Access control
Chapter 6

Access control

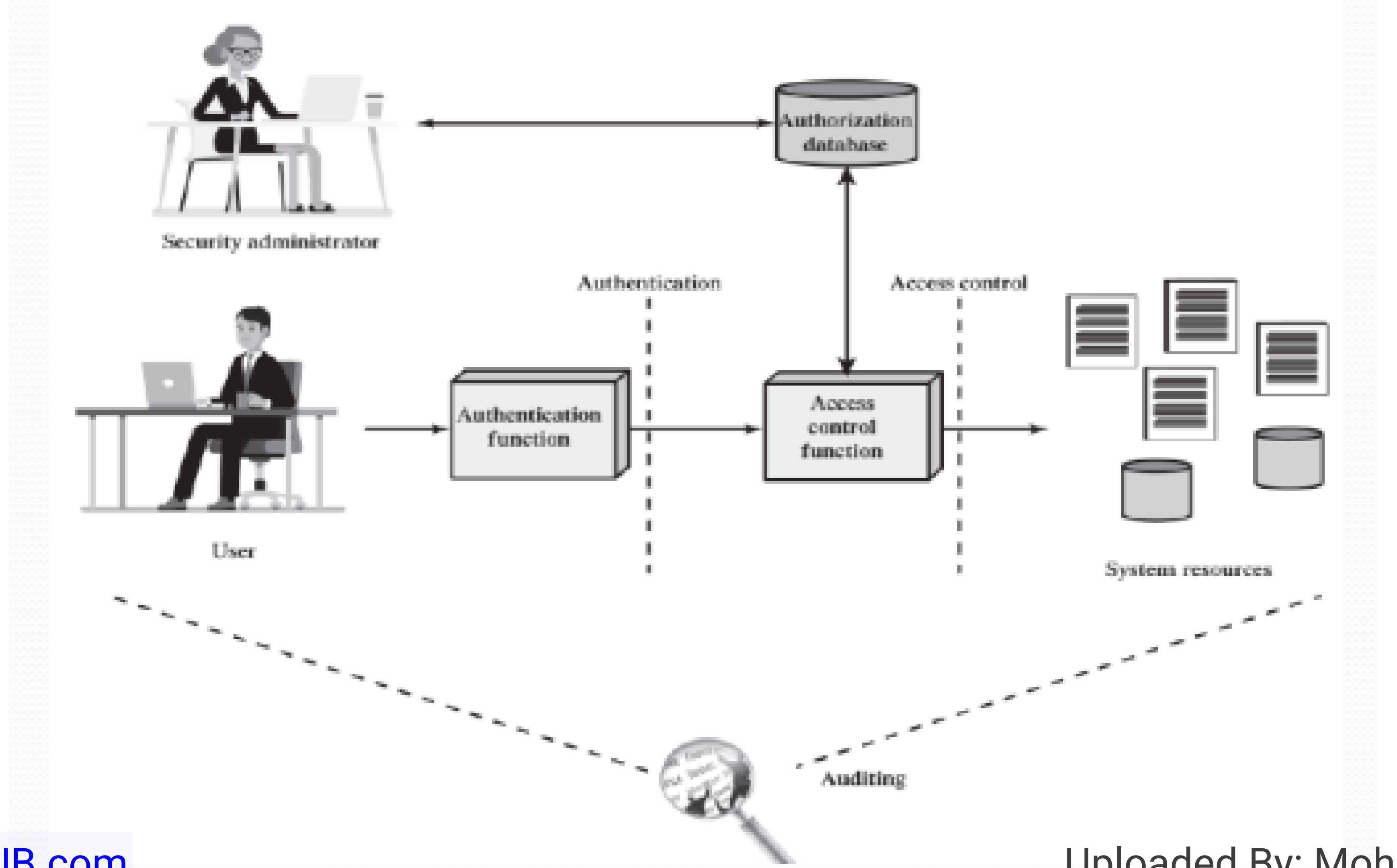
-The main purpose is to control who can do what on a system

-The prevention of unauthorized use of a system resource

-System resources, such as applications, operating systems, firewalls, routers, files, and databases.

