Chapter 14 - Protection

1. Definitions

> Protection is to ensure that only authorized processes can operate on certain resources limited to its rights.

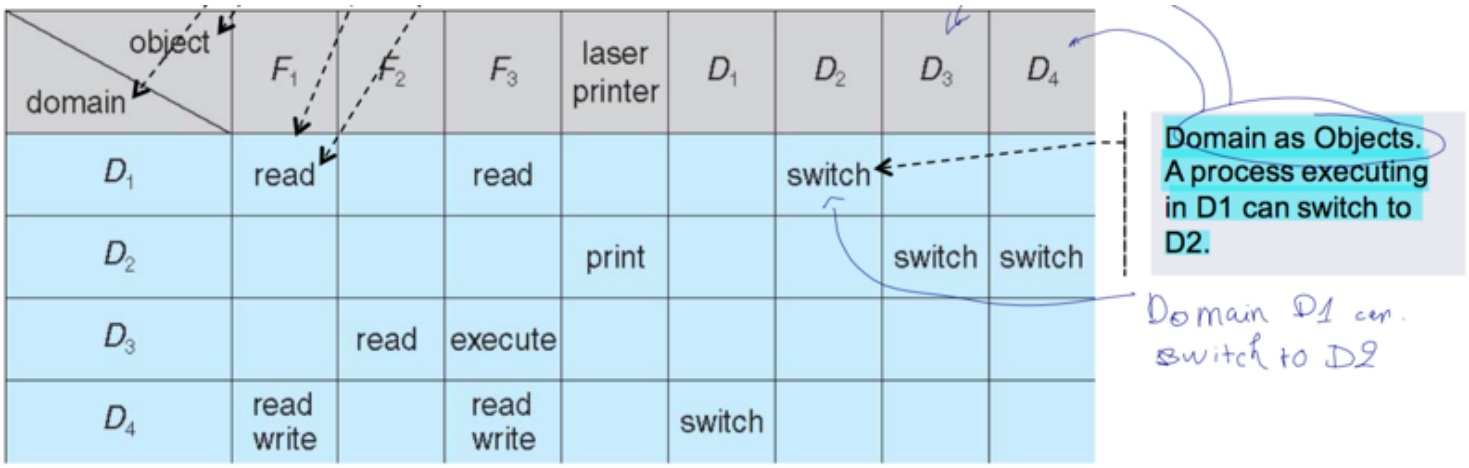
- Protection is internal. Security is external.
- **Key Principle** of Protection is the Principle of Least Privilege.

2. Domain

- Domain = Collection of Acess Rights = Objects, {Right-Set}
- Protection Domain = Process + Domain = Process + (Objects, {Right-Sets})
- Domain covers User, Process, Procedure and Domain Switching capability (User, Kernel mode).
- Domain can be static or dynamic.
- Unix Domain associates with User (instead of with process).

3. Access Matrix

Access matrix is a model of protection.

