OPERATING SYSTEMS

Input - Output Management and Security - 4

Nitin V Pujari
Faculty, Computer Science
Dean - IQAC, PES University

OPERATING SYSTEMS

Access Matrix

Nitin V Pujari
Faculty, Computer Science
Dean - IQAC, PES University

OPERATING SYSTEMS

Course Syllabus - Unit 5

10 Hours

Unit-5:Unit 5: IO Management and Security

I/O Hardware, polling and interrupts, DMA, Kernel I/O Subsystem and Transforming I/O Requests to Hardware Operations - Device interaction, device driver, buffering System Protection: Goals, Principles and Domain of Protection, Access Matrix, Access control, Access rights. System Security: The Security Problem,Program Threats, System Threats and Network Threats. Case Study: Windows 7/Windows 10

OPERATING SYSTEMS

Course Outline

47	I/O Hardware, polling and interrupts	13.1,13.2
48	DMA	13.2.3
49	Transforming I/O Requests to Hardware Operations, Device interaction, device driver, buffering.	13.5
50	Goals, Principles and Domain of Protection	14.1-14.3
51	Access Matrix	14.4
52	Access control, Access rights	14.5-14.7
53	The Security Problem	15.1
54	Program Threats	15.2
55	System Threats and Network Threats	15.3
56	Case Study : Windows File System	17.5

- # Access Matrix

Access Matrix

- Each **User** may be a domain.
- Each **Process** may be a domain.
- Each **Procedure** may be a domain.
- A **Protection Domain** specifies the resources that a process may access.

Access Matrix

- Our general model of protection can be viewed abstractly as a matrix, called an **Access Matrix.**

- The rows of the access matrix represent **domains**, and the columns represent **objects**.

- Each entry in the matrix consists of a set of **access rights**.

- The entry access (i,j) defines the set of operations that a process executing in domain Di can invoke on object Oj

Access Matrix

- Figure Illustrates the concept of **access matrix**

- There are four domains and four objects—three files (F1 , F2 , F3) and one laser printer.

- A process executing in domain D1 can read files F1 and F3

- A process executing in domain D4 has the same privileges as one executing in domain D1 ; but in addition, it can also write onto files F1 and F3.

- The laser printer can be accessed only by a process executing in domain D2

object / domain	F_1	F_2	F_3	printer
D_1	read		read	
D_2				print
D_3		read	execute	
D_4	read write		read write	

Access Matrix

- The access-matrix scheme provides us with the mechanism for specifying a variety of policies.
- The mechanism consists of implementing the access matrix and ensuring that the semantic properties outlined hold.
- More specifically, one must ensure that a process executing in domain Di can access only those objects specified in row i, and then only as allowed by the access-matrix entries.

domain \ object	F_1	F_2	F_3	printer
D_1	read		read	
D_2				print
D_3		read	execute	
D_4	read write		read write	

Access Matrix

- The access matrix can implement policy decisions concerning protection.
- The policy decisions involve which rights should be included in the (i, j) th entry.
- The domain in which each process executes is typically decided by the Operating System

domain \ object	F_1	F_2	F_3	printer
D_1	read		read	
D_2				print
D_3		read	execute	
D_4	read write		read write	

Access Matrix

- The users normally decide the contents of the access-matrix entries.
- When a user creates a new object Oj , the column Oj is added to the access matrix with the appropriate initialization entries, as dictated by the creator.
- The user may decide to enter some rights in some entries in column j and other rights in other entries, as needed

domain \ object	F_1	F_2	F_3	printer
D_1	read		read	
D_2				print
D_3		read	execute	
D_4	read write		read write	

Access Matrix

- The access matrix provides an appropriate mechanism for defining and implementing strict control for both static and dynamic association between processes and domains.
- When we switch a process from one domain to another, we are executing an operation (switch) on an object (the domain).
- We can control domain switching by including domains among the objects of the access matrix.

domain \ object	F_1	F_2	F_3	laser printer	D_1	D_2	D_3	D_4
D_1	read		read			switch		
D_2				print			switch	switch
D_3		read	execute					
D_4	read write		read write		switch			

Access Matrix

- When we change the content of the access matrix, one is performing an operation on an object: the access matrix.
- Again, one can control these changes by including the access matrix itself as an object.
- Actually, since each entry in the access matrix can be modified individually, we must consider each entry in the access matrix as an object to be protected.
- Now, we need to consider only the operations possible on these new objects (domains and the access matrix) and decide how we want processes to be able to execute these operations

domain \ object	F_1	F_2	F_3	laser printer	D_1	D_2	D_3	D_4
D_1	read		read			switch		
D_2				print			switch	switch
D_3		read	execute					
D_4	read write		read write		switch			

Access Matrix

- Processes should be able to switch from one domain to another. Switching from domain Di to domain Dj is allowed if and only if the access right switch ∈ access (i, j).
- Thus, in Figure, a process executing in domain D2 can switch to domain D3 or to domain D4 .
- A process in domain D4 can switch to D1 , and one in domain D1 can switch to D2
- Allowing controlled change in the contents of the access-matrix entries requires three additional operations: copy , owner , and control .

domain \ object	F_1	F_2	F_3	laser printer	D_1	D_2	D_3	D_4
D_1	read		read			switch		
D_2				print			switch	switch
D_3		read	execute					
D_4	read write		read write		switch			

Access Matrix

- The ability to copy an access right from one domain (or row) of the access matrix to another is denoted by an asterisk (*) appended to the access right
- The copyright allows the access right to be copied only within the column that is, for the object for which the right is defined.
- In **Figure a**, a process executing in domain D2 can copy the read operation into any entry associated with file F2 .
- Hence, the access matrix of **Figure a** can be modified to the access matrix shown in **Figure b**.

domain \ object	F_1	F_2	F_3
D_1	execute		write*
D_2	execute	read*	execute
D_3	execute		

(a)

domain \ object	F_1	F_2	F_3
D_1	execute		write*
D_2	execute	read*	execute
D_3	execute	read	

(b)

Access matrix with ***copy*** rights.

Access Matrix

- This scheme has two additional variants:
  - A right is copied from access (i, j) to access (k, j); it is then removed from access (i, j).
    - This action is a of a right, rather than a copy.
  - Propagation of the copyright may be limited. That is, when the right R* is copied from access (i, j) to access (k, j), only the right R (not R*) is created.
    - A process executing in domain Dk cannot further copy the right R.

object / domain	F_1	F_2	F_3
D_1	execute		write*
D_2	execute	read*	execute
D_3	execute		

(a)

object / domain	F_1	F_2	F_3
D_1	execute		write*
D_2	execute	read*	execute
D_3	execute	read	

(b)

Access matrix with ***copy*** rights.

Access Matrix

- A system may select only one of these three copyrights, or it may provide all three by identifying them as separate rights:
  - copy
  - transfer
  - limited copy .

- If access (i, j) includes the owner right, then a process executing in domain Di can add and remove any right in any entry in column j.

object / domain	F_1	F_2	F_3
D_1	owner execute		write
D_2		read* owner	read* owner write
D_3	execute		

(a)

object / domain	F_1	F_2	F_3
D_1	owner execute		write
D_2		owner read* write*	read* owner write
D_3		write	write

(b)

Access matrix with `owner` rights.

Access Matrix

- For example, in **Figure a**, domain D1 is the owner of F1 and thus can add and delete any valid right in column F1 .
- Similarly, domain D2 is the owner of F2 and F3 and thus can add and remove any valid right within these two columns.
- Thus, the access matrix of **Figure a** can be modified to the access matrix shown in **Figure b**

object / domain	F_1	F_2	F_3
D_1	owner execute		write
D_2		read* owner	read* owner write
D_3	execute		

(a)

object / domain	F_1	F_2	F_3
D_1	owner execute		write
D_2		owner read* write*	read* owner write
D_3		write	write

(b)

Access matrix with `owner` rights.

Access Matrix

- The copy and owner rights allow a process to change the entries in a column.
- A mechanism is also needed to change the entries in a row.
- The control right is applicable only to domain objects.
- If access (i, j) includes the control right, then a process executing in domain Di can remove any access right from row j.
- Then, a process executing in domain D2 could modify domain D4 , as shown in Figure

object / domain	F_1	F_2	F_3	laser printer	D_1	D_2	D_3	D_4
D_1	read		read			switch		
D_2				print			switch	switch control
D_3		read	execute					
D_4	write		write		switch			

Modified access matrix of Figure

Access Matrix

- The copy and owner rights provide us with a mechanism to limit the propagation of access rights.
- However, they do not give us the appropriate tools for preventing the propagation (or disclosure) of information.
- The problem of guaranteeing that no information initially held in an object can migrate outside of its execution environment is called the **Confinement Problem**.
- These operations on the domains and the access matrix are not in themselves important, but they illustrate the ability of the access-matrix model to allow us to implement and control dynamic protection requirements.
- New objects and new domains can be created dynamically and included in the access-matrix model
- System designers and users must make the policy decisions concerning which domains are to have access to which objects in which ways

- # Access Matrix

THANK YOU

Nitin V Pujari
Faculty, Computer Science
Dean - IQAC, PES University

nitin.pujari@pes.edu

For Course Deliverables by the Anchor Faculty click on www.pesuacademy.com