Protection and the control of information sharing in Multics

Mainpoints: permission rather than exclusions, real-time authentications, least privilege, human interfaces.

## mechanism

- access control lists

- hierarchical control of access specifications

- identification and authentication of users

- primary memory protection

## design principles

- base the protection on permission rather than exclusion

- check access for \*\*current\*\* authority

- the design is not a secret

- principle of least privilege

- easy-to-use user interface

- decentralization protection

- protect subsystem

## the storage system and access control lists

The basic unit of the storage system: segments(x86 use that), each of segment is associated with an access control list

\tip - segment: logical unit which contains different number of pages. - page: the space unit.

Implement segment sharing by pointing to the same page table.

segment <- descriptor <- Access <- process <- principal identifier

Principal identifiers are separated into groups: 1) herself; 2) project groups; 3) compartment

\tip - t**hey use MMU(hardware) to check the access right** to ensure every access has a check.

Using a hardware to check means that we cannot replace the hardware, and Multics is not transferable among machines.

acl

* easy to set up
* slow to check(has to search through the access right list)

capabilities

* easy to check and transfer
* hard to revoke

Object types: segments; message queues; directories; media descriptors;

^

|

Different control modes;

The drawbacks of capabilities:

* link permanent
* Different rights of one user
* To get the list of people who have rights, one has to do a lot searching;

Backup copying && bulk I/O —— all done by operator-controlled process just as ordinary users.

## hierarchical control of access specifications

All objects are arranged into a single hierarchical tree of directories.

This hierarchical control -> some super-power administer may appear

Another kind of control: self-control, but the entirely private working zone may be abuse.

## authentication of users

authentication - identify the principal identifier

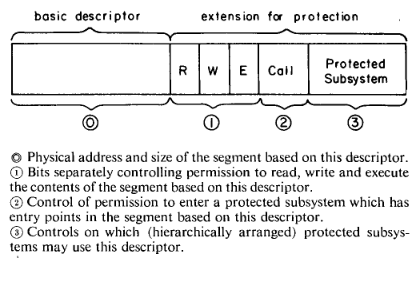
a user: a unique name && password

Other features:

1. authentication interactively
2. password change ( system provides)
3. password of garbling characters
4. time-out
5. logging penetration
6. anonymous suer

Everyone only knows his/her OWN password.

## primary memory protection



three extensions for the descriptors

supervisor(没有看懂) - kernel

subsystem: ring of protection

subsystem: data and procedures which can only be accessed through certain entries (gates).

\tip - like, ring0: kernel; ring….; ring4: user apps; those rings with lower levels can get access to those higher levels.

## weaknesses of the multics protection mechanisms

* programs can hurt the protection systems. Those programs are in the most protected areas. The internal codes in the system are required to be reduced.
* complexity of the user interface
* weak communications
* weak operator interface
* easy-to-guess passwords
* secondary storage residues uncleared
* overpriviledged administrator
* ponderous backup
* anti-ai
* potential vulnerability

Multics: basic, extendable, securable,

notes