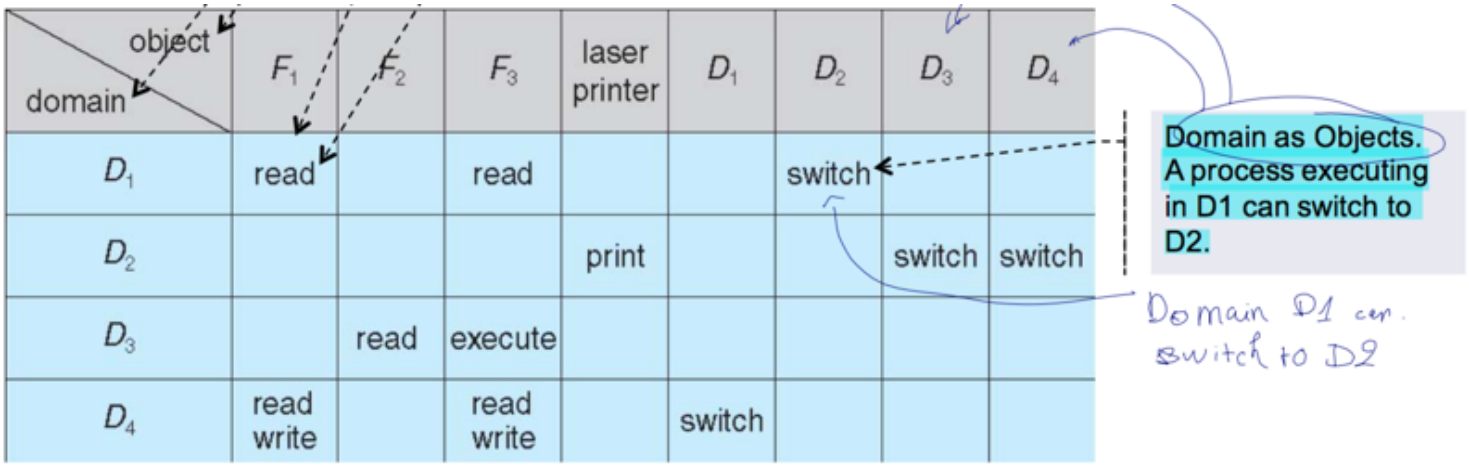


- Domain can have acess rights (switch, control) to other domain.
- Access Right copy is denoted by (*), indicating the access right can be cloned to other domain in the same object.
- Owner access right allow add/remove of rights.

3.1. Implementation of Access Matrix

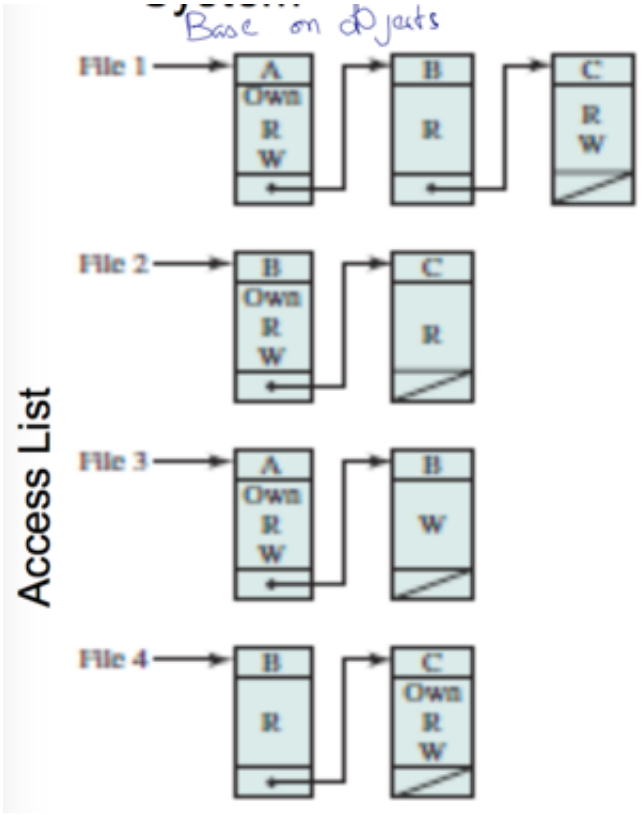
3.1.1. Global Table

- Simple, large table storing access rights between Domain, Object, Right-set.
- May need to use Virtual Memory for storing.



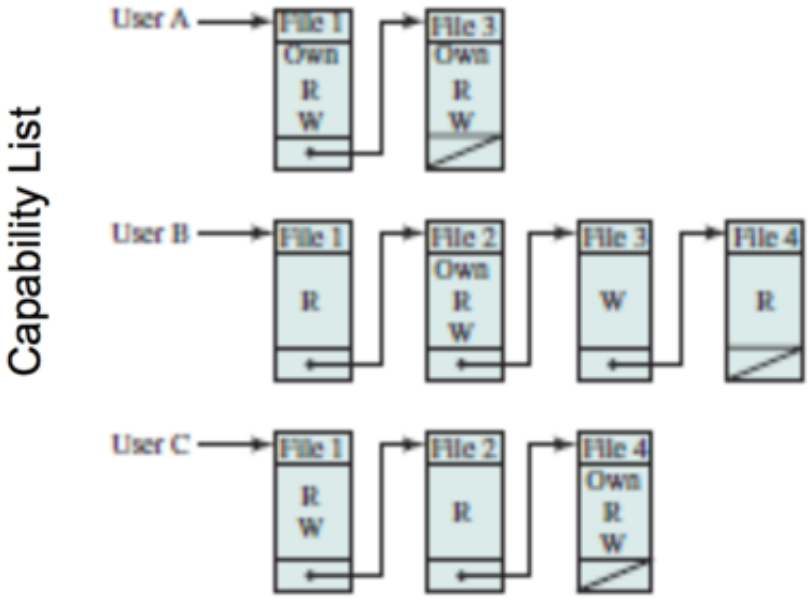
3.1.2. Access List

- Is an implementation of Access Matrix assoc with each Object.
- Base on Object, create link of Domain/User and allowed operations.
- Likely used in UNIX.