-Using a system does NOT mean someone can do what he/she likes

Access control



Access Control and Security Functions

Authentication: **verification that the identity of a user or other entity are valid.**

Authorization **is the process of giving someone permission to do or access something**

Auditing **is an independent check of system records and activities to:**

1.Make sure the controls are good enough.

2.Ensure everything follows the rules.

3.Find any issues or breaches.

4.Suggest improvements

Access Control policies

Access control policies are generally grouped into the following categories:

Discretionary access control (DAC).

Mandatory access control (MAC).

Role-based access control (RBAC).

Attribute-based access control (ABAC).

Access Control policies

Discretionary access control (DAC):

is a way to control who can access something based on their identity and set rules.

Users(not necessarily to be the security admin) can also give permission to others to access resources

Mandatory access control (MAC).

is a system where access is determined by comparing security labels with security clearances. Users cannot give access to other

Access Control policies

Role-based access control (RBAC):

Based on roles of users in a system, and rules for roles are used to control access.

Attribute-based access control (ABAC):

decides who gets access by looking at things like user details, the resources being accessed, and what's happening right now

Elements of Access Control System

Subject : entity capable of access resources

such as user , application

Object : resource to which access is controlled

such as file ,program

Access right: describes way in which a subject may access an object

Such as write , read ,create

Elements of Access Control System

Owner : **This may be the creator of a resource, such as a file. For system resources, ownership may belong to a system administrator. For project resources, a project administrator may be assigned ownership**

Group : **A named group of users may also be granted access rights. In most schemes, a user may belong to multiple groups**

World : **The least amount of access is granted to users who are able to access the system, but are not included in the categories owner and group for this resource**

Requirements of Access Control System

- **Reliable input:** Making sure information is real and trustworthy.

- **Fine and coarse specifications:** Rules for controlling access, from detailed to general.

- **Least privilege:** Giving only the necessary permission for a job.

- **Separation of duty:** Sharing tasks among different people.

- **Open and closed policies:** Closed means you can only access what's allowed, while open means you can access almost everything unless it's forbidden.

- **Administrative policies:** Rules about who can change access rule

Requirements of Access Control System

implemented using an access matrix

- lists subjects in one dimension (rows)
- lists objects in the other dimension (columns)
- each entry (cell) specifies access rights of the specified subject to that object
- Can decompose by either row or column

