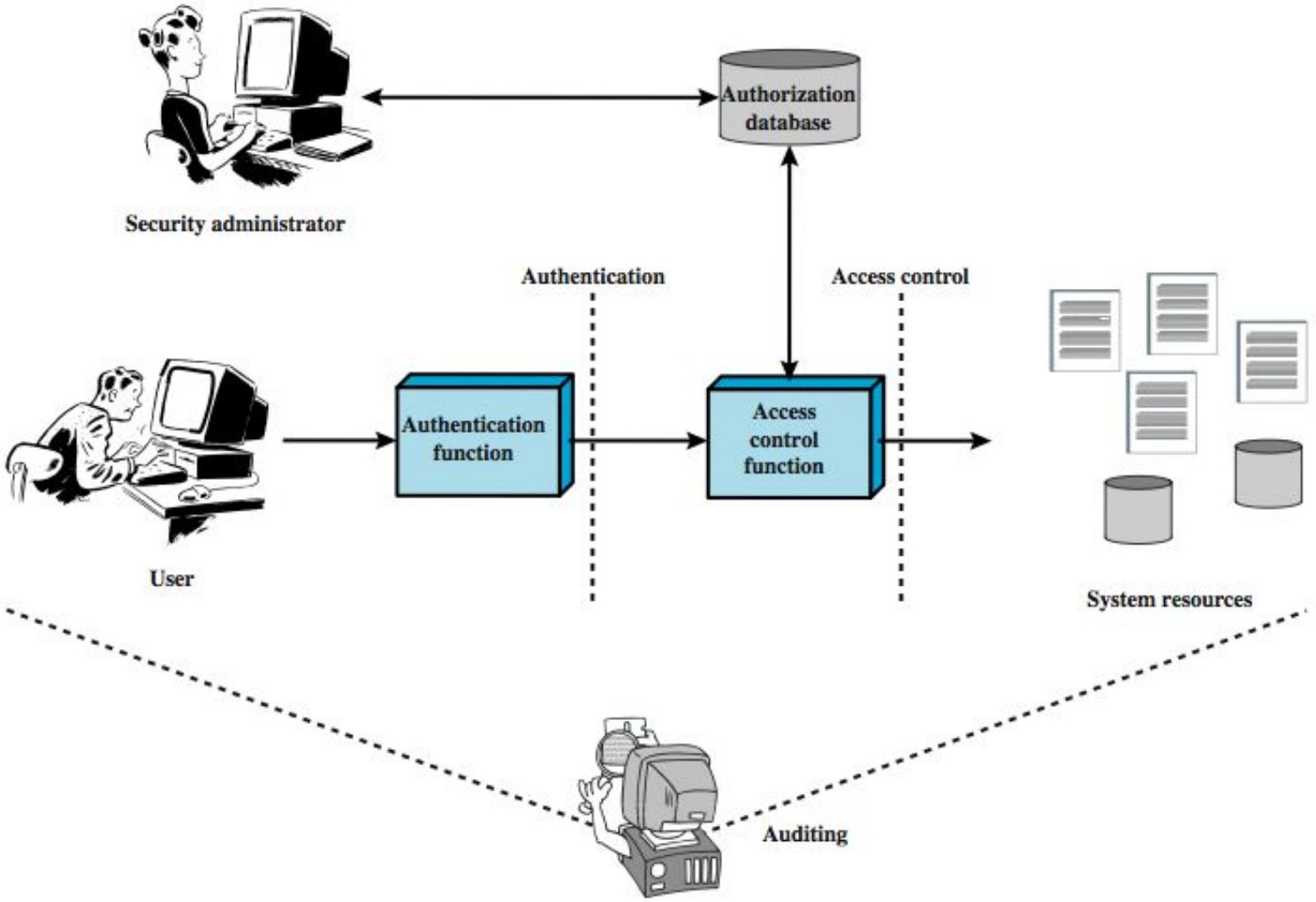
Computer Security: Principles and Practice

Chapter 4 – Access Control

Access Control

- “The prevention of unauthorized use of a resource, including the prevention of use of a resource in an unauthorized manner“
- central element of computer security
- assume have users and groups
  - authenticate to system
  - assigned access rights to certain resources on system

