CS323 Operating Systems Security

Yuanyuan Zhou

Lecture 39

5/5/2003

Reminder

- Conflict exam signup: by noon
- Graduate students 1-unit project
- Last lecture
  - Very condensed brief review: main taken-aways from this class
  - Graduate student 1-unit project demo/presentation
    - Survey/code is due the same day

Content

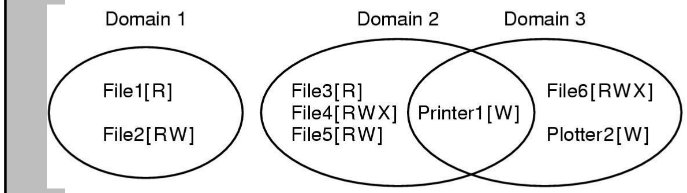
- Goals of Protection
- Mechanisms and Policies
- Protection Domain
- Access Matrix
- Implementation of the Access Matrix
- Access Lists
- Capability Lists
- Mixed Approaches: Locks and Keys
- Summary

Protection Domain

- A computer system is a set of processes and objects
- Processes and objects have unique names
- Objects are abstract data types with well-defined operations
- A process operates within a protection domain
- A protection domain specifies the resources a process may access and the types of operations that may be invoked on the objects.
- **The Principle of Least Privilege *Need to know*: The protection domain of a process should be as small as possible consistent with the need of that process to accomplish its assigned task.**

Protection Mechanisms

Protection Domains



Examples of three protection domains

Access Matrix

	Object							
Domain	File1	File2	File3	File4	File5	File6	Printer1	Plotter2
1	Read	Read Write						
2			Read	Read Write Execute	Read Write		Write	
3						Read Write Execute	Write	Write

Access Matrix with Domains as Objects

	Object										
main	File1	File2	File3	File4	File5	File6	Printer1	Plotter2	Domain1	Domain2	Domain3
1	Read	Read Write								Enter	
2			Read	Read Write Execute	Read Write		Write				
3						Read Write Execute	Write	Write			

Implementation of the Access Matrix

- Global Table - assume <D, O, R>
  - On invocation of a method R on an object O by a process P running in a domain D , the table Domain column is searched for D,
  - the Object row is searched for an entry O,
  - the entry at the intersection* of the row and column is searched for the method R.
- Table may be Sparse
- Table may be too large to store in main memory (use virtual memory - overhead)
- Objects that may be accessed from every Domain need to be entered in every row
- Needs a searching operation
- In parallel or distributed system, access to table may be bottleneck

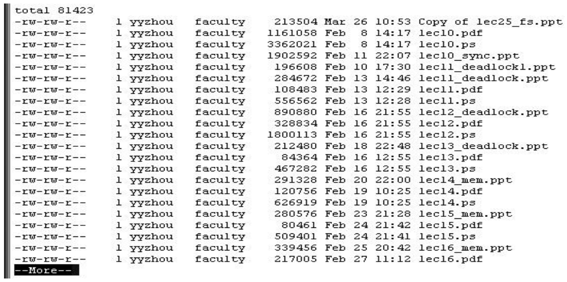
Copy Rights

- The access matrix is an object that can be changed
- The **copy** right allows an access right to be copied into the same column of other rows in the matrix.
- Variants:
  - **copy**: the right and copy right is copied
  - **transfer**: when a right is copied from one Domain to another, the old Domain loses the right.
  - **limited copy**: a right can be copied, but not the right to copy.
  - **copy right**: the right to copy a copy right is a separate right

Access Lists

- Each column in the access matrix is implemented as an access list for one Object.
  - On invocation of a method R on an object O by a process P running in a domain D,
  - the access is dereferenced to the Object O
  - the access list is searched for D,
  - the methods are searched for an entry R.
- Empty entries in Access Matrix can be discarded.
- Storage for access lists is proportional to the number of Objects
- A default can be associated with an access list so that any Domain not specified in the list can access the Objects using default methods.
- It is easy for the owner of the Object to grant access to another Domain or revoke access.
- It is easy to determine which processes can access an object.
- However, all processes can find out that the Object exists.
- ACL entries can be for individual users or for a group of users.