Requirements of Access Control System

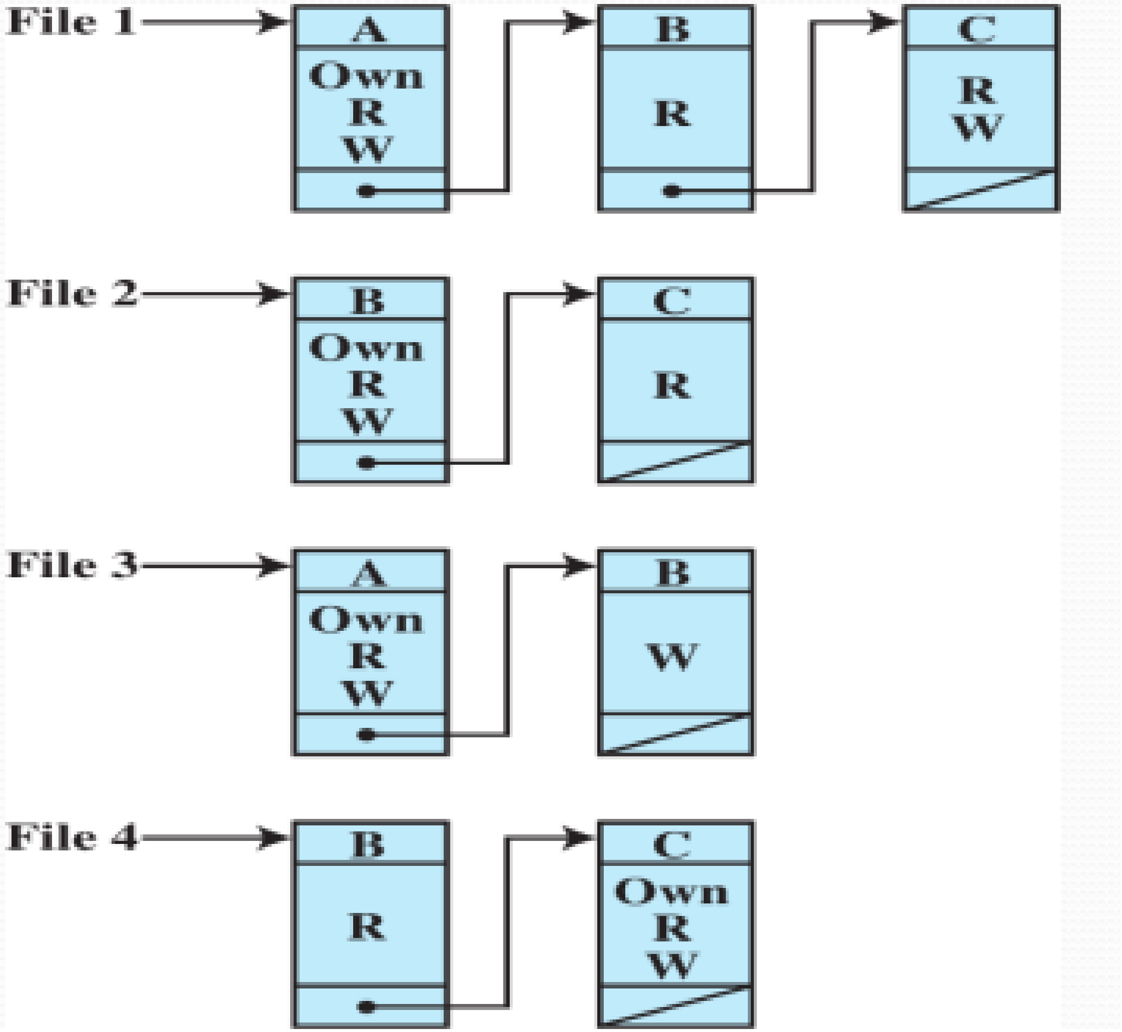
Access Control Lists (ACL):Think of it like a list attached to each object (like a file or folder) that says who can do what with it.Example: For "File_1.txt," the list might say "User_A can read, User_B can write, User_C can read and write.

Capability Lists :Imagine a list for each person (subject) that says what they can do with each object.
Example: For "User_A," their list might say "Can read File_1.txt, can't write, can't read Directory_1.

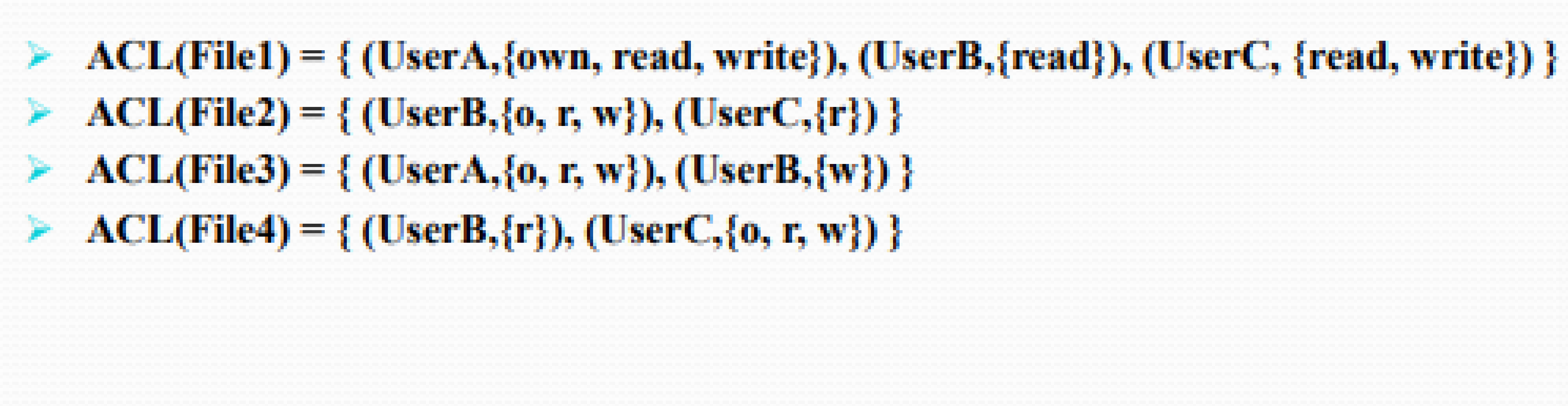
Authorization Tables: Listing subject, access mode and object; easily implemented in database.

