OPERATING SYSTEMS

I/O Management, System Protection and Security

Arya S S
Department of Computer Science

OPERATING SYSTEMS

System Protection - Goals, Principles and Domain of Protection

Arya S S

Department of Computer Science

OPERATING SYSTEMS
Slides Credits for all PPTs of this course

- The slides/diagrams in this course are an **adaptation**, **combination**, and **enhancement** of material from the following resources and persons:

1. Slides of Operating System Concepts, Abraham Silberschatz, Peter Baer Galvin, Greg Gagne - 9th edition 2013 and some slides from 10th edition 2018
2. Some conceptual text and diagram from Operating Systems - Internals and Design Principles, William Stallings, 9th edition 2018
3. Some presentation transcripts from A. Frank – P. Weisberg
4. Some conceptual text from Operating Systems: Three Easy Pieces, Remzi Arpaci-Dusseau, Andrea Arpaci Dusseau

Goals of Protection

- In one protection model, computer consists of a collection of objects, hardware or software
- Each object has a unique name and can be accessed through a well-defined set of operations
- Protection problem - ensure that each object is accessed correctly and only by those processes that are allowed to do so
- Protection provides a mechanism for the enforcement of the policies governing resource use
  - Some policies are fixed in the design of the system, while others are formulated by the management of a system and by the individual users as well (to protect their own files and programs).

Principles of Protection

- Guiding principle – **principle of least privilege**
  - Programs, users and systems should be given just enough **privileges** to perform their tasks
  - Failure of a component does the minimum damage and allows the minimum damage to be done.
  - OS follows this principle for its features, programs, system calls and data structures
  - Can be static (during life of system, during life of process)
  - Or dynamic (changed by process as needed) – **domain switching**, **privilege escalation**

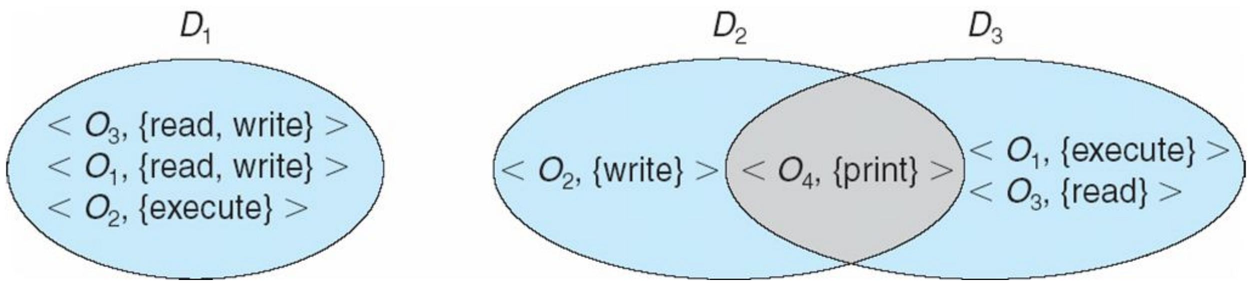
Principles of Protection (Cont.)

- Must consider "grain" aspect
  - Rough-grained privilege management easier, simpler, but least privilege now done in large chunks
    - For example, traditional Unix processes either have abilities of the associated user, or of root
  - Fine-grained management more complex, more overhead, but more protective
    - Provide/disable privileges as needed
    - Create audit trail for privileged function access
    - File ACL lists, RBAC

Domain of Protection

- A computer system is a collection of processes and objects.
  - Hardware objects (CPU, memory segments, printers, disks, and tape drives)
  - Software objects (files, programs, and semaphores)
- Each object has a unique name and can be accessed only through well-defined and meaningful operations.
- A process should be allowed to access only those resources for which it has authorization.
- A process should be able to access only those resources that it currently requires to complete its task.(**need-to-know principle**)
- Need-to-know principle, is useful in limiting the amount of damage a faulty process can cause in the system

Domain Structure

- A process operates within a **protection domain**, which specifies the resources that the process may access.
- Each domain defines a set of objects and the types of operations that may be invoked on each object.
- The ability to execute an operation on an object is an **access right**
- **Access-right=<object-name, rights-set>**
  where rights-set is a subset of all valid operations that can be performed on the object.
- 

System with 3 protected domains

Domain Structure(Cont.)

- The association between a process and a domain may be either static or dynamic.
- **Static**: the set of resources available to the process is fixed throughout the process's lifetime.
    - But a process may execute in two different phases, for eg ,need read access in one phase and write access in another phase.
    - So the fixed domain should contain both read and write access for that process.
    - This violates need to know principle.
    - So we must allow the contents of domain to be modified during each phase of the process.
- If the association is **dynamic**, a mechanism is available to allow domain switching.

Domain Structure (Cont.)

- A domain can be realized in a variety of ways:
    - User
    - Process
    - Procedure
- Standard dual-mode (monitor-user mode) model of operating-system execution is followed in system.
- But in a multi programming environment these two protection domains are insufficient, since users want to be protected from one another.
- So a more elaborate scheme is used in UNIX and MULTICS.

THANK YOU

Arya S S

Department of Computer Science Engineering

aryadeep@pes.edu