

Especificação do Projeto 2 – Simulador de Paginação



Sistemas Operacionais

Prof. Lucas Figueiredo e Prof. Jamilson Bispo

Objetivo do Projeto

Desenvolver um simulador que implemente o conceito de paginação em sistemas operacionais. O simulador deve representar a memória física e virtual, gerenciar processos e suas páginas, e lidar com faltas de página.

Além disso, incluirá funcionalidades para interação com o sistema de arquivos do Linux e tratamento de interrupções.

Este projeto visa proporcionar uma compreensão prática dos conceitos estudados sobre gerenciamento de memória, sistema de arquivos e dispositivos de entrada/saída.

Definindo a Paginação

A paginação é uma técnica de gerenciamento de memória que permite que um sistema operacional use a memória secundária (como um disco rígido) como uma extensão da memória principal (RAM).

Quando um processo precisa acessar dados que não estão atualmente na memória principal, ocorre uma falta de página (*page fault*), e o sistema operacional transfere a página necessária da memória secundária para a memória principal.

A memória virtual por paginação divide tanto a memória lógica quanto a física em blocos de tamanho fixo chamados páginas e *frames*, respectivamente. As páginas são carregadas na memória principal apenas quando são referenciadas. Isso significa que partes do programa podem residir na memória secundária até serem necessárias.

Dessa forma o sistema operacional consegue gerenciar eficientemente o espaço de endereçamento virtual, permitindo que programas maiores do que a memória física disponível sejam executados e reduzindo o impacto da fragmentação interna nos processos.

Requisitos do Projeto

Os requisitos detalhados para cada etapa do projeto estão descritos abaixo e orientados seguindo um cronograma semanal de desenvolvimento.

A decisão sobre as estruturas de dados utilizadas para representar cada elemento do simulador (memória lógica, memória física, disco, páginas, frames, etc.) fica a critério dos alunos.

O projeto deve ser desenvolvido na linguagem C. Os alunos poderão utilizar quaisquer bibliotecas que julgarem necessárias.

As operações realizadas pelo simulador deverão contemplar um delay/sleep, possibilitando a análise do fluxo e andamento das operações realizadas pelo simulador. Esse delay deve ser maior para operações que envolvam a memória secundária (disco).

Semana 1: Implementação Básica da Paginação

- Implementar a divisão da memória lógica em páginas e da memória física em *frames*.
- Criar uma tabela de páginas para mapear páginas lógicas para *frames* físicos.
- Desenvolver funções para traduzir endereços lógicos em endereços físicos usando a tabela de páginas.
- **Arquivo de Configuração:** Os alunos deverão criar um arquivo de configuração que defina os parâmetros essenciais do simulador, como:
 - tamanho da memória
 - tamanho das páginas
 - processos (tamanho e acessos)
 - caminho de saída dos logs
 - parâmetro de delay (sleep) para operações, com um maior delay para operações que envolvem acesso à memória secundária, refletindo seu tempo de acesso mais lento.

A estrutura do arquivo de configuração fica a critério dos alunos, assim como o uso de outros parâmetros. Eles devem incluir instruções claras sobre como preencher e parametrizar este arquivo para uso do simulador.

Semana 2: Gerenciamento de Faltas de Página e Substituição

Implementar o manejo de faltas de página quando um processo tenta acessar uma página não carregada na memória física. O simulador deve simular um tempo maior para a operação de carregamento da página da memória secundária.

- Implementar uma política simples de substituição de páginas, como FIFO ou LRU. Considere o tempo necessário para mover dados entre a memória principal e a secundária.
- **Logs das Operações:** Gerar logs das operações do simulador, incluindo traduções de endereços e substituições de páginas (acessos à memória secundária).

Semana 3: Interação com Sistema de Arquivos e Tratamento de Interrupções

Interação com Sistema de Arquivos

- Implementar a funcionalidade para salvar os logs gerados pelo simulador em um arquivo.
- Adicionar a capacidade de carregar um estado inicial do simulador a partir do arquivo de configuração, permitindo reiniciar simulações ou carregar configurações predefinidas.
- A definição de uma configuração escolhida deverá ser implementada como um argumento de chamada no terminal.

