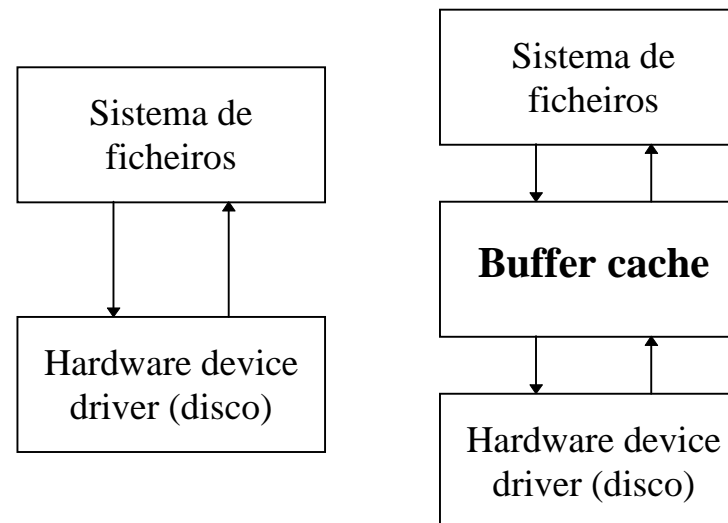


Buffer Cache

Acesso a disco é muito mais lento do que o acesso a memória.

Buffer cache minimiza os acessos a disco mantendo em memória aqueles blocos mais recentemente utilizados. O conjunto de buffers em memória e as operações de gestão desses buffers constituem a “buffer cache”



Operação de leitura : Se existe uma cópia do bloco na cache ela é utilizada, senão é o bloco é lido do disco para um dos buffers da cache.

Operação de escrita : É mantida uma cópia do bloco na cache. Nem sempre o bloco é imediatamente escrito em disco.

Os buffers

O número de buffers é configurável de acordo com a memória disponível e o tipo de utilização do sistema.

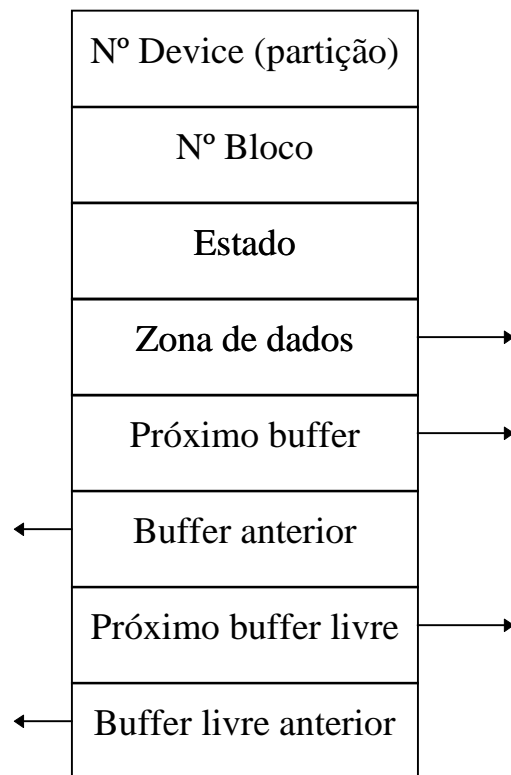
Cada buffer é constituído por 2 partes :

- Zona que contém a cópia de um bloco do disco. O mesmo buffer pode guardar blocos de disco diferentes em instantes diferentes.
- Cabeçalho que contém informação de controle sobre esse buffer.

O SO identifica o conteúdo do buffer através do exame do seu cabeçalho.

A um bloco de disco apenas pode estar associado um buffer.

Estrutura do cabeçalho de um buffer

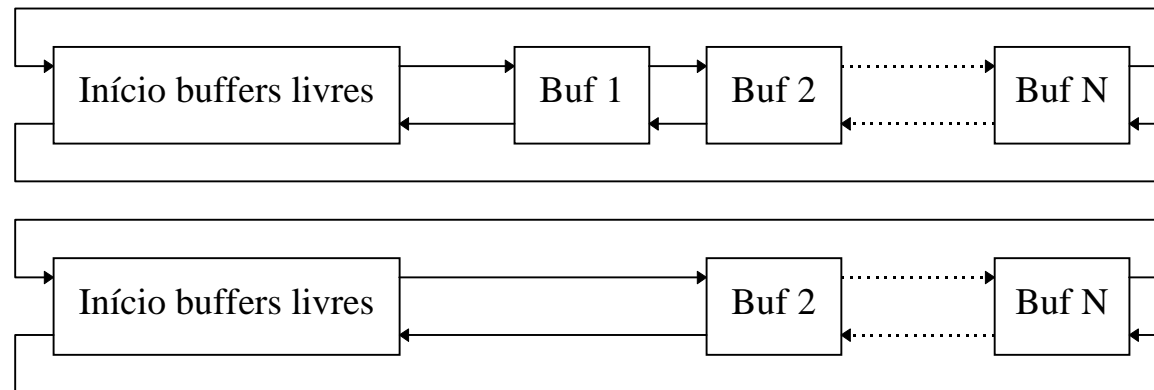


Estado de um buffer :

