Cache de blocos

Este projeto faz parte do PPOS v2.

Este projeto tem por objetivo implementar um cache em memória RAM para os blocos lidos e/ou escritos no disco.

O problema

O acesso a disco é uma operação demorada, pois o disco é lento. Para melhorar o desempenho, pode-se guardar uma cópia em RAM dos blocos lidos ou escritos no disco, para agilizar acessos futuros. As estratégias de *caching* a implementar neste projeto são as seguintes:

- *Read-through*: ao **ler um bloco** do disco, guarda-se uma cópia dele no cache em RAM; **futuras leituras** devem consultar o cache para ver se o conteúdo do bloco desejado está nele.
- *Write-through*: ao **escrever um bloco** no disco, guarda-se uma cópia do mesmo no cache em RAM; **futuras leituras** devem consultar o cache para ver se o conteúdo do bloco desejado está nele nele.



Este projeto implementa um cache com escrita *síncrona*, que é mais simples. Em um cache com escrita assíncrona (ou *write-back cache*), as tarefas escreveriam somente no cache, deixando a atualização do bloco no disco para uma tarefa específica de gestão de cache.

O cache pode ser visto como um vetor ou lista de blocos com capacidade limitada, geralmente inferior ao tamanho do disco. Se ele estiver cheio e um novo bloco precisar ser armazenado, outro bloco deverá ser descartado do cache antes. Neste projeto, deve-se usar o algoritmo LRU (*Least-Recently Used*) para escolher o bloco a descartar do cache. Em outras palavras, quando for necessário remover um bloco do cache, deve ser removido aquele que **está há mais tempo no cache sem ser acessado**.

Interface de acesso ao cache

O acesso ao cache de disco pelas tarefas é feito através das funções definidas em `kernel/cache.h` e implementadas (pelo aluno) em `kernel/cache.c`:

Iniciar o subsistema

```
void cache_init()
```

Esta função, chamada na inicialização do PPOS (em `kernel/ppos.c`) inicia o subsistema de cache de disco.

Leitura de bloco com cache

```
int cache_read(int block, void* buffer)
```

Parâmetros:

- `block`: número do bloco a ler do disco (entre 0 e *número de blocos* - 1);

- `buffer`: endereço do *buffer* onde devem ser colocados os dados lidos do disco; esse *buffer* deve ter capacidade para `block_size` bytes.
- retorno: 0 em caso de sucesso ou -1 em caso de erro.

Essa função faz a leitura de blocos usando o cache, da seguinte forma:

1. procura o bloco no cache
2. se encontrou o bloco
    1. copia conteúdo do cache para o buffer (usando `mem_copy`)
3. senão
    1. lê o bloco do disco (usando `block_read`)
    2. obtém uma entrada livre no cache (ou libera uma)
    3. copia o conteúdo do bloco lido para o cache
4. atualiza a data de acesso daquele bloco no cache

Escrita de bloco com cache

```
int cache_write(int block, void* buffer)
```

Parâmetros:

- `block`: número do bloco a escrever no disco (entre 0 e *número de blocos* - 1);
- `buffer`: endereço do *buffer* com os dados a escrever no disco; esse *buffer* deve ter capacidade para `block_size` bytes.
- retorno: 0 em caso de sucesso ou -1 em caso de erro.

Essa função faz a escrita de blocos usando o cache, da seguinte forma:

1. se o bloco não estiver no cache ainda
    1. obtém uma entrada livre no cache (ou libera uma)
2. copia o conteúdo do bloco no cache (usando `mem_copy`)
3. atualiza a data de acesso daquele bloco no cache
4. escreve o bloco no disco (usando `block_write`)

A implementação

Este projeto usa como base o projeto de acesso ao disco desenvolvido anteriormente. Neste projeto você deve:

- Implementar as funções de acesso ao cache;
- Implementar a política LRU, para liberar entradas se o cache estiver cheio;
- Gerenciar a exclusão mútua no acesso ao cache.



Existe o risco de que duas tarefas tentem acessar/atualizar a mesma entrada do cache simultaneamente, o que pode levar à corrupção dos dados ou do cache. Isso deve ser evitado com semáforos.

Outras informações

- Duração estimada: 4 horas.
- Dependências:
    - Tarefas cooperativas

- Despachante de tarefas
- Preempção por Tempo
- Tarefas que esperam
- Tarefas que dormem
- Spinlocks e Semáforos
- Acesso ao disco

From:
https://wiki.inf.ufpr.br/maziero/ - **Prof. Carlos Maziero**

Permanent link:
https://wiki.inf.ufpr.br/maziero/doku.php?id=ppos-v2:cache_de_blocos

Last update: **2026/07/29 18:03**