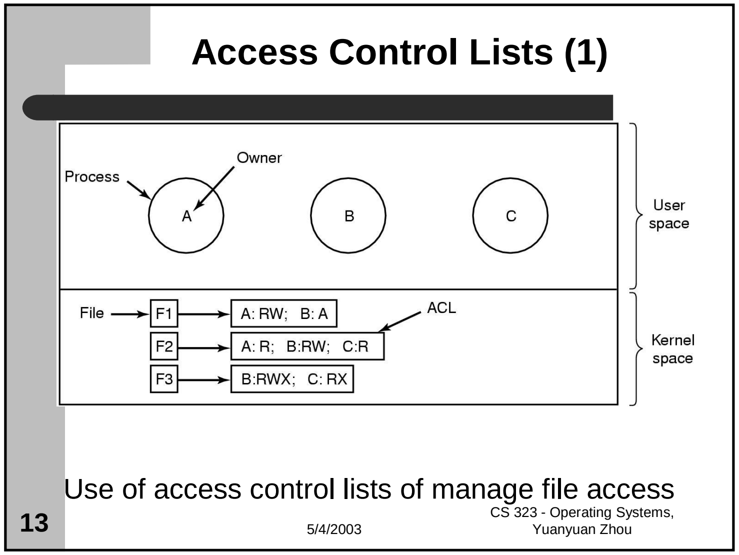
Access List Example



```
total 81423
-rw-rw-r--    1 yyzhou   faculty    213504 Mar 26 10:53 Copy of lec25_fs.ppt
-rw-rw-r--    1 yyzhou   faculty   1161058 Feb  8 14:17 lec10.pdf
-rw-rw-r--    1 yyzhou   faculty   3362021 Feb  8 14:17 lec10.ps
-rw-rw-r--    1 yyzhou   faculty   1902592 Feb 11 22:07 lec10_sync.ppt
-rw-rw-r--    1 yyzhou   faculty    196608 Feb 10 17:30 lec11_deadlock1.ppt
-rw-rw-r--    1 yyzhou   faculty    284672 Feb 13 14:46 lec11_deadlock.ppt
-rw-rw-r--    1 yyzhou   faculty    108483 Feb 13 12:29 lec11.pdf
-rw-rw-r--    1 yyzhou   faculty    556562 Feb 13 12:28 lec11.ps
-rw-rw-r--    1 yyzhou   faculty    890880 Feb 16 21:55 lec12_deadlock.ppt
-rw-rw-r--    1 yyzhou   faculty    328834 Feb 16 21:55 lec12.pdf
-rw-rw-r--    1 yyzhou   faculty   1800113 Feb 16 21:55 lec12.ps
-rw-rw-r--    1 yyzhou   faculty    212480 Feb 18 22:48 lec13_deadlock.ppt
-rw-rw-r--    1 yyzhou   faculty     84364 Feb 16 12:55 lec13.pdf
-rw-rw-r--    1 yyzhou   faculty    467282 Feb 16 12:55 lec13.ps
-rw-rw-r--    1 yyzhou   faculty    291328 Feb 20 22:00 lec14_mem.ppt
-rw-rw-r--    1 yyzhou   faculty    120756 Feb 19 10:25 lec14.pdf
-rw-rw-r--    1 yyzhou   faculty    626919 Feb 19 10:25 lec14.ps
-rw-rw-r--    1 yyzhou   faculty    280576 Feb 23 21:28 lec15_mem.ppt
-rw-rw-r--    1 yyzhou   faculty     80461 Feb 24 21:42 lec15.pdf
-rw-rw-r--    1 yyzhou   faculty    509401 Feb 24 21:41 lec15.ps
-rw-rw-r--    1 yyzhou   faculty    339456 Feb 25 20:42 lec16_mem.ppt
-rw-rw-r--    1 yyzhou   faculty    217005 Feb 27 11:12 lec16.pdf
--More--
```