Access Control Principles



Access Control Policies

- Discretionary Access Control (DAC)
  - Based on identity / group of requestors
  - An entity can extends to other entities the permissions
- Mandatory Access Control (MAC)
  - Security labels on resources
  - Policy controlled by the administrator
- Role-Based Access Control (RBAC)
  - Based on the roles of the users

Access Control Elements

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

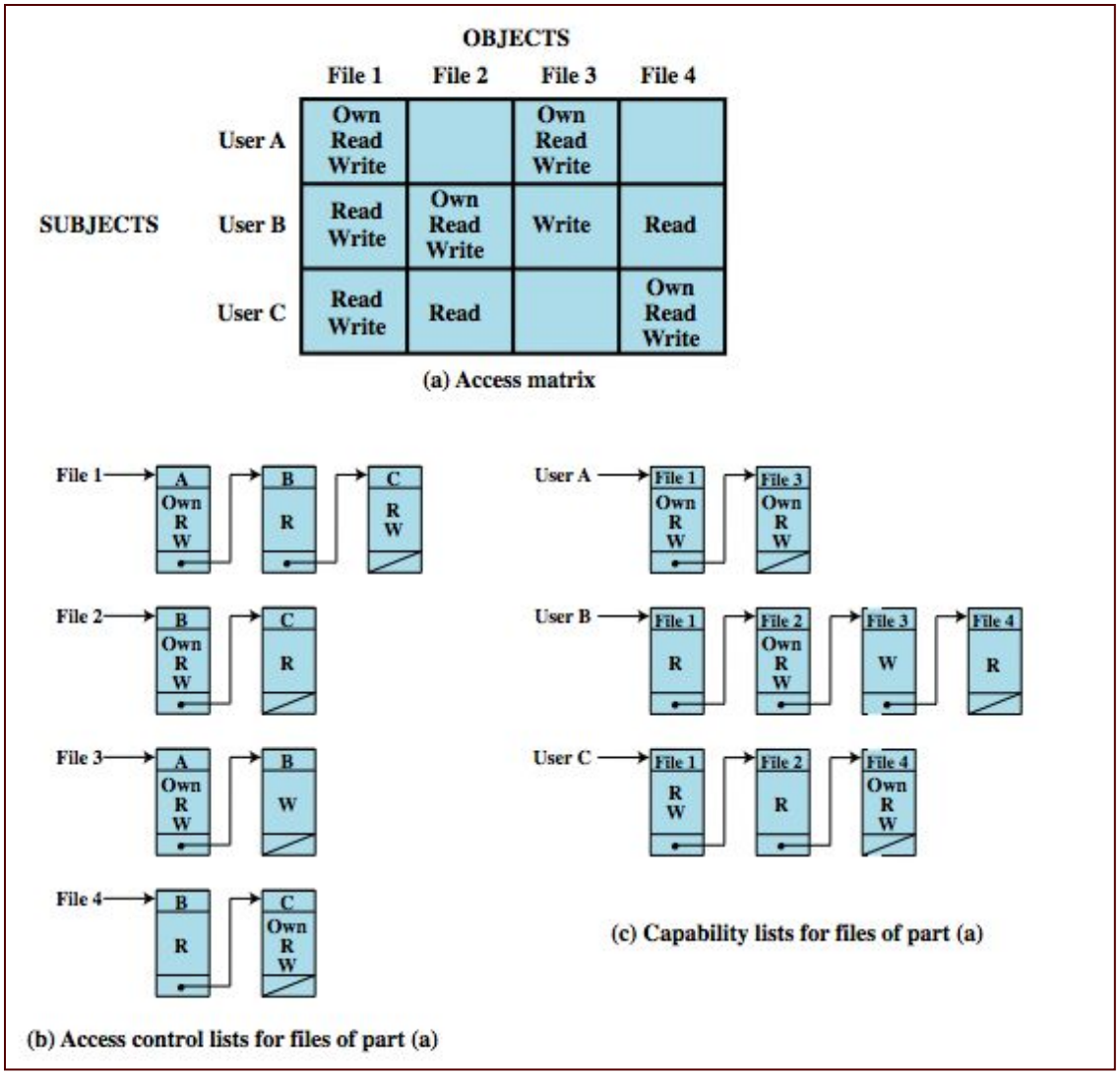
Discretionary Access Control

- often provided using a 2D access matrix
  - lists subjects in one dimension (rows)
  - lists objects in the other dimension (columns)
  - each entry specifies access rights of the specified subject to that object
- Do you see any problem with this representation?

