Capítulo 14: Proteção

Capítulo 14: Proteção

- Objetivos da proteção
- Princípios da proteção
- Domínio de proteção
- Matriz de acesso
- Implementação da matriz de acesso
- Controle de acesso
- Revogação de direitos de acesso
- Sistemas baseados em capacidade
- Proteção baseada em linguagem

Objetivos

- Discutir os objetivos e princípios de proteção em um sistema computadorizado moderno
- Explicar como os domínios de proteção, combinados com uma matriz de acesso, são usados para especificar os recursos que um processo pode acessar
- Examinar sistemas de proteção baseados em capacidade e linguagem

Objetivos da proteção

- O sistema operacional consiste em uma coleção de objetos, hardware ou software
- Cada objeto tem um nome exclusivo e pode ser acessado por um conjunto bem definido de operações
- Problema da proteção – garantir que cada objeto seja acessado corretamente e somente por aqueles processos que têm permissão para tal

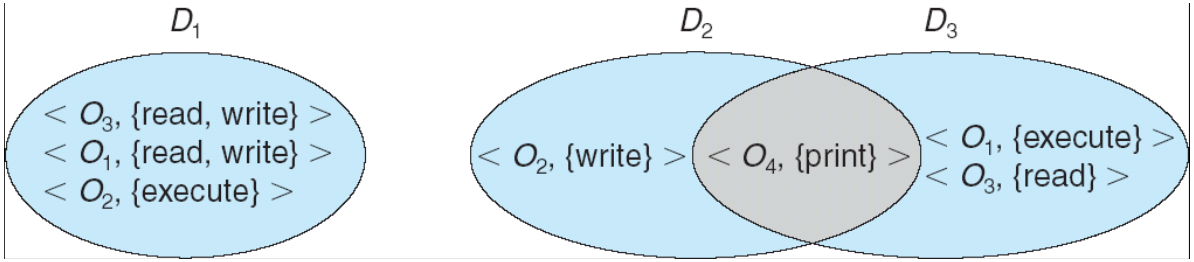
Princípios de proteção

- Princípio orientador – princípio do menor privilégio
  - Programas, usuários e sistemas devem receber apenas privilégios suficientes para realizar suas tarefas

Estrutura de domínio

- Direito de acesso = <*nome-objeto*, *conjunto-direitos*> onde *conjunto-direitos* é um subconjunto de todas as operações válidas que podem ser realizadas sobre o objeto.

- Domínio = conjunto de direitos de acesso



Matriz de acesso

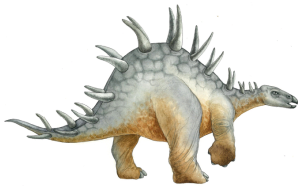
- Veja a proteção como uma matriz (*matriz de acesso*)
- Linhas representam domínios
- Colunas representam objetos
- *Access(i, j)* é o conjunto de operações que um processo executando no $Domínio_i$ pode invocar em $Objeto_j$

Matriz de acesso

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	

Uso de uma matriz de acesso

- Se um processo no Domínio D_i tenta realizar "op" sobre o objeto O_j, então "op" deve estar na matriz de acesso
- Pode ser alterada dinamicamente
  - Operações para incluir e excluir direitos de acesso
  - Direitos de acesso especiais:
    - *owner de* O_i
    - *copy op de* O_i *para* O_j
    - *control –* D_i *pode modificar direitos de acesso de* D_j
    - *transfer – troca do domínio* D_i *para* D_j



Uso da matriz de acesso (cont.)

- Projeto de matriz de acesso separa mecanismo da política.
  - Mecanismo
    - Sistema operacional oferece matriz de acesso + regras.
    - Garante que a matriz de acesso só é manipulada por agentes autorizados e que as regras são impostas estritamente.
  - Política
    - Usuário dita a política.
    - Quem pode acessar que objeto e em que modo.