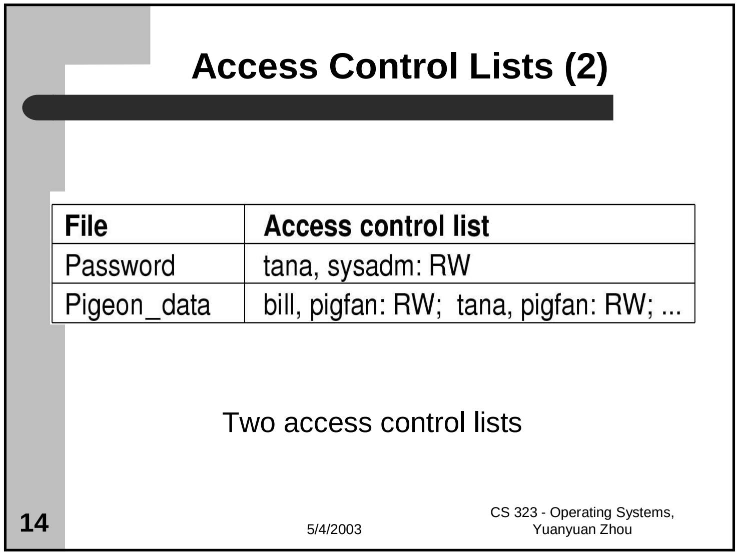
Implementations of Access Lists

- File Systems
  - Opening a file is checked against an access list to determine if a process may open the file with a given set of access methods.
- Login Shells
  - The login to a system is checked against an access list (usually the password file owned by root).
  - Rlogins are checked against an .rhost file that contains the names of machines from which a rlogin is permitted.



Access Control Lists (1)

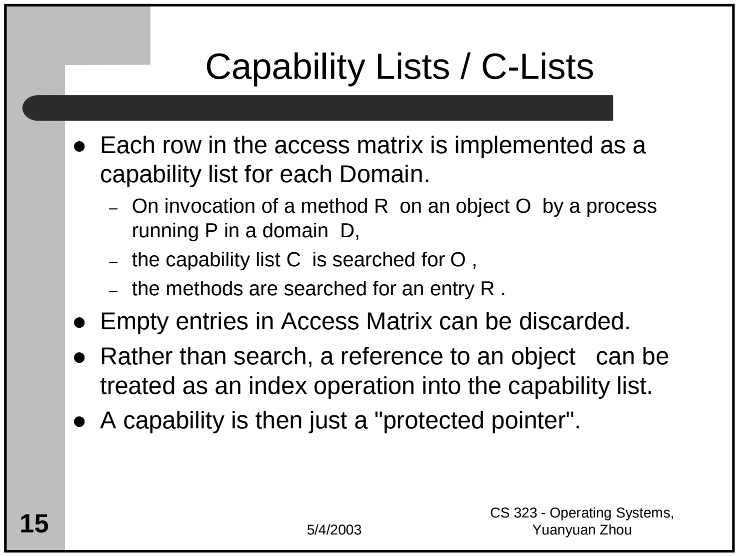
Use of access control lists of manage file access



Access Control Lists (2)

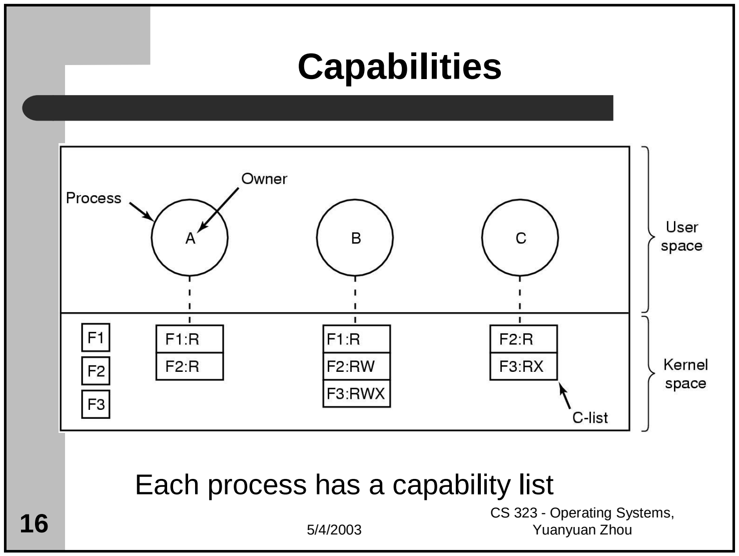
File	Access control list
Password	tana, sysadm: RW
Pigeon_data	bill, pigfan: RW; tana, pigfan: RW; ...