Discretionary Access Control

- often provided using a 2D access matrix
  - lists subjects in one dimension (rows)
  - lists objects in the other dimension (columns)
  - each entry specifies access rights of the specified subject to that object
- Do you see any problem with this representation?
- access matrix is often sparse
- can decompose by either row or column

Access Control Structures



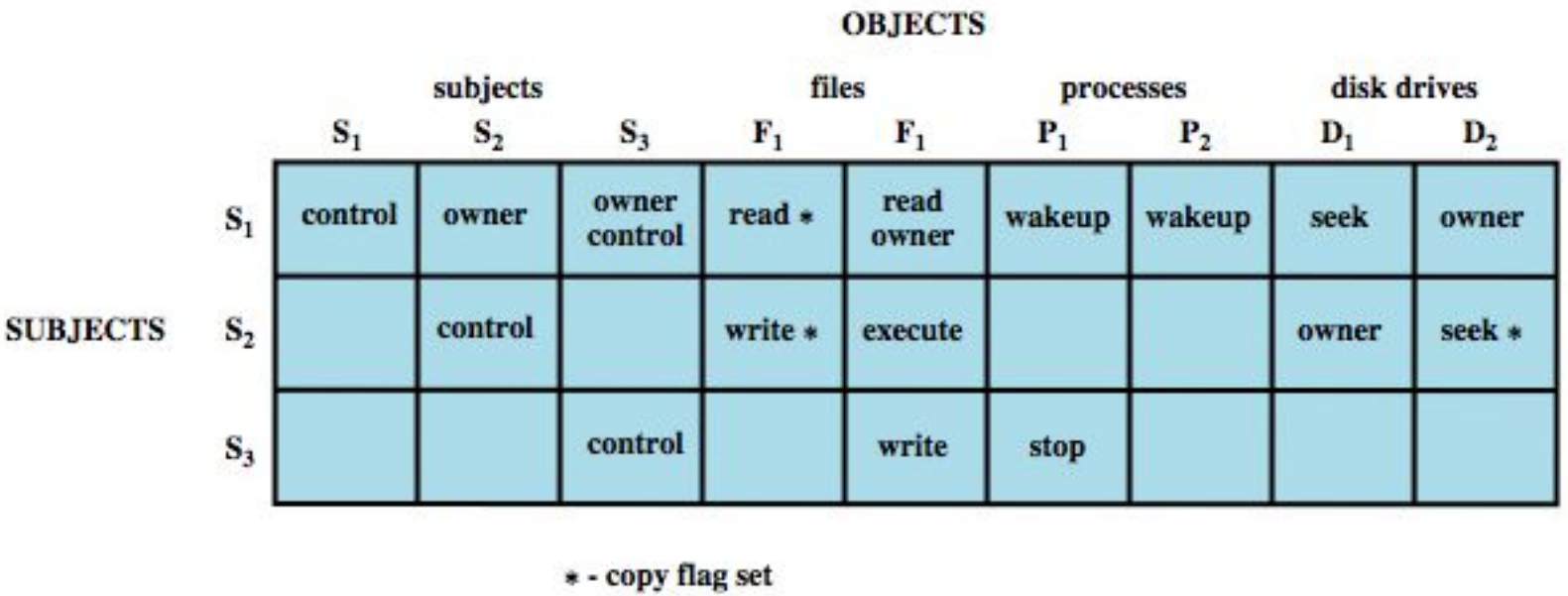
Access Control Structures

- Access Control List: list of users for each object
  - Pros
    - We can set default permissions
    - Easy to define groups permissions
  - Cons
    - Not optimize for determining access rights of users
- Capability tickets
  - Pros
    - Easy to determine set of rights of users
  - Cons
    - Tickets must be hold by OS