Access Control Lists (ACL):

- One list for each object.
- ACL more efficient than access matrix.



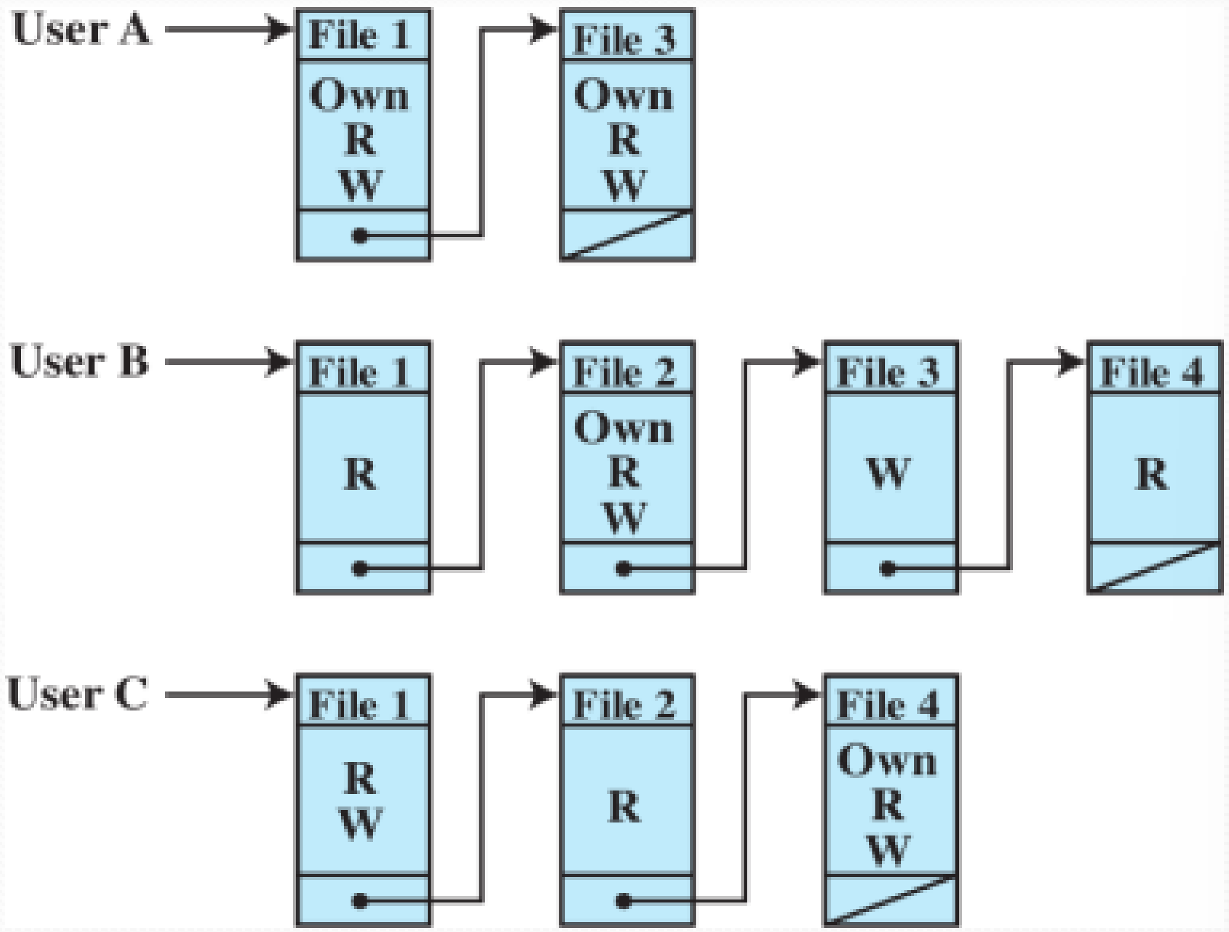
Access Control Lists (ACL):