Two access control lists



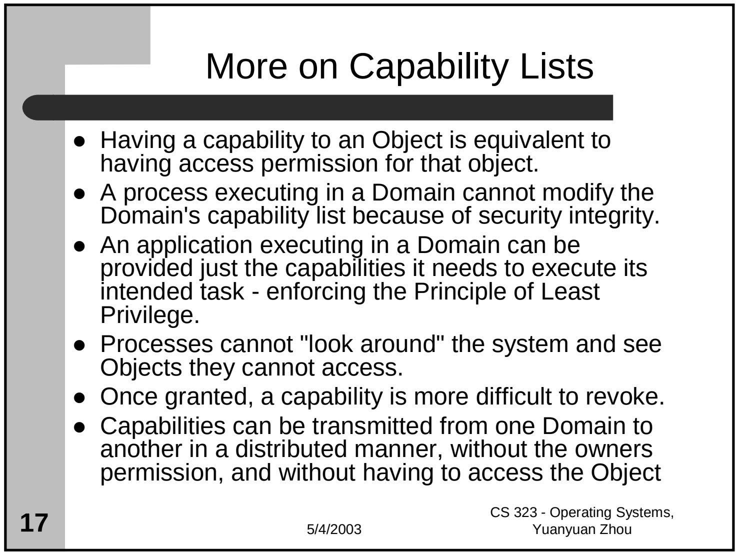
Capability Lists / C-Lists

- Each row in the access matrix is implemented as a capability list for each Domain.
  - On invocation of a method R on an object O by a process running P in a domain D,
  - the capability list C is searched for O ,
  - the methods are searched for an entry R .
- Empty entries in Access Matrix can be discarded.
- Rather than search, a reference to an object can be treated as an index operation into the capability list.
- A capability is then just a "protected pointer".



