Protection

Protection rings
- Limit the power that you have

User vs. kernel
- More damage you can do
- Harder to verify

8 rings, 0 highest power
- ring j has powers [j,7]

Each segment is associated with 1 ring
- Segment descriptor contains
  - which ring as well as r,w,x bits
  - access bracket: b1 and b2 s.t. b1 <= b2
  - limit: b3 > b2
  - list of gates: entry points into segment

If process in ring j calls segment with (b1,b2)
1. Access is allowed if b1 <= j <= b2
2. Current ring number stays the same

Else trap to the OS
1. if j < b1, call is allowed because call is from higher privilege segment to lower privilege segment
   a. Parameters passed to segment must be copied into an area accessible by called segment
2. if j > b2 call is allowed if j <= b3 and call is directed to a gate
   a. this allows segments with lower privileges to access segments with higher privileges

Call to lower privilege (1→ 2)
- Calling procedure might specify parameters that new domain does not have access to
- Software intervention needed to handle this
- Gate must be specified for downward return. Return gate must be created at call and deleted after return

Call to higher privilege (2 → 1)
- Called procedure must find new stack area
- Called procedure must validate called had rights to parameters

Benefits of protection rings
- Allows a layered supervisor to be included in VM of each process
- User constructed subsystems may exist in middle rings (2-3) eg compiler
- User can protect himself while debugging his own (or borrowed) program

UNIX
- Domain associated with every user
- Change user id temporarily
- Accomplished through the file system
  - Each file has owner id and domain bit (setuid bit)
  - When exec a file, if setuid is on, user_id is set to owner of file (which may or may not be same as user exec file)

Capabilities (HYDRA)
- Each domain has list of object and operation pairs
- Object (with its operations) is called a capability
- Possession of a capability means you have that access
- Os is the only entity that monkeys with capabilities (assigns, changes, revokes)
- Capabilities never allowed to migrate into address space directly accessible by user processes

HYDRA
- Resources (physical and abstract) play a central role in OS
- Therefore, mechanisms provided by Hydra intended to support abstracted notion of a resource (object)
- Objects can be constructed from other objects

Key aspects of Hydra
- Generalized notion of resource
- Definition of execution domain
- Provides protetion mechanism for application of operations (procedures) to instances of resources (objects)

Object
- Unique name
- Type part
- Representation {
  - Data
  - Capability – references to other objects
- }
- …Capabilities allow construction of new types from existing types

Execution domains change precisely when a procedure is entered or exited
Capabilities (specify new domain) and procedure calls (domain change) are the basis for protection
Process is stack of LNS's which represents cumulative state of single sequential task
LNS is created on procedure call
- Template for procedure
- Actual parameters

Combine both of these and rights can be more than you had previously (amplification)

Ex: Exists a system call that can write to disk, but not to any arbitrary directory. Your program has rights to write to a specific directory, but can't write to disk directly. It can only call the write system call. When you pass your right to write the directory to the system call to write to disk, the amplification allows a write to the specific directory. Upon return, the amplification is revoked and rights are returned to their previous state.

ACL vs Capabilities

ACL
- Each object has an access list of <domain, right_set>, which define all domains with non-empty set of access rights for object.
- Correspond to needs of users
- Determining set of access rights for each domain is difficult
- Every access to object must be checked
- Revocation of rights is easy
- Hard to revuse access to single domain but give to all others

Capabilities
- Useful for localizing particular process (domain) info
- Fast
- Revocation of rights is really hard
- Hard to hand out capabilities to everyone