Proteção
& Conclusão do Curso
Sistemas Operacionais

Prof. Pedro Ramos
pramos.costar@gmail.com

Pontifícia Universidade Católica de Minas Gerais
ICEI - Departamento de Ciência da Computação

Proteção

Hoje: Proteção

- Objetivos da Proteção
- Domínio de Proteção
- Matriz de Acesso
- Sistemas Baseados em Capacidades (permissões ou credenciais)

Proteção

• O sistema operacional é composto por uma coleção de objetos, de hardware ou software.

• Cada objeto possui um **nome único e pode ser acessado por meio de um conjunto bem definido de operações**.

• Problema de **proteção – garantir que cada objeto seja acessado corretamente e apenas pelos processos que têm permissão para isso.**

Proteção: Estrutura de Domínio

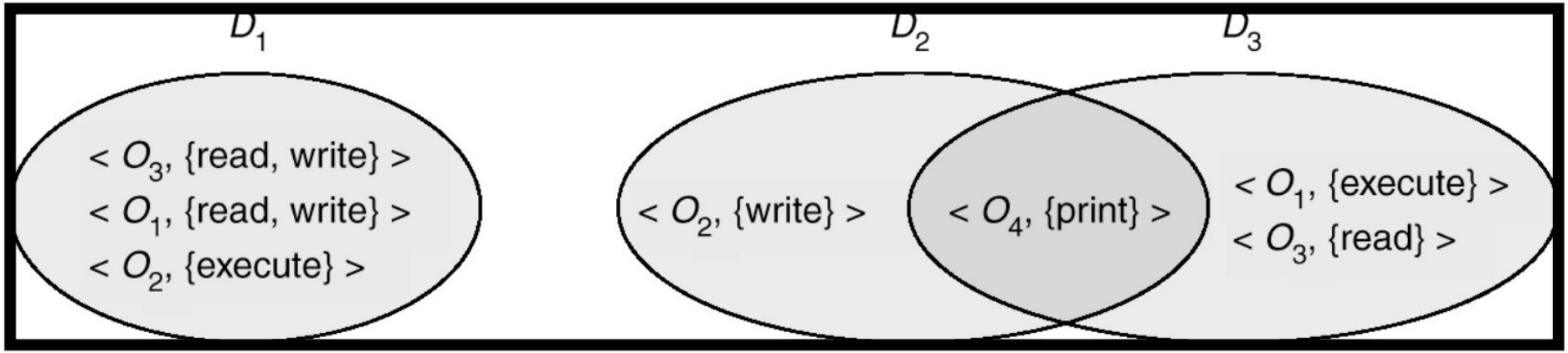
- Direito de acesso =
  <**nome-do-objeto**, **conjunto-de-direitos**>

em que o conjunto-de-direitos é um subconjunto de todas as operações válidas que podem ser realizadas sobre o objeto.

- Domínio = conjunto de direitos de acesso

  – associado a **usuários**, **grupos** de usuários e seus **processos**

Proteção: Estrutura de Domínio



Implementação de Domínio (UNIX)

- O sistema consiste em 2 domínios:
  - Usuário
  - Supervisor

- UNIX
  - Domínio = ID do usuário (user-id)
  - A troca de domínio é realizada por meio do sistema de arquivos.

- Cada arquivo possui um bit de domínio associado a ele (bit setuid).

- Quando o arquivo é executado e o setuid está ativado, o user-id passa a ser o do proprietário do arquivo executado. Quando a execução termina, o user-id é restaurado.

MATRIZ DE ACESSO

- A proteção é vista como uma matriz (matriz de acesso)
- As linhas representam os domínios
- As colunas representam os objetos
- Access(i, j) é o conjunto de operações que um processo executando no Domínio□ pode invocar sobre o Objeto□

MATRIZ DE ACESSO

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	

Sistemas Baseados em Capacidades (Exemplos Modernos)

- seL4 Microkernel
  – Utiliza um modelo estrito de capacidades para controlar acesso a todos os recursos do sistema (memória, threads, portas de comunicação, etc.).
  – Cada operação em um recurso só é possível se o processo possuir a capacidade apropriada.
  – Verificações de acesso em tempo de execução com forte garantia formal de segurança.

- WebAssembly (Wasm)
  – Em ambientes como navegadores ou runtimes, módulos só podem acessar recursos (como arquivos, rede) se forem explicitamente concedidas capacidades pelo host.
  – Segue o princípio do menor privilégio: o módulo só pode fazer o que as capacidades permitirem.
  – Exemplo: acesso ao sistema de arquivos só é permitido se o host fornecer uma capacidade de leitura específica.

Encerramento e Revisão da Disciplina

A prova final cobre:

- Maior ênfase em sistemas de memória, paginação, arquivos, entrada/saída (I/O) e sistemas distribuídos

- A prova final é abrangente (conteúdo completo da disciplina)

Visão Geral da Disciplina

- Processos e Threads
- Memória
- Entrada/Saída (I/O), Sistemas de Arquivos
- Redes e Sistemas Distribuídos

Proteção

Abstração de Hardware	Exemplos de Serviços do SO	Abstração para o Usuário
Processador	Gerenciamento de processos, escalonamento, traps, proteção, contabilidade, sincronização	Processo
Memória	Gerenciamento de memória, proteção, memória virtual	Espaços de endereçamento
Dispositivos de I/O	Concorrência com a CPU, tratamento de interrupções	Terminal, mouse, impressora, chamadas de sistema
Sistema de Arquivos	Gerenciamento de arquivos, persistência	Arquivos
Sistemas Distribuídos	Rede, segurança, sistema de arquivos distribuído	Chamadas de procedimento remoto, sistema de arquivos em rede