3.1.3. Capability List

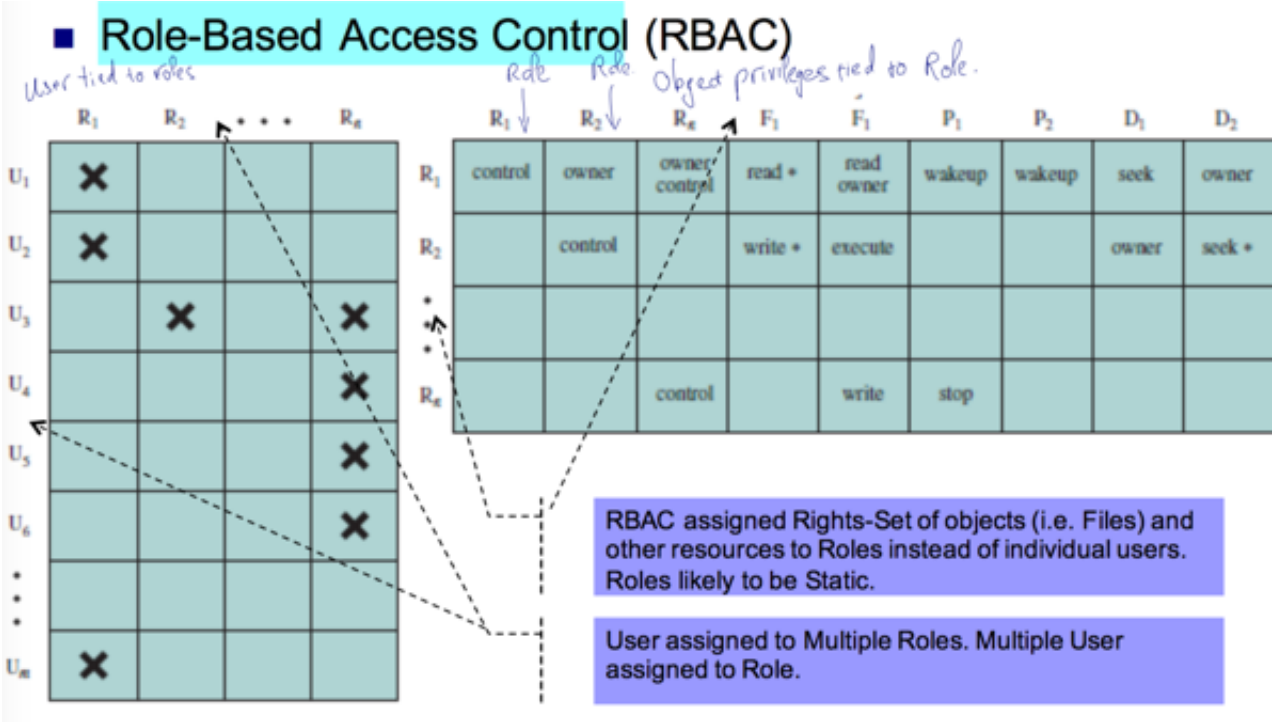
- Is an implementation of Access Matrix assoc with each Domain.
- Base on User/Domain, create link of Object and allowed operations.
- usually used for file descriptor table.



3.1.4. Role Based Access Control (RBAC)

- Is another form of Access Matrix used in Solaris, but adding concept of **Role**, and **Privileges** (Right Set).

- **Role is assigned to User/Process, and privileges and objects is assigned to Role**.
    - Users/Process —–> Roles —–> {Objects, Set of Rights}



4. Access Rights Revocation

- **Access List**: delete access rights from access list. Simple, immediate. **Revocation of Access-Rights is implemented using an Access List**.
- **Capability List**: requires to locate the capability in the system before revoking.

5. Capability-Based System

5.1. Hydra System

- Fixed set of access rights known and interpreted by the system.
- operations on Objects are through pre-defined procedures.

5.2. Cambridge CAP System

- Simpler but powerful
- Data Capability & software capability

compare Hydra & Cambridge.

6. Language-Based Protection

pros & cons of language based protection

- **More flexible** than the OS, but high overhead with access validation.
- Compiler-Based enforcement: restriction is built in to compiler.
- Example: **JVM** (Untrusted Classes, Stack Inspection)