Tratamento de Interrupções:

- Adicionar suporte para receber comandos durante a execução do simulador. Por exemplo, permitir que o usuário pause ou continue a simulação através da entrada do teclado.
- Para isso, os alunos deverão implementar um tratamento de sinais simples para tratar interrupções, como SIGINT (Ctrl+C) para pausar a simulação e SIGCONT para continuar.

Apresentação Final

Na semana entre os dias 18/11 e 22/11, em sua respectiva aula de laboratório, cada grupo deverá apresentar seu simulador, explicando o funcionamento de cada componente. A apresentação deve incluir:

- Descrever como as tabelas de páginas, memória física, e processos foram modelados e implementados.
- Apresentar o arquivo utilizado para inicializar o simulador, explicando sua estrutura e como ele é utilizado.
- Demonstrar como sinais são usados para pausar e continuar a execução do simulador.
- Apresentar logs que demonstrem o comportamento do sistema em termos de criação e finalização dos processos, tradução de endereços, faltas de página, substituição de páginas, e controle de recursos.

Critérios de Avaliação

O projeto será avaliado com base nos seguintes critérios, com uma pontuação total de **10 pontos**:

PROJETO ENTREGUE (6 PONTOS)

1. Funcionalidade e Correção (2 pontos)

- O simulador funciona conforme os requisitos propostos em cada semana.
- O código implementa corretamente os conceitos de paginação, incluindo a tradução de endereços e o manejo de faltas de página.
- O projeto evita problemas comuns em simulações, como erros de mapeamento ou falhas na substituição de páginas.

Configuração e Flexibilidade (1 ponto)

- O simulador utiliza um arquivo de configuração para definir parâmetros essenciais, como tamanho da memória, tamanho das páginas, processos e *delay* nas operações.
- O arquivo de configuração é bem estruturado e permite fácil modificação dos parâmetros para diferentes cenários.
- Os alunos fornecem um template claro do arquivo de configuração e instruções sobre como preenchê-lo.

Criação e Organização dos Logs (1 ponto)

- O sistema gera logs que acompanham as operações do simulador, incluindo traduções de endereços, faltas de página e substituições.
- Os logs são bem estruturados e fornecem informações claras e úteis para acompanhar o comportamento do sistema.
- A organização dos logs permite identificar facilmente eventos críticos como faltas de página e substituições.

Documentação (2 pontos)

- O código é bem documentado com comentários explicativos, especialmente em áreas que envolvem lógica complexa ou cálculos.
- O grupo entrega um documento descrevendo a estrutura do simulador, a função de cada parte do código e as decisões de design.
- A documentação cobre tanto a implementação quanto o comportamento do sistema durante a execução.

APRESENTAÇÃO (4 PONTOS)

Clareza e Organização da Apresentação (2 pontos)

- A apresentação é clara, organizada e cobre todos os aspectos do projeto.
- O grupo explica de forma coerente como cada parte do simulador foi implementada, destacando a lógica da paginação e substituição.
- O grupo justifica as escolhas feitas ao longo do desenvolvimento, como o uso de determinadas políticas de substituição.

2. Demonstração do Funcionamento do Sistema (1 ponto)

- O grupo demonstra o simulador funcionando ao vivo ou apresenta resultados que comprovam o correto funcionamento.
- A demonstração inclui a tradução correta dos endereços e o manejo das faltas de página.
- A demonstração foca em mostrar como o sistema gerencia recursos limitados e resolve problemas relacionados à paginação.

3. Capacidade de Responder Perguntas (1 ponto)

- O grupo responde adequadamente às perguntas feitas sobre a implementação, a lógica da paginação e a resolução de problemas associados.
- Todos os membros do grupo demonstram conhecimento sobre o funcionamento do projeto e a lógica das suas implementações.

Entrega

Os grupos deverão entregar, no moodle:

- O código-fonte completo do simulador.
- O arquivo de configuração utilizado nos testes.
- Um documento explicativo descrevendo a estrutura do simulador, as decisões tomadas durante o desenvolvimento e os resultados obtidos.
- Logs gerados durante as simulações.

Havendo quaisquer dúvidas sobre o projeto, entrar em contato com o Prof. Lucas Figueiredo em aula, por e-mail ou Moodle.