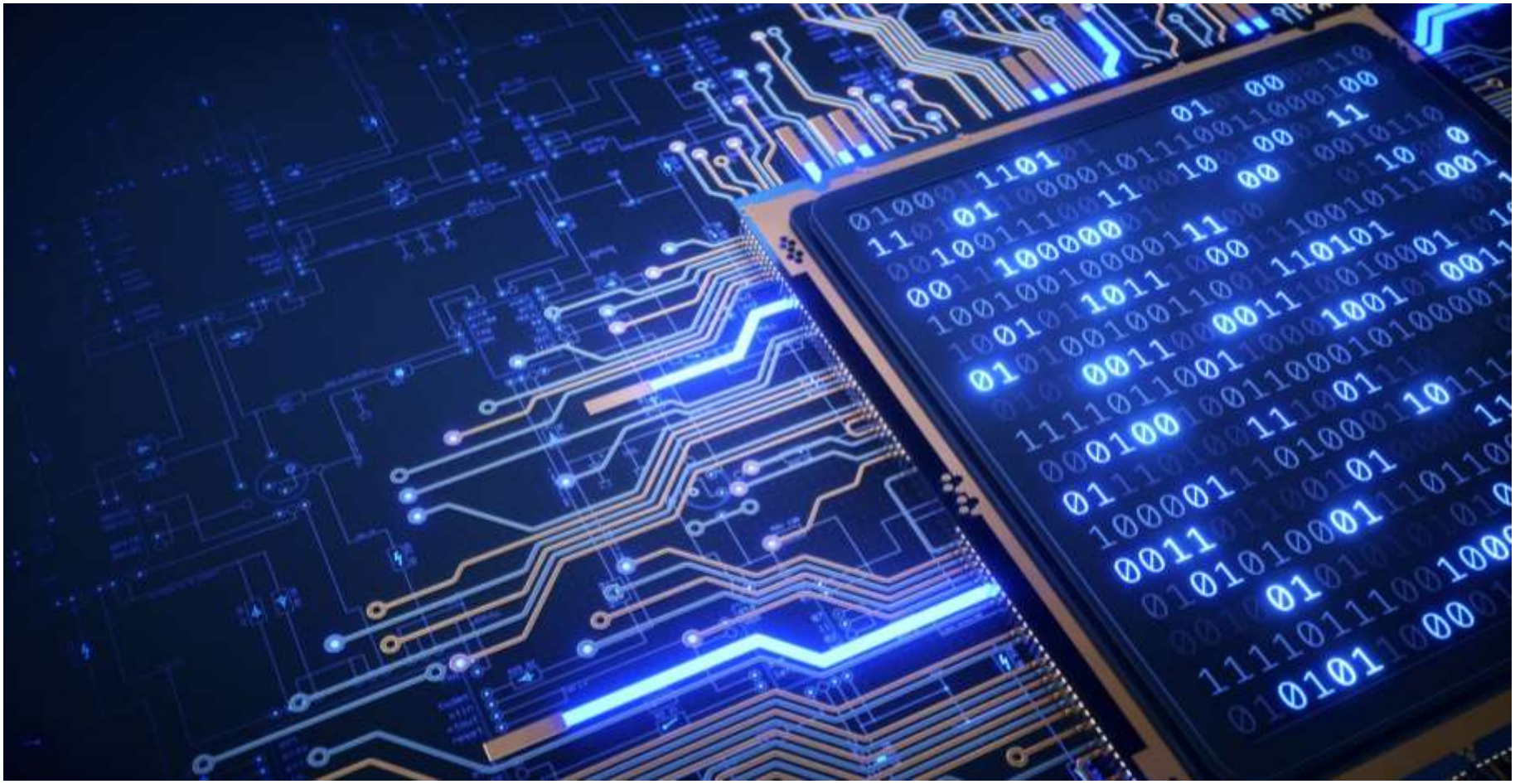


Chapter 3: Access Control

Information Security

Nguyễn Đăng Quang

Goals

01

Understand the importance of Access Control

02

Explore ways in which Access Control can be implemented

03

Understand how Access Control is implemented

Access Control

Mechanisms for enforcing separation of users and processes on a system.