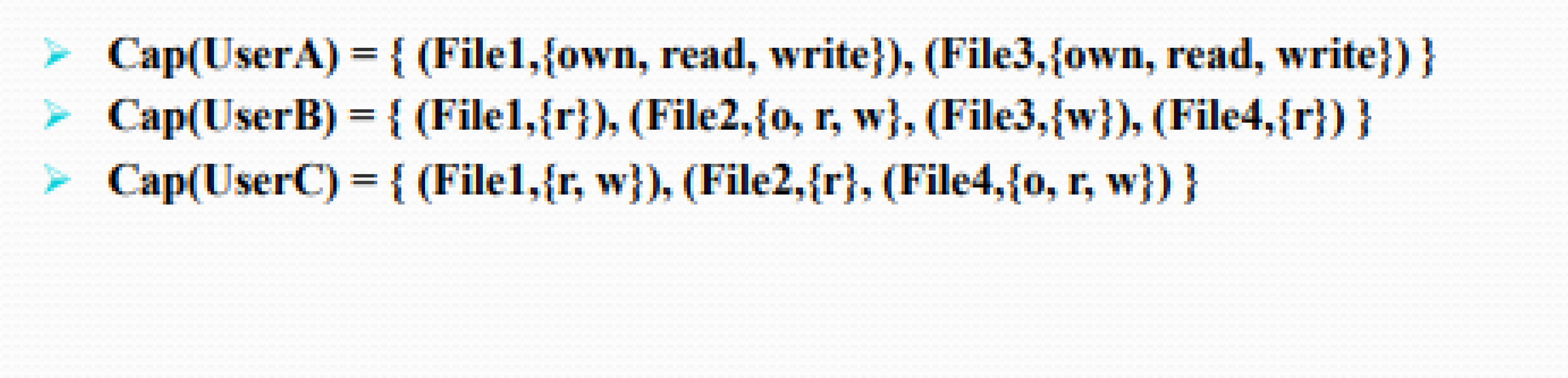
- ACL(File1) = { (UserA,{own, read, write}), (UserB,{read}), (UserC, {read, write}) }
- ACL(File2) = { (UserB,{o, r, w}), (UserC,{r}) }
- ACL(File3) = { (UserA,{o, r, w}), (UserB,{w}) }
- ACL(File4) = { (UserB,{r}), (UserC,{o, r, w}) }

Access Capability

- One list for each subject.