Access Control Model

- We want to generalize the DAC model
- Three main requirements
  - Good representation
  - Enforcing the access rights
  - Capability of users to alter specific rights
- Extension of the objects universe
  - Process
  - Device
  - Memory location
  - Subjects

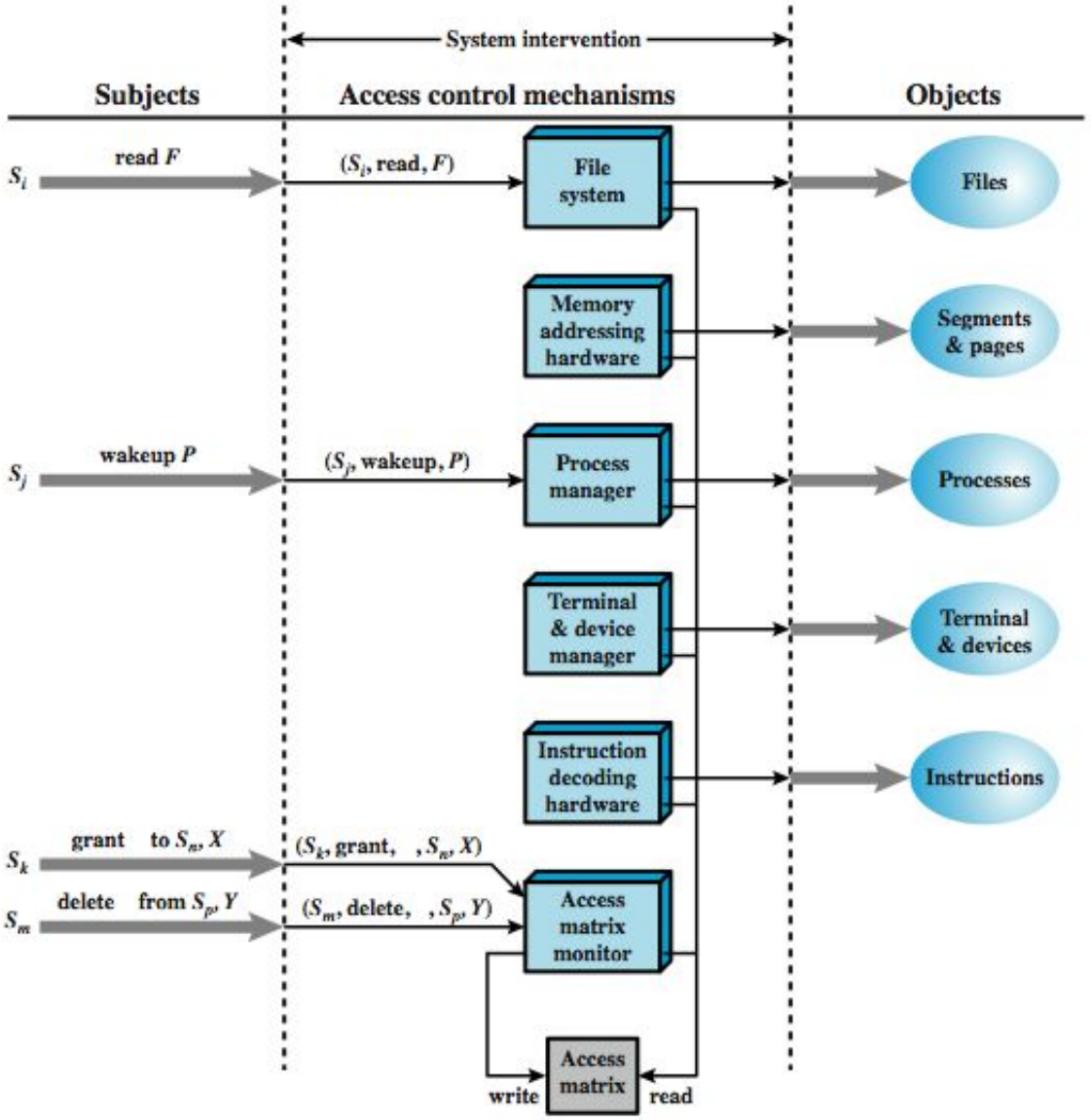
Access Control Model



		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

Access Control Function



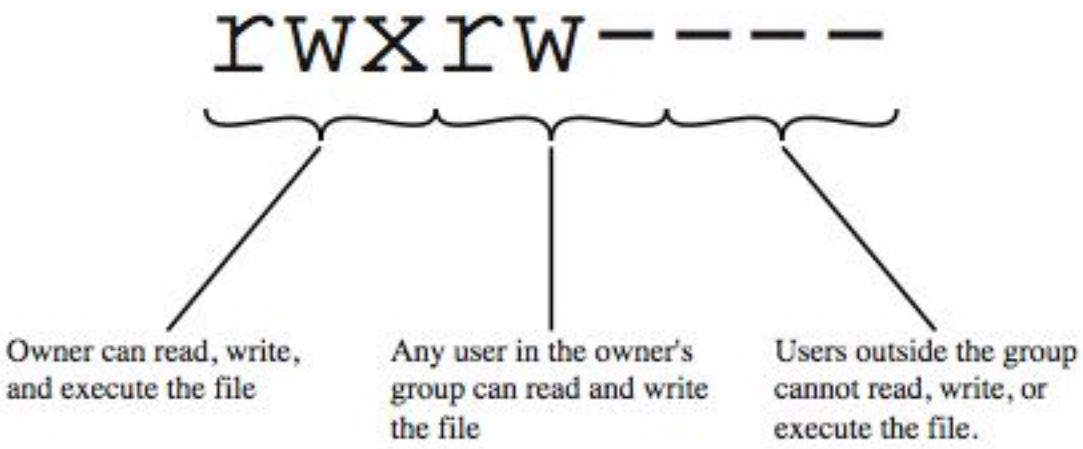
Protection Domains

- set of objects with associated access rights
- in access matrix view, each row defines a protection domain
  - but not necessarily just a user
  - may be a limited subset of user's rights
  - applied to a more restricted process
- may be static or dynamic

UNIX File Concepts

- UNIX files administered using inodes
  - It is a control structure with key info on file
    - attributes, permissions of a single file
    - One active inode per each file
  - have inode table / list for all files on a disk
    - copied to memory when disk mounted
- directories form a hierarchical tree
  - may contain files or other directories
  - Each of these with its own inode