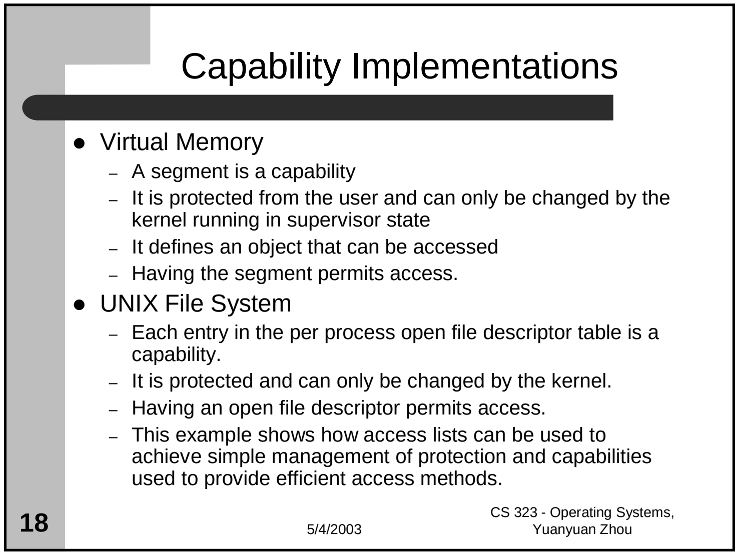
Capabilities

Each process has a capability list



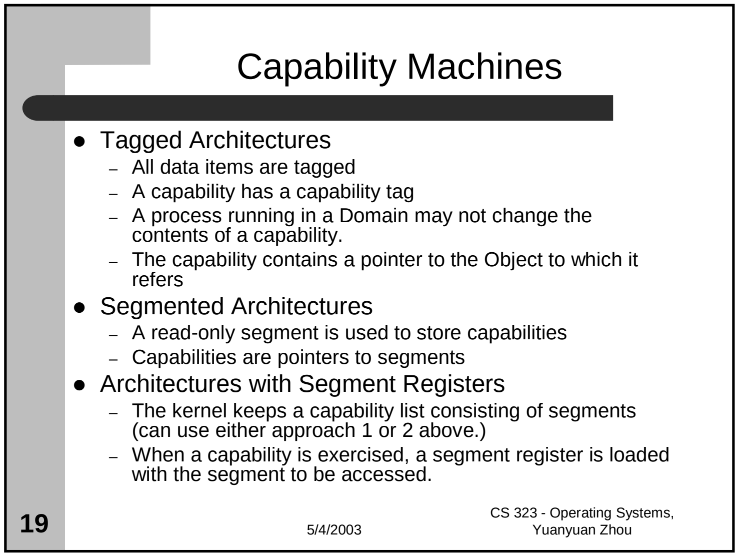
More on Capability Lists

- Having a capability to an Object is equivalent to having access permission for that object.
- A process executing in a Domain cannot modify the Domain's capability list because of security integrity.
- An application executing in a Domain can be provided just the capabilities it needs to execute its intended task - enforcing the Principle of Least Privilege.
- Processes cannot "look around" the system and see Objects they cannot access.
- Once granted, a capability is more difficult to revoke.
- Capabilities can be transmitted from one Domain to another in a distributed manner, without the owners permission, and without having to access the Object



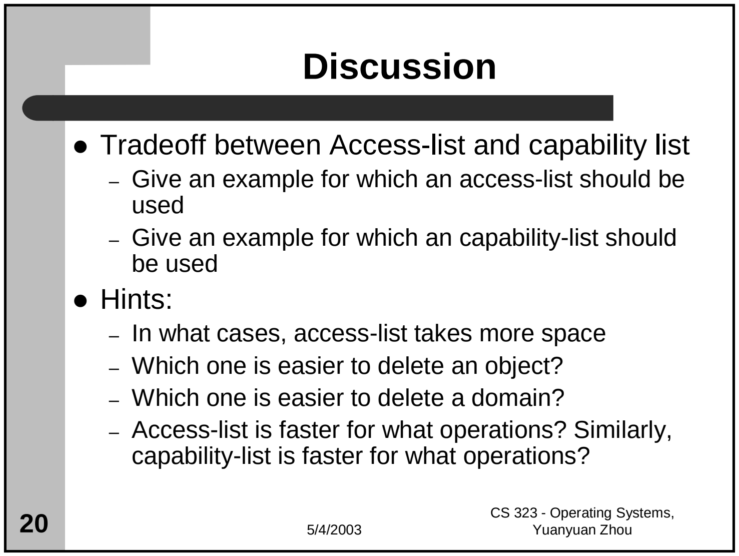
Capability Implementations

- Virtual Memory
  - A segment is a capability
  - It is protected from the user and can only be changed by the kernel running in supervisor state
  - It defines an object that can be accessed
  - Having the segment permits access.
- UNIX File System
  - Each entry in the per process open file descriptor table is a capability.
  - It is protected and can only be changed by the kernel.
  - Having an open file descriptor permits access.
  - This example shows how access lists can be used to achieve simple management of protection and capabilities used to provide efficient access methods.