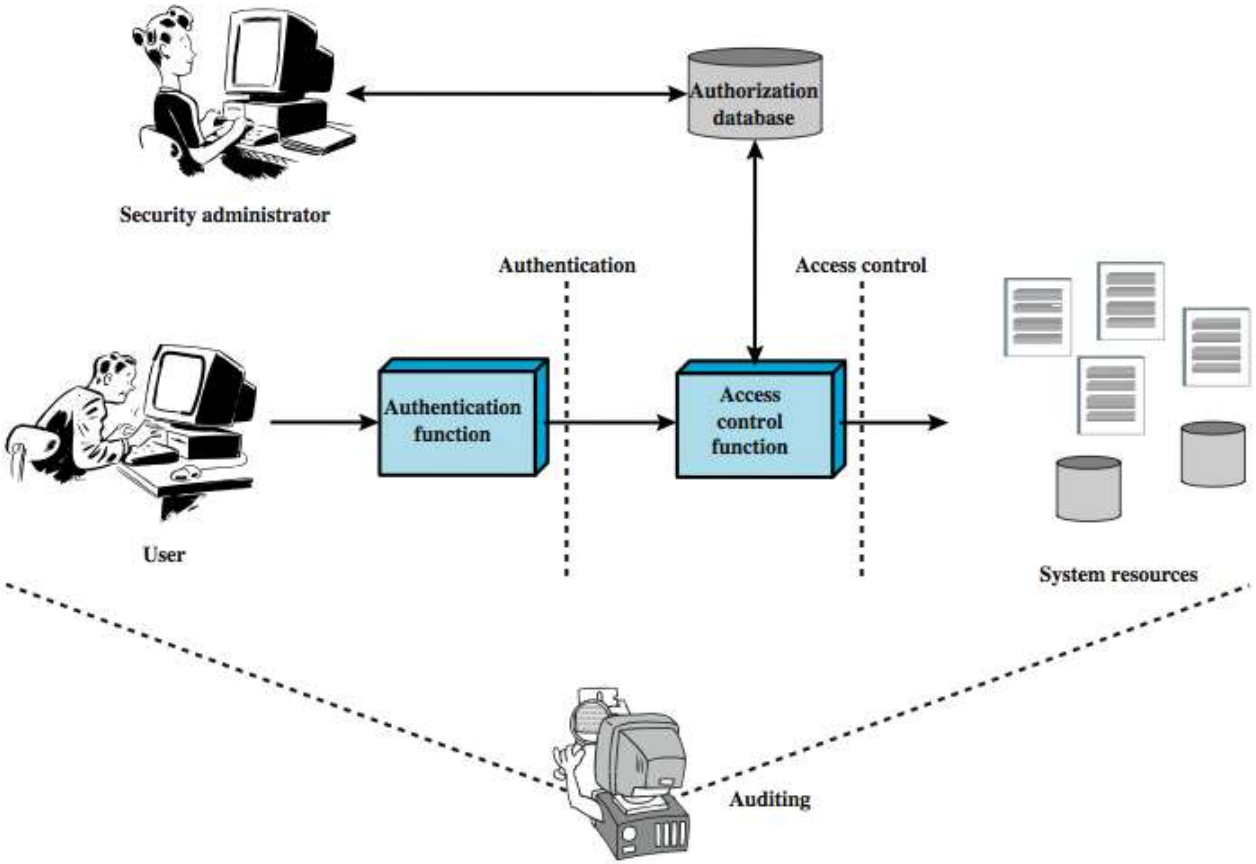
Include user accounts, groups, file ownership, file permissions...

Central element of computer security.