- Locked ou busy (unlocked ou free)
- Contém dados válidos
- O SO deve escrever o buffer em disco antes de o atribuir a outro bloco (delayed-write ou dirty)
- Em execução de uma operação de escrita ou leitura
- Um processo está à espera que o buffer fique livre

Lista de buffers livres.

A buffer cache atribui buffers através de um algoritmo do tipo LRU (Least Recently Used).



Na inicialização do sistema todos os buffers são colocados na lista de buffers livres.

Retirar um buffer da lista :

- Se qualquer buffer serve é retirado primeiro buffer da lista
- Se queremos retirar da lista o buffer que contém o bloco número B podemos retirar esse buffer do meio, ou do fim, da lista.

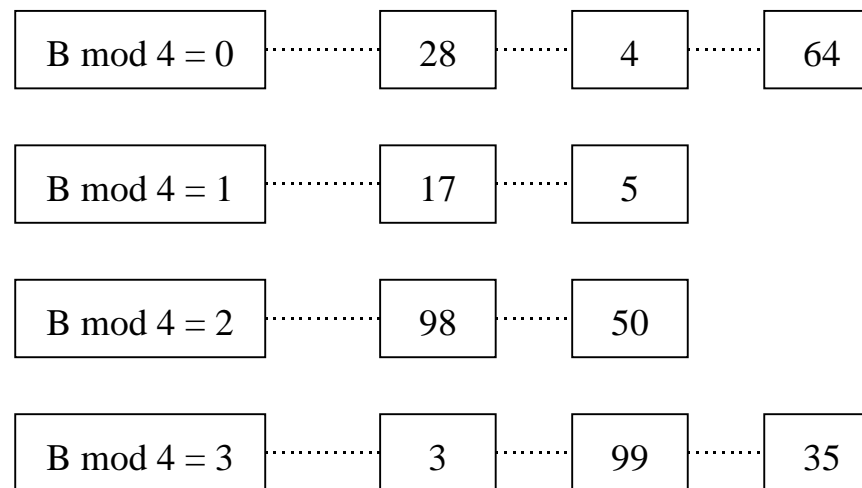
Inserir um buffer na lista :

- Geralmente o buffer é inserido no fim da lista.

Tabelas de hashing

Para procurar um determinado bloco B na cache não é necessário percorrer todos os buffers da cache.

Os buffers estão organizados em listas biligadas independentes de acordo com o n° do bloco e o n° do device que contêm.



Todos os buffers pertencem a uma e uma só lista de hashing.

Em cada lista não pode haver 2 vezes o mesmo bloco.

Um buffer pode estar simultaneamente na lista de buffers livres e numa lista de hashing.

Alocação de um buffer

Primitivas de mais alto nível determinam qual o device o e bloco a aceder.

Quer para operações de leitura, quer para operações de escrita é utilizado o algoritmo getblk para alocar um buffer para um bloco.

Cenários de execução do algoritmo getblk :

1. O bloco está na sua lista de hash e está livre.
2. O bloco não está na lista de hash, getblk aloca um buffer da lista de buffers livres e lê o bloco.
3. O bloco não está na lista de hash, mas o 1º bloco da lista de buffers livres está marcado “delayed-write”, esse buffer tem de ser escrito em disco e outro buffer da lista de blocos livres deve ser alocado.
4. O bloco não está na lista de hash e a lista de buffers livres está vazia.
5. O bloco está na lista de hash mas está “locked”

Vantagens e desvantagens da Buffer cache

- Minização do número de transferências entre o disco e a memória (cache hits).
- O acesso a disco é tratado de uma forma uniforme. Existe apenas uma interface para aceder a dados de um disco.
- Ajuda a manter a coerência dos blocos. Se 2 processos tentam aceder ao mesmo bloco a Buffer cache ordena os acessos.
- Torna o sistema mais vulneravel a “crashes”.
- Torna a transferência de grandes quantidades de informação mais lenta