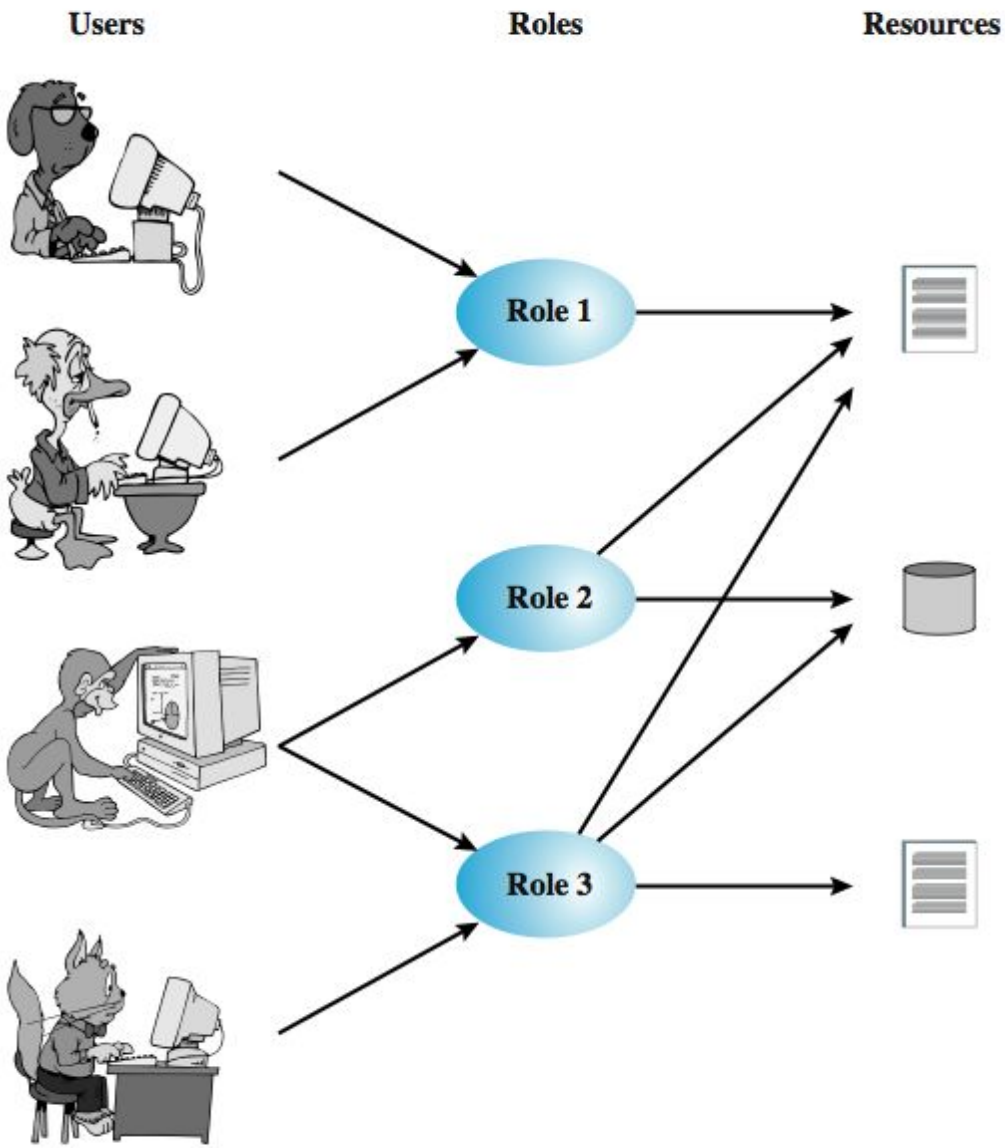
UNIX File Access Control



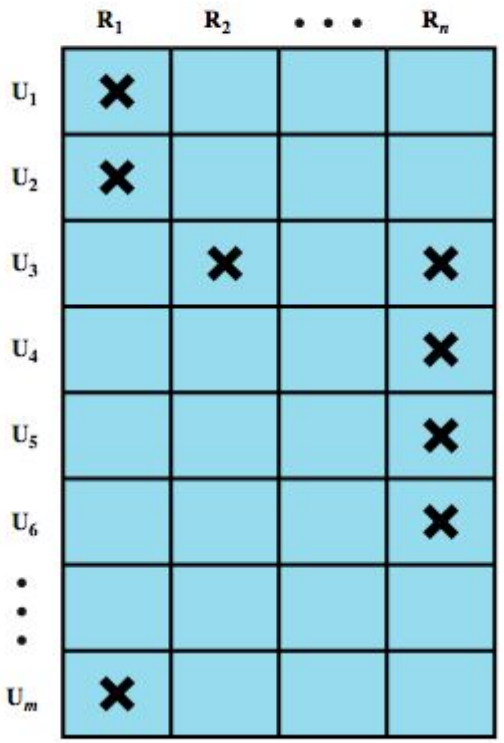
UNIX File Access Control

- “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

Role-Based Access Control



Role-Based Access Control



ROLES	OBJECTS								
	R1	R2	Rn	F1	F1	P1	P2	D1	D2
R1	control	owner	owner control	read *	read owner	wakeup	wakeup	seek	owner
R2		control		write *	execute			owner	seek *
• • •									
Rn			control		write	stop			

Summary

- introduced access control principles
  - subjects, objects, access rights
- discretionary access controls
  - access matrix, access control lists (ACLs), capability tickets
  - UNIX traditional and ACL mechanisms
- role-based access control