Authentication vs Authorization

- Authentication — Are you who you say you are?
  - Restrictions on who (or what) can access system
- **Authorization** — Are you allowed to do that?
  - Restrictions on actions of authenticated users
- Authorization is a form of **access control**
- Classic view of authorization…
  - Access Control Lists (ACLs)
  - Capabilities (C-lists)

Access Control Principles



Access Control Requirements

- Reliable input: a mechanism to authenticate
- Fine and coarse specifications: regulate access at varying levels (e.g., an attribute or entire DB)
- Least privilege: min authorization to do its work
- Administrative policies: who can add, delete, modify rules

Access control policies

- **Discretionary** access control (DAC): based on the discretion of the data owner.
- **Mandatory** access control (MAC): A system-wide access policy.
- **Role-based** access control (RBAC): based on user roles.
- **Rule-based** access control: based on a set of predefined rules

Discretionary Access Control

Based on the **Discretion** of data owner:

- Owner or creator of the resource specifies which subjects have which access to a resource.
- Implemented in commercial Windows, Linux, Mac

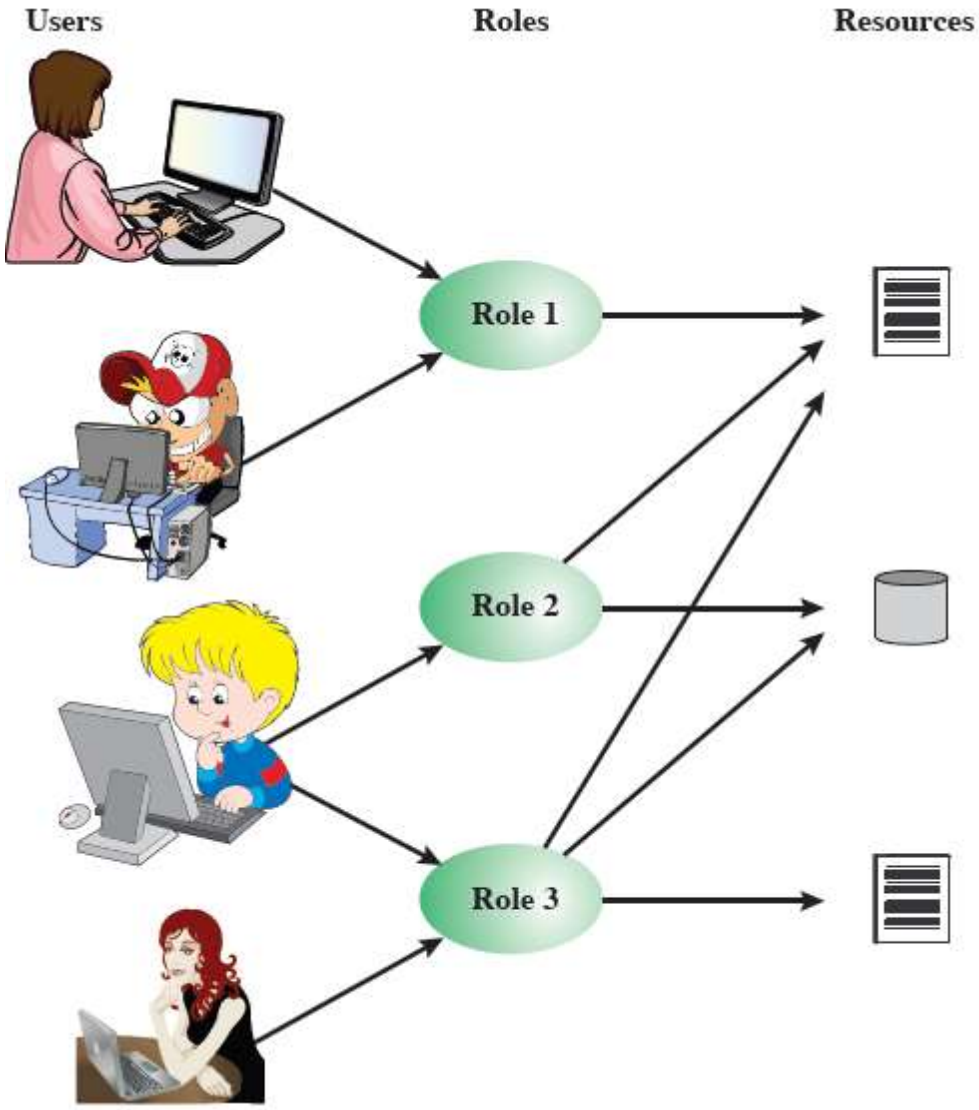
Mandatory Access Control

- OS makes the decision based on a security label system.
- Data owner cannot grant access.
- Users and Data are given a clearance level (confidential, secret, top secret).
- Rules for access are configured by the security officer and enforced by the OS.
- e.g., SELinux, Trusted Solaris, TrustedBSD …

Role-Based Access Control

- Also called **Non-discretionary** access control.
- Access based on role not identity.
- Users are assigned a role, the roles dictates access to a resources, rather than the user.
- Groups in Windows and Linux are examples

Role-Based Access Control



Rule-based Access Control

- Uses specific rules that indicate what users can/cannot access objects.