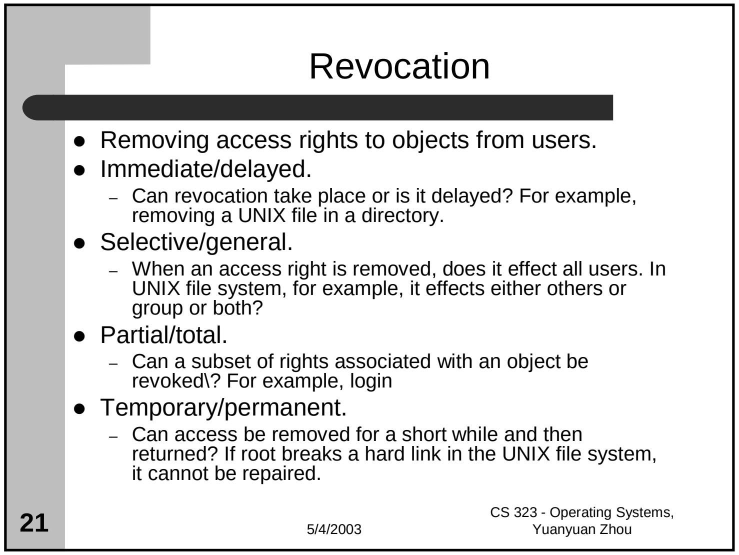
Capability Machines

- Tagged Architectures
  - All data items are tagged
  - A capability has a capability tag
  - A process running in a Domain may not change the contents of a capability.
  - The capability contains a pointer to the Object to which it refers
- Segmented Architectures
  - A read-only segment is used to store capabilities
  - Capabilities are pointers to segments
- Architectures with Segment Registers
  - The kernel keeps a capability list consisting of segments (can use either approach 1 or 2 above.)
  - When a capability is exercised, a segment register is loaded with the segment to be accessed.



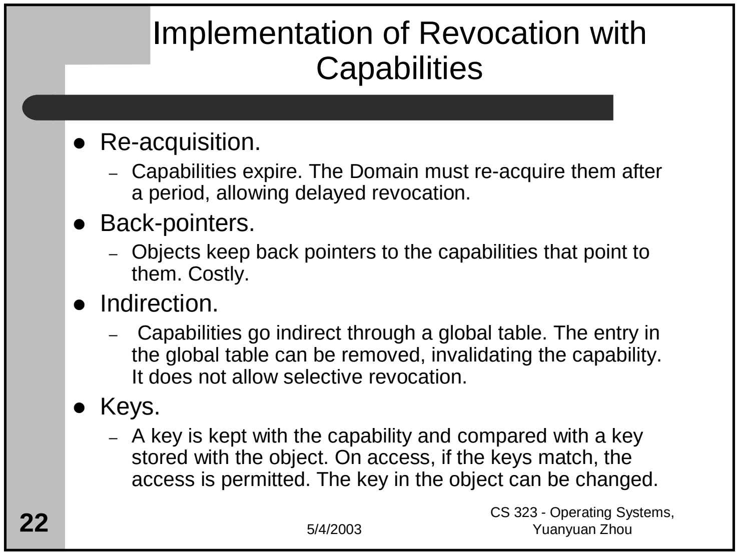
Discussion

- Tradeoff between Access-list and capability list
  - Give an example for which an access-list should be used
  - Give an example for which an capability-list should be used
- Hints:
  - In what cases, access-list takes more space
  - Which one is easier to delete an object?
  - Which one is easier to delete a domain?
  - Access-list is faster for what operations? Similarly, capability-list is faster for what operations?



Revocation

- Removing access rights to objects from users.
- Immediate/delayed.
  - Can revocation take place or is it delayed? For example, removing a UNIX file in a directory.
- Selective/general.
  - When an access right is removed, does it effect all users. In UNIX file system, for example, it effects either others or group or both?
- Partial/total.
  - Can a subset of rights associated with an object be revoked\? For example, login
- Temporary/permanent.
  - Can access be removed for a short while and then returned? If root breaks a hard link in the UNIX file system, it cannot be repaired.



Implementation of Revocation with Capabilities

- Re-acquisition.
  - Capabilities expire. The Domain must re-acquire them after a period, allowing delayed revocation.
- Back-pointers.
  - Objects keep back pointers to the capabilities that point to them. Costly.
- Indirection.
  - Capabilities go indirect through a global table. The entry in the global table can be removed, invalidating the capability. It does not allow selective revocation.
- Keys.
  - A key is kept with the capability and compared with a key stored with the object. On access, if the keys match, the access is permitted. The key in the object can be changed.