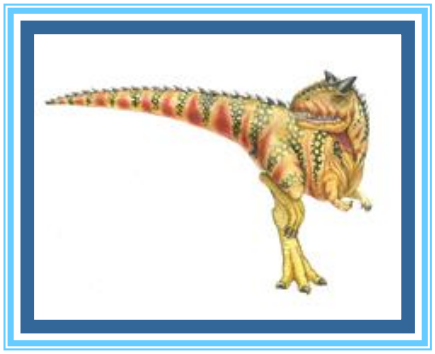
Protection & Security



Chapter 14: Protection

- Security and Protection - Goals, Principles in normal OS for security,
- Access Control models and methods.

Goals of Protection

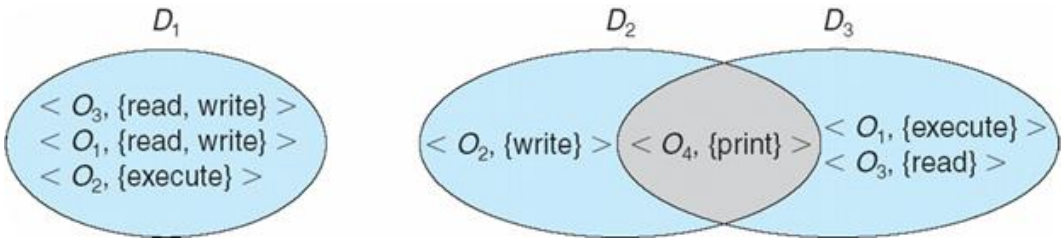
- protection is a method of safeguarding data and processes against malicious and intentional intrusion.
- Protection model
  - Computer consists of a collection of objects (hardware or software)
  - Each object has a unique name
  - Objects can be accessed through a well-defined set of operations
- Protection problem - ensure that each object is accessed correctly and only by those processes that are allowed to do so
  - Controlling shared access
  - Implementing an interface to allow shared access
  - Identification and authentication
  - Naming and communication among processes
  - Scheduling and reclaiming and reusing objects

Principles of Protection

- For protection we have protection policies
  - How the processes are to access the resources (CPU, memory, software and even the OS) present in the computer system,
  - Guiding principle – **principle of least privilege**
    - Programs, users and systems should be given just **enough privileges** to perform their tasks
    - Limit the access of each process with respect to their resource handling.
    - A process is bound to use only those resources which it requires to complete its task, in the time limit that it requires and also the mode in which it is required.
  - Limits damage if entity has a bug, gets abused
  - Can be static (during life of system, during life of process)
  - Or dynamic (changed by process as needed) – **domain switching**, **privilege escalation**

Domain Structure

- Access-right = <*object-name*, *rights-set*>
  where *rights-set* is a subset of all valid operations that can be performed on the object
- Domain = set of access-rights (a set of objects and the operations that can be performed on them)
- A domain element is described as <object, {set of operations on object}>.
- A domain can consist of a process or a procedure or a user.
- Then, if a domain corresponds to a procedure, then changing domain would mean changing procedure ID.
- Objects may share common operations. Then the domains overlap.

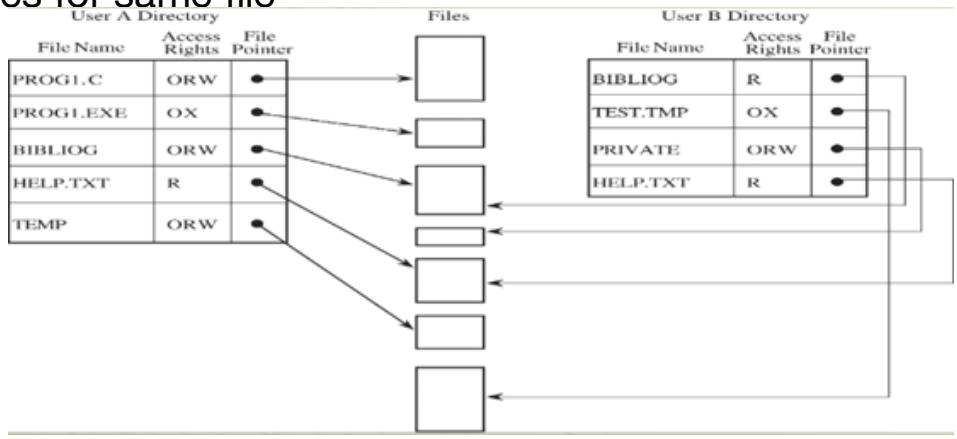


Object Control Mechanisms

- Directory

  Each user (subject) has a file directory, which lists all files accessible by user - User has a list to determine access to an object