Destaques do Gerenciamento de Processos

- O que é uma troca de contexto (context switch)? O que acontece durante uma troca de contexto? O que pode causar uma troca de contexto?
- Qual a diferença entre um processo e uma thread?
- O que são os algoritmos FCFS (First-Come, First-Served), Round Robin, SJF (Shortest Job First) e Fila com Realimentação Multinível (Multilevel Feedback Queue)?
- O que é um processo I/O-bound? O que é um processo CPU-bound? Existe alguma razão para tratá-los de forma diferente no escalonamento?
- O que é uma thread? Qual a diferença entre threads de nível de usuário e de nível de kernel?
- O que é um semáforo? Quais são as três principais utilizações de um semáforo?
- O que é um monitor? O que é uma variável de condição?
- O que é espera ocupada (busy waiting)?
- Quais são as quatro condições necessárias para que ocorra um deadlock (interbloqueio)?
- Qual a diferença entre detecção de deadlock e prevenção de deadlock?
- Após detectar um deadlock, quais são as opções possíveis para se recuperar dele?

Destaques do Gerenciamento de Memória e Entrada/Saída (I/O)

- O que é memória virtual e por que a utilizamos?
- O que é paginação e o que é uma página?
- O que o sistema operacional armazena na tabela de páginas?
- O que é uma TLB (Translation Lookaside Buffer)? Como ela é utilizada?
- O que é uma falta de página (page fault), como o sistema operacional sabe que precisa tratá-la, e o que o sistema faz quando uma falta de página ocorre?
- Algoritmos de substituição de páginas: FIFO, MIN, LRU. Entenda como funcionam, vantagens e desvantagens de cada um.
- Como o sistema operacional se comunica com os dispositivos de I/O?
- Para que servem os buffers de I/O?
- Para que servem os caches de I/O? Como eles afetam as operações de leitura e escrita nos dispositivos de I/O?
- O que é o tempo de busca (seek time)?
- O que é a latência rotacional?
- O que é o tempo de transferência?
- Algoritmos de escalonamento de disco: FIFO, SSTF, SCAN, C-SCAN. Como funcionam, vantagens e desvantagens.

Gerenciamento de Memória

Tópicos que você deve entender:

O que é memória virtual e por que a usamos?

Estratégias de alocação de memória:
- Alocação contígua (algoritmos first-fit e best-fit)
- Paginação
- Segmentação
- Segmentação paginada

Gerenciamento de Memória

Para cada estratégia, compreenda os seguintes conceitos:

- Tradução de endereços
- Suporte de hardware necessário
- Como lidar com fragmentação
- Capacidade de crescimento dos processos
- Capacidade de compartilhar memória com outros processos
- Capacidade de mover processos
- Proteção de memória
- O que precisa acontecer em uma troca de contexto para suportar o gerenciamento de memória

Sistemas de Arquivos

Tópicos que você deve entender:

O que é um arquivo? O que é um tipo de arquivo?
Quais tipos de acesso são típicos para arquivos?
O que o sistema operacional faz ao abrir e fechar um arquivo?
O que é um diretório?
O que é um link?
O que acontece se a estrutura de diretórios for um grafo?
Como o sistema operacional suporta múltiplos usuários acessando arquivos compartilhados?

Estratégias para alocar arquivos no disco. Vantagens e desvantagens:

- Alocação *contígua*
- Alocação *encadeada*
- Alocação *indexada*

Sistemas de Entrada/Saída (I/O)

Tópicos que você deve entender:

- Acesso Direto à Memória (DMA - Direct Memory Access)
- Polling e Interrupções
- Cache e Buffer

Sistemas Distribuídos

Qual a diferença entre um sistema **distribuído** e um sistema **paralelo**?

Quais são as vantagens dos sistemas **distribuídos** em relação aos sistemas **isolados**?

Quais são as vantagens dos sistemas **isolados** em relação aos sistemas **distribuídos**?

Redes

O que é uma LAN?

O que é uma WAN?

Quais são as topologias de rede mais comuns? Quais são mais adequadas para WANs? E para LANs?

Como falhas em nós afetam as diferentes topologias de rede?

Quais são os custos de comunicação esperados para as diferentes topologias de rede?

O que são pacotes?

O que é uma pilha de protocolos de rede? O que é o TCP/IP?

Chamada de Procedimento Remoto (RPC)

O que é RPC?

Como o RPC difere de uma chamada de procedimento normal?

Que computações extras são necessárias para realizar um RPC em vez de uma chamada de procedimento normal?

Você usaria RPC para comunicar dois processos na mesma máquina?

Preparação para a Prova

• Você deve ter uma boa noção de como as **partes do sistema operacional se encaixam** e como mudanças em uma parte podem impactar outras.

- Não será necessário escrever código em Java.
- Não haverá perguntas sobre detalhes técnicos do Unix, Windows ou mac OS.
- Haverá questões fechadas que podem conter conteúdo de tudo estudado até aqui na disciplina..

PERGUNTAS E DÚVIDAS SOBRE O TRABALHO?

ou

Algum comentário?

REFERÊNCIAS

- **TANENBAUM, Andrew.** Sistemas operacionais modernos.
- **SILBERSCHATZ, Abraham et al.** Fundamentos de sistemas operacionais: princípios básicos.
- **MACHADO, Francis; MAIA, Luiz Paulo.** Arquitetura de Sistemas Operacionais.
- **CARISSIMI, Alexandre et al.** Sistemas operacionais.