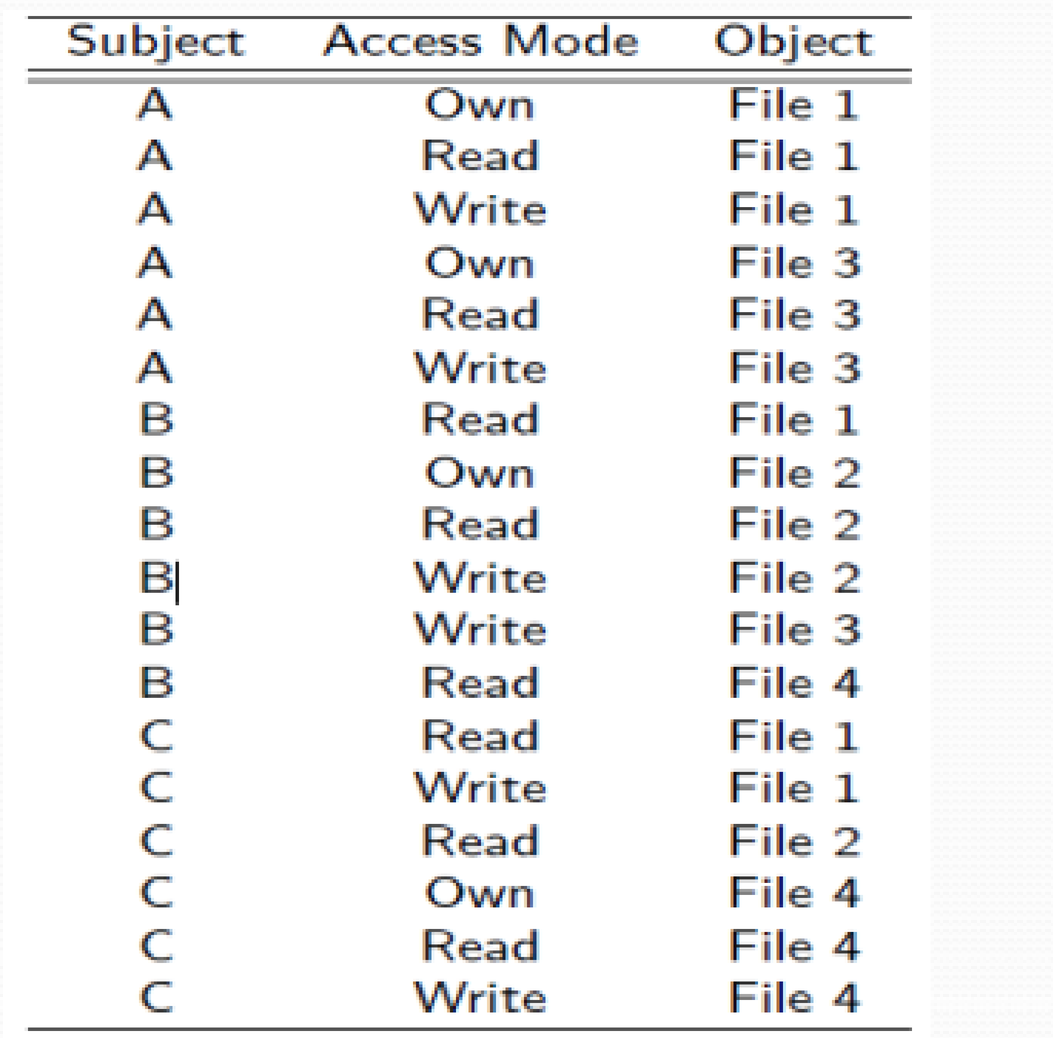


Access Capability



- Cap(UserA) = { (File1,{own, read, write}), (File3,{own, read, write}) }
- Cap(UserB) = { (File1,{r}), (File2,{o, r, w}, (File3,{w}), (File4,{r}) }
- Cap(UserC) = { (File1,{r, w}), (File2,{r}, (File4,{o, r, w}) }

