Proteção

Referência:

Silberschatz, Abraham. Sistemas Operacionais com Java. 7 ed. Rio de Janeiro: Elsevier, 2008.

- **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 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 fazer.**

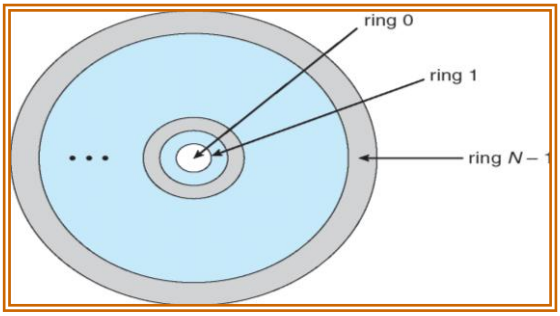
Implementação de domínio (UNIX)

- **Sistema consiste em 2 domínios:**
  - **Usuário**
  - **Supervisor**
- **UNIX**
  - **Domínio = user-id**
  - **Troca de domínio realizada via sistema de arquivos.**

- **Cada arquivo tem associado a ele um bit de domínio (bit setuid).**
- **Quando o arquivo é executado e setuid = on, então user-id é definido como o owner do arquivo sendo executado. Quando a execução termina, user-id é reiniciado.**

Implementação do domínio (MULTICS)

- **Sejam D_i e D_j dois anéis de domínio quaisquer.**
- **Se $j < I \Rightarrow D_i \subseteq D_j$**



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$_j$**

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 expandido para proteção dinâmica.**

- **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***
- **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

- **Cada coluna = Lista de controle de acesso para um objeto**
  Define quem pode realizar que operação.
  Domínio 1 = Read, Write
  Domínio 2 = Read
  Domínio 3 = Read
  ⋮
- **Cada Linha = Lista de capacidade (como uma chave) Para cada domínio, que operações são permitidas em quais objetos.**
  - **Objeto 1 – Read**
  - **Objeto 4 – Read, Write, Execute**
  - **Objeto 5 – Read, Write, Delete, Copy**

Controle de acesso

- **Proteção pode ser aplicada a recursos não de arquivo**
- **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**

Proteção baseada em linguagem

- **Especificação da proteção em uma linguagem de programação permite que a descrição de alto nível das políticas para a alocação e uso dos recursos.**
- **Implementação de linguagem pode oferecer software para imposição de proteção quando a verificação automática com suporte do hardware não está disponível.**
- **Interpretar especificações de proteção para gerar chamadas sobre qualquer sistema de proteção fornecido pelo hardware e sistema operacional.**