kernel
- services added to kernel at runtime as they are needed
  - e.g. device drivers
- small kernel size
- slower
- e.g.: LINUX

Layered

- layered structure
  - hierarchical
- e.g.: THE operating system

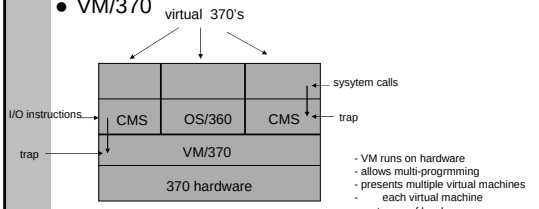
5	operator
4	user programs
3	I/O control
2	operator – process interaction
1	memory and disk control
0	CPU sharing and multi-programming

- layer 0: processor
- layer 1: memory management
- ...

Each layer independent of operations of layers below.
e.g.: for layer 2 operations, data may be on memory or disk

Virtual Machine

- VM/370



- VM runs on hardware
- allows multi-progrmming
- presents multiple virtual machines
- each virtual machine exact copy of hardware
- each virtual machine may have different operating system

Exo-Kernel

- developed at MIT
- similar to virtual machine concept
  - copy of system
  - difference: each virtual machie gets subset of system resources
- external kernel
  - controls that virtual machines do not exceed thir allocated resources
- each virtual machine may have different operating systems

Server-Client Model

- minimal kernel - micro-kernel
- most of operating system at user mode
- server and client processes
  - e.g. file read operation
    - client process asks from server process
    - server carries out operation
    - gives reply to client
- kernel coordinates communication and interaction between servers and clients

Server-Client Model

- servers in user mode
  - file server
  - process server
  - terminal server
  - memory server
- operating system consists of many smaller sub-units:
  - easy to manage
  - error in one does not affect others (units do not access hardware directly)
  - implementation problems: not possible to implement especially some I/O device drivers at user mode
- suitable for distributed systems