Operating Systems Structure

Section 1.5

Structure Categories

- Monolithic systems
  - The big mess
- Layered systems
  - THE, MULTIX
- Virtual machines
  - VM/370, JVM
- Exokernels
  - Modern Virtual machine applications
  - Client-server systems

Execution of a system call

Program runs in user mode
Service is needed
Program takes control
Execute a trap
Instruction to transfer control
OS carries out the system call
OS takes control
Instruction to transfer control
OS takes control

Steps to execute a system call

`read(fd, &buffer, nbytes)`

1-3 Calling program pushes parameters onto the stack in reverse order
4 Invoke system call
5 Puts the system call number in a register
Steps to execute a system call

**read(fd, buffer&, nbytes)**

6. Execute a trap to enter kernel mode

7. Dispatch the specified call handler

8. Specified call run to completion – if a wait occurs the OS may temporary transfer to another process

Monolithic structure

1. A main program that invokes the requested service procedure
2. A set of service procedures that carry out the system calls
3. A set of utility procedures that help the service procedures.

Layered System (THE)

- Operator process
- User program
- I/O Management
- Operator/Process Communication
- Memory management
- CPU management

9. Execute a trap to enter user mode

10. Return value is passed to user program

11. Clear stack by adjusting the stack pointer
Layered System (MULTICS)

- CPU Management
- Memory Management
- I/O Management
- User Program
- Operator program

Virtual machines

- Provides multiprogramming
- Provides interface of typical system calls
- Each CMS is a complete copy of the low level machine and can have its own OS

Operation of a VM

- CMS program system call is trapped by its own copy of the machine
- CMS issues hardware call that is trapped by VM layer that translates it into a real system call to the physical machine
- Many OS’s provide this feature as a different mode to support programs that were written for older versions of the OS

Exokernel

- Provide virtual hardware configuration
- Modern virtual machine
  - VirtualBox, Parallels, VMWare, QEMU, Virtual PC
Client - Server

- Provide only a minimal kernel = Server
- Provide services as user processes = Client
- Client processes run in user mode
- Client processes do not have direct access to hardware

Distributed client-server

- Client server model can easily be adapted for distributed systems
- System calls are sent via network communication for a remote machine to execute.
- Typically each client have a copy of a relevant portion of the kernel to be able to perform its own operations