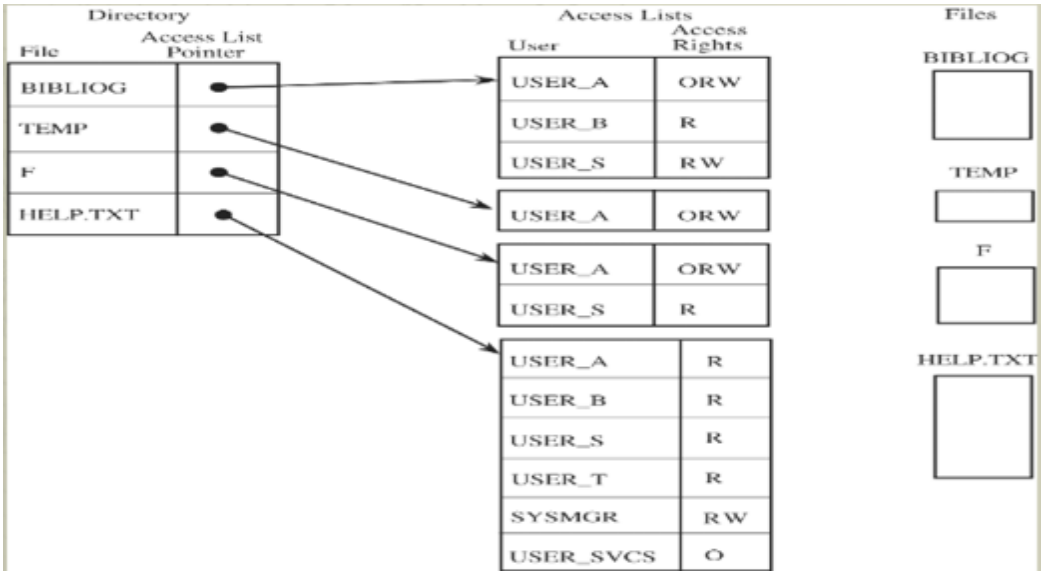
  Problem – large lists if many shared objects, revocation of access is time consuming , Cannot revoke rights of everyone to an object and different users use different names for same file



Object Control Mechanisms

- Access Control List

  One list for each object - The list showing all subjects & their access rights



Access Matrix

- View protection as a matrix (**access matrix**)
- Rows represent domains
- Columns represent objects
- **`Access(i, j)`** is the set of operations that a process executing in $Domain_i$ can invoke on $Object_j$

domain \ object	F_1	F_2	F_3	printer
D_1	read		read	
D_2				print
D_3		read	execute	
D_4	read write		read write	

Use of Access Matrix

- If a process in Domain D_i tries to do "op" on object O_j, then "op" must be in the access matrix
- User who creates object can define access column for that object
- Can be expanded to dynamic protection
  - Operations to add, delete access rights
  - Special access rights:
    - *owner of* O_i
    - *copy op from* O_i *to* O_j *(denoted by "*")*
    - *control* – D_i *can modify* D_j *access rights*
    - *transfer – switch from domain* D_i *to* D_j
  - *Copy* and *Owner* applicable to an object
  - *Control* applicable to domain object

Access Matrix of Figure A with Domains as Objects

domain \ object	F_1	F_2	F_3	laser printer	D_1	D_2	D_3	D_4
D_1	read		read			switch		
D_2				print			switch	switch
D_3		read	execute					
D_4	read write		read write		switch			

Access Matrix with *Copy* Rights

domain \ object	F_1	F_2	F_3
D_1	execute		write*
D_2	execute	read*	execute
D_3	execute		

(a)

domain \ object	F_1	F_2	F_3
D_1	execute		write*
D_2	execute	read*	execute
D_3	execute	read	

(b)

Access Control Models

Access control is the process of:

- Identifying a person
- Authenticating them by looking at their identification
- Granting a person only the key to the door or computer that they need access to and nothing more
- Granting access via a username and password
- allowing them access to files, computers, or other hardware or software they need
- ensuring they have the right level of permission to do their job
- Access control models have four types:
  a. Mandatory Access Control (MAC)
  b. Role-Based Access Control (RBAC)
  c. Discretionary Access Control (DAC)
  d. Rule-Based Access Control (RBAC or RB-RBAC)

Access Control Models

- **The Mandatory Access Control, or MAC,** model gives only the owner and custodian management of the access controls.
- End user has no control over any settings that provide any privileges to anyone.
- **The Role-Based Access Control, or RBAC,** model provides access control based on the position an individual fills in an organization.
- **The Discretionary Access Control, or DAC,** model is the least restrictive model compared to the most restrictive MAC model. DAC allows an individual complete control over any objects they own along with the programs associated with those objects.
- **Rule-Based Access Control**, will dynamically assign roles to users based on criteria defined by the custodian or system administrator. For example, if someone is only allowed access to files during certain hours of the day, we can set up Rules accordingly

Security

System is **secure** if resources used and accessed as intended under all circumstances

Intruders (**crackers**) attempt to breach security

Threat is potential security violation
Attack is attempt to breach security
Example violations

Breach of confidentiality - Unauthorized reading of data

Breach of integrity - Unauthorized modification of data

Breach of availability - Unauthorized destruction of data

Theft of service - Unauthorized use of resources

Denial of service (DOS) - Prevention of legitimate use

Cryptography - a security tool (Encryption, Hashing . Digital Signatures)