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

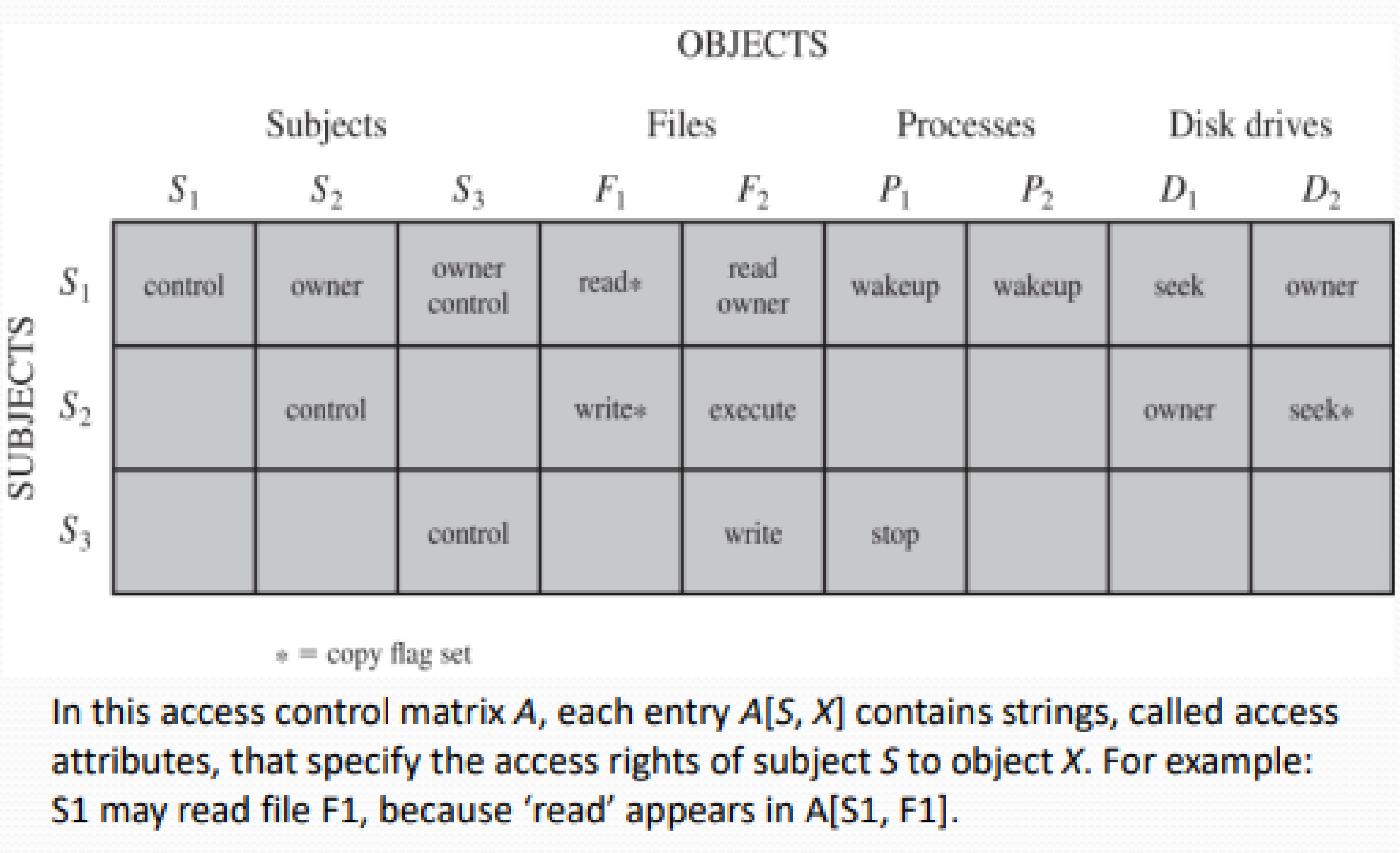
Graham-Denning Model

The Graham-Denning Model is a way to control who can do what with different parts of a computer system. Here's how it works:

1. Processes:
    1. Think of processes as the programs running on the computer.
    2. Access rights include things like being able to stop a process, start it up again, or delete it altogether.
2. Devices:
    1. Devices are things like your hard drive, printer, or mouse.
    2. Access rights let you read from or write to a device, control how it works (like moving a disk head), or decide who can use it.
3. Memory:
    1. Memory is where the computer stores information temporarily.
    2. Access rights control who can read from or write to different parts of the memory.

4. Subjects:
    1. Subjects are the users or programs that want to do things on the computer.
    2. Access rights determine what they're allowed to do, like read a file, write to a folder, or run a program.

Graham-Denning Model



OBJECTS

		Subjects			Files		Processes		Disk drives	
		S_1	S_2	S_3	F_1	F_2	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

In this access control matrix *A*, each entry *A*[*S*, *X*] contains strings, called access attributes, that specify the access rights of subject *S* to object *X*. For example: S1 may read file F1, because 'read' appears in A[S1, F1].

STUDENTS-HUB.com

Uploaded By: Mohammad ElRimawi

Graham-Denning Model

•Ownership:

•Everything in the system has an owner. For example, a file has an owner who controls it.

•Controller:

•Every user or program has someone or something that controls it. Sometimes, it controls itself.

•Transferable Rights:

•Some permissions can be passed from one user to another. It's like sharing a toy.

•Copy Flag:

•When you see a copy flag (*), it means a user can share that permission with others, either by

giving it away or keeping a copy for themselves

Graham-Denning Model

- A subject issues a request of type α for object X.

- The request causes the system (the operating system) to generate a message of the form (S0, α, X) to the controller for X.

- The controller examines the access matrix A to determine if α is in A[S0, X]. If so, the access is allowed; if not, the access is denied and a protection violation occurs. The violation should trigger a warning and appropriate action.