- Before a subject can access an object, it must meet a set of predefined rules.

  Example: if user has a proper clearance and it's between 9AM-5PM then allow access.
- Is considered a "compulsory control" because rules are enforced and not modifiable by users.
- Routers and FWs use Rule-based access control.

  iptables –A INPUT –p –tcp –dport 80 –j DROP

Access Control Elements

Subject: entity that can access objects	Object: access-controlled resource	Access right: way in which subject accesses an object
• a process representing user/application • often have 3 classes: **owner**, **group**, **world**	• e.g., files, directories, records, programs etc. • number/type depend on environment	e.g., read,write,execute, delete, create, search

Access Control Matrix (ACM)

Contains the information relevant to access control, in which

Row corresponds to sources of the request: users/group/objects,

Column corresponds to resource that need to be protected.

ACM[U,O] state captured who has access to the resources of the system.

In a large system, the matrix will be enormous in size and mostly sparse

An Access Matrix

- Access matrix is often large, but sparse
- Can decompose by either row or column

		OBJECTS			
		File 1	File 2	File 3	File 4
SUBJECTS	User A	Own Read Write		Own Read Write	
	User B	Read	Own Read Write	Write	Read
	User C	Read Write	Read		Own Read Write

Access Control Structures

Access control lists (decomposed by column) for an object Oi is [(ui1,rights1),(ui2,right2),...]

Use to look by objects

Advantage

- Easy to determine who can access a given object.
- Easy to revoke all access to an object

Disadvantage

- Difficult to know the access right of a given subject.
- Difficult to revoke a user's right on all objects.

Used by most mainstream operating systems.

Access Control Structures

Capability tickets (decomposed by row) for a user Ui is [(Oi1,rights1),(Oi2,right2),...]

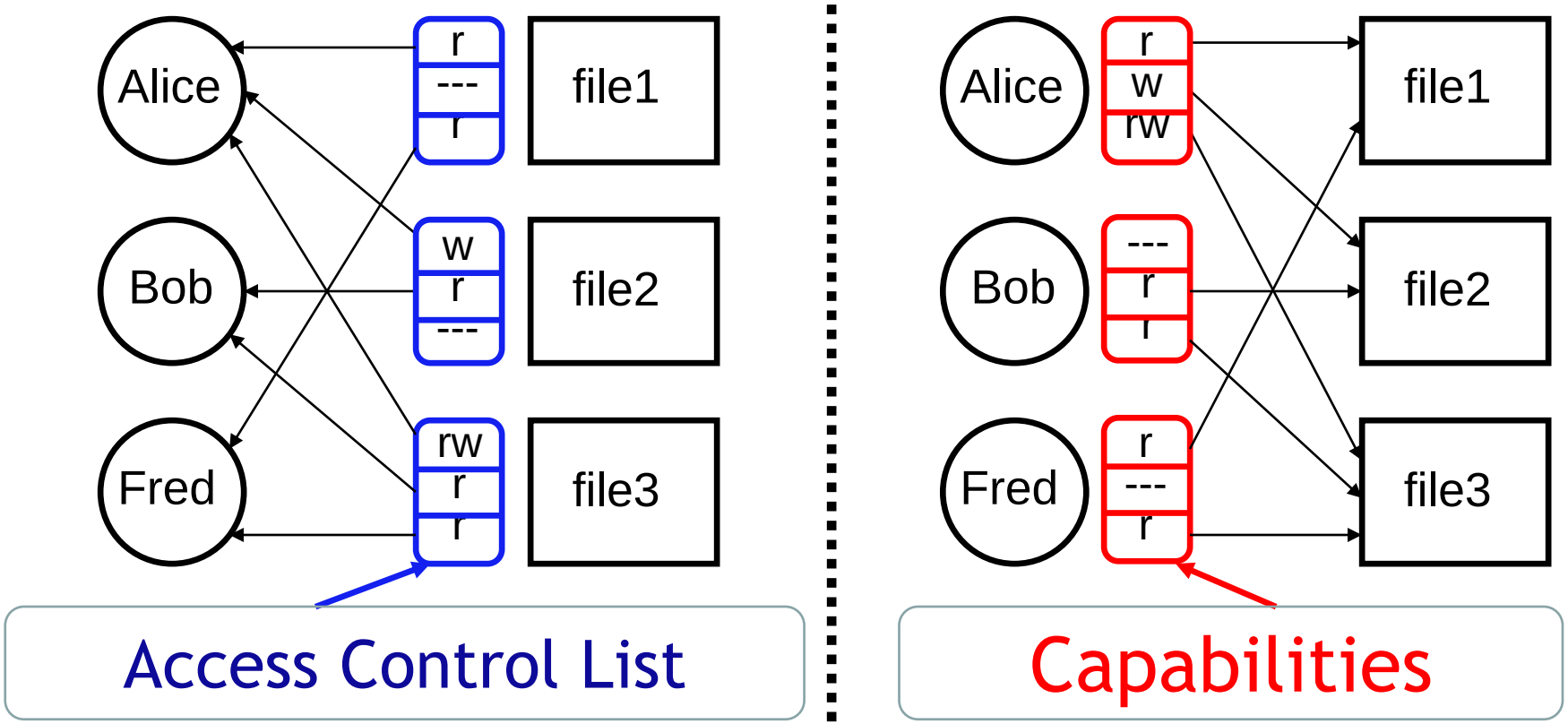
Advantage

- Easy to know the access right of a given subject.
- Easy to revoke a users access right on all objects.

Disadvantage

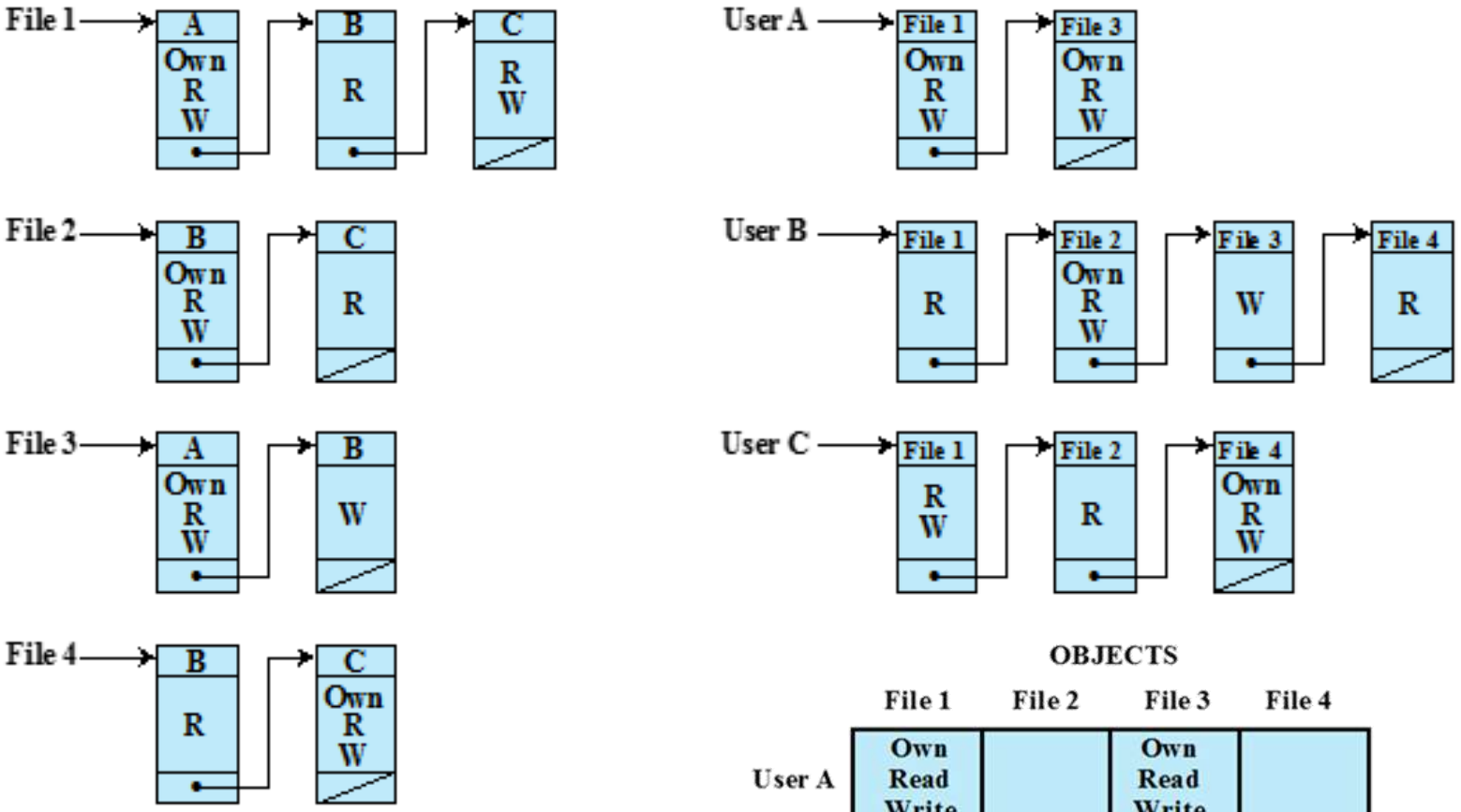
- Difficult to know who can access a given object.
- Difficult to revoke all access right to an object.

ACLs vs Capabilities



- Note that arrows point in opposite directions…
- With ACLs, still need to associate users to files

Access matrix data structures



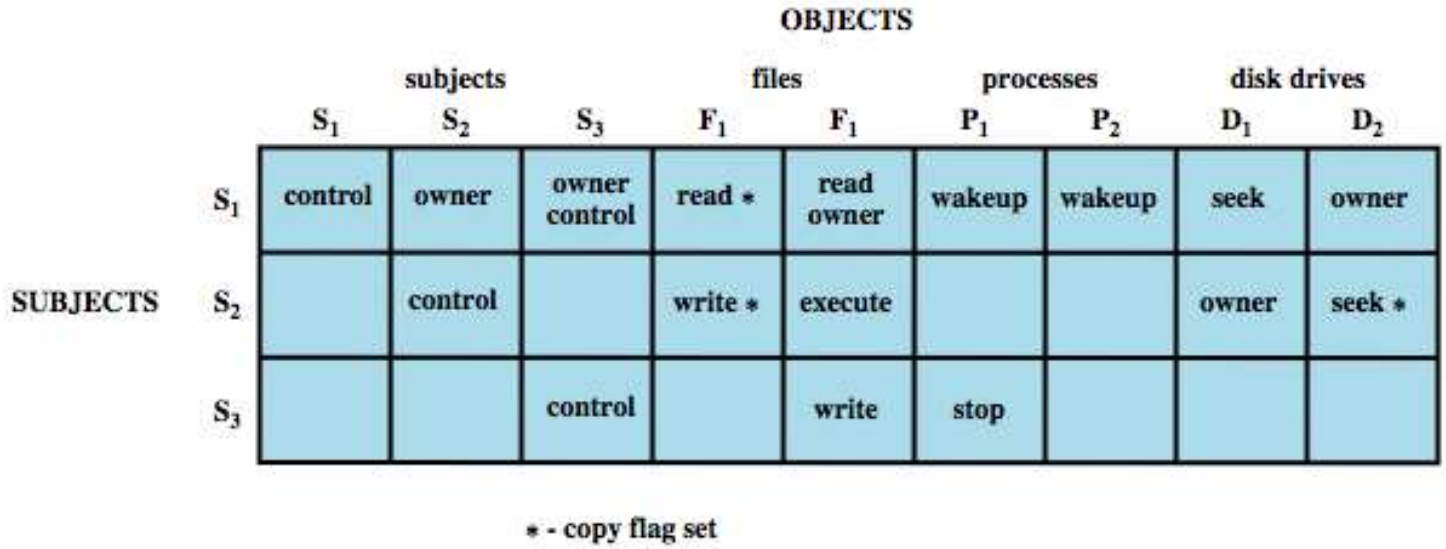
OBJECTS

SUBJECTS

	File 1	File 2	File 3	File 4
User A	Own Read Write		Own Read Write	
User B	Read	Own Read Write	Write	Read
User C	Read Write	Read		Own Read Write

Alternate Authorization Table

Subject	Access Mode	Object
A	Own	File 1
A	Read	File 1
A	Write	File 1
A	Own	File 3
A	Read	File 3
A	Write	File 3
B	Read	File 1
B	Own	File 2
B	Read	File 2
B	Write	File 2
B	Write	File 3
B	Read	File 4
C	Read	File 1
C	Write	File 1
C	Read	File 2
C	Own	File 4
C	Read	File 4
C	Write	File 4



		OBJECTS								
		subjects			files		processes		disk drives	
		S_1	S_2	S_3	F_1	F_1	P_1	P_2	D_1	D_2
SUBJECTS	S_1	control	owner	owner control	read *	read owner	wakeup	wakeup	seek	owner
	S_2		control		write *	execute			owner	seek *
	S_3			control		write	stop			

* - copy flag set

An Access Control Model

Extend the universe of objects to include processes, devices, memory locations, subjects

UNIX File Access Control

1. Unique user identification number (user ID)
2. Member of a primary group identified by a group ID
3. 12 protection bits
   - 9 specify read, write, and execute permission for the owner of the file, members of the group and all other users
   - 2 specify SetUID, SetGID
   - 1 is the sticky bit (only owner can remove, delete, …, a directory)

Extra			owner			group			others		
su	sg	t	r	w	x	r	w	x	r	w	x

UNIX File Access Control

- “set user ID”(SetUID) or “set group ID”(SetGID)
  - System temporarily uses rights of the file owner/group in addition to the real user’s rights when making access control decisions
  - Enables privileged programs to access files/resources not generally accessible
- Sticky bit
  - on directory limits rename/move/delete to owner
- Superuser
  - is exempt from usual access control restrictions

Unix Access Control

2 most important user id of a process:

- Real user ID (uid): the real owner of the process (the user running the process),
- Effective user ID (euid): the ID used in access control (what privilege the process has)

Unix Access Control

- For a non-setuid program, when it is executed by a user with uid 1000, its process's real and effective user id are the same, both being 1000.
- For a setuid program executed by the same user, the real uid will still be 1000, but the effective uid will depend on the user that own the program.

Summary

Introduced access control policies:

DAC, MAC, RBAC, Rule-based

Introduced AC elements, AC Matrix:

Subjects, Objects, Access rights

AC Data Structure

Access Matrix, Access Control Lists (ACLs), Capability tickets (C-List)

UNIX traditional and ACL mechanisms