Implementação da matriz de acesso

- **Tabela Global** - conjunto de triplas ordenadas <domínio, objeto, direitos>
- **Lista de permissões (por objeto)** - cada coluna é uma lista: <Domínio 1, Read, Write>;< Domínio 2, Read>; <Domínio 3 , Read>
- **Lista de capacidades (por domínio)** - cada linha é uma lista: <Objeto 1, Read>, <Objeto 4, Read, Write, Execute>, <Objeto 5, Read, Write, Delete, Copy>
- **Lock-key** – meio termo entre as duas últimas: cada objeto possui uma lista de padrões de bits (locks – fechaduras) e cada domínio possui uma lista de padrões de bits (keys - chaves)

Matriz de acesso com domínios como objetos

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			

Matriz de acesso com direitos *Copy (*)*

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)

Um processo executando em D_2 pode copiar a permissão read para qualquer entrada associada a F_2 (fig. (a) para (b))

Matriz de acesso com direitos *Owner*

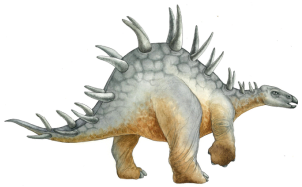
domain \ object	F_1	F_2	F_3
D_1	owner execute		write
D_2		read* owner	read* owner write
D_3	execute		

(a)

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

(b)

A direito de proprietário (“owner”) permite que o processo crie ou exclua novos direitos para o objeto (fig. (a) para (b))



Matriz de acesso com direito *control*

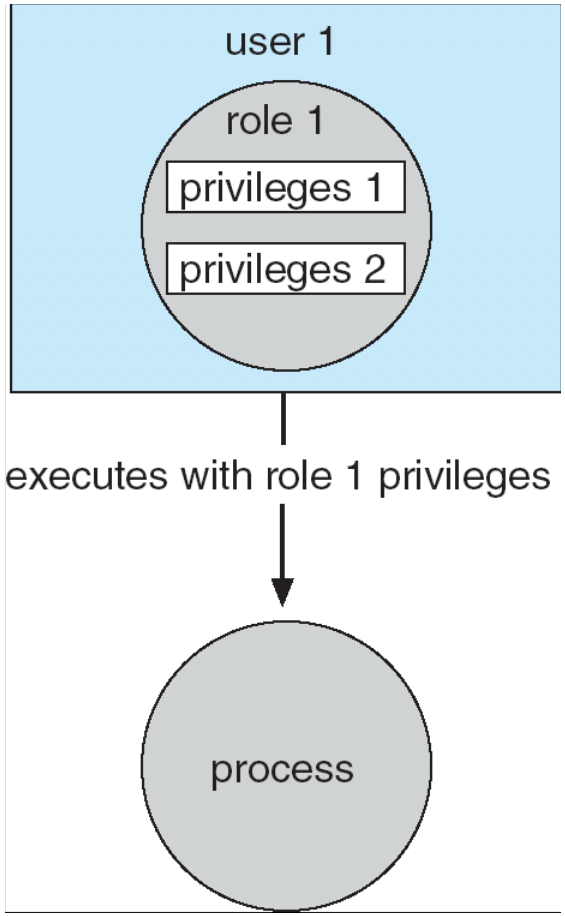
domain \ object	F_1	F_2	F_3	laser printer	D_1	D_2	D_3	D_4
D_1	read		read			switch		
D_2				print			switch	switch control
D_3		read	execute					
D_4	write		write		switch			

Um processo executando em D_2 pode alterar as permissões de D_4

Controle de acesso

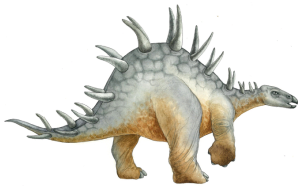
- Utiliza a mesma forma de controle de acesso a arquivos (owner, can read, can write, etc...)
- Solaris 10 oferece **controle de acesso baseado em *role*** para implementar menor privilégio
  - Privilégio é direito de executar chamada do sistema ou usar uma opção dentro de uma chamada do sistema
  - Pode ser atribuído a processos
  - Roles atribuídos a usuários concedendo acesso a privilégios e programas

Controle de acesso baseado em role no Solaris 10



Revogação de direitos de acesso

- **Imediata ou adiada** – revogação é feita imediatamente? Se não, é possível saber quando será feita?
- **Seletiva ou geral** – quando um direito de acesso a um objeto é revogado, afeta todos os usuários ou é possível especificar um grupo de usuários?
- **Parcial ou total** – Um subconjunto dos direitos associados a um objeto pode ser revogado ou temos que revogar todos os direitos?
- **Temporária ou permanente** – uma vez revogado, não pode mais ser concedido?



Proteção baseada em linguagem (exemplo: Java)

- Proteção é tratada pela Java Virtual Machine (JVM)
- Uma classe recebe um domínio de proteção quando é carregada pela JVM.
- O domínio de proteção indica quais operações a classe pode executar.
- Se um método de biblioteca for invocado e realizar uma operação privilegiada, a pilha é inspecionada para garantir que a operação pode ser realizada pela biblioteca.

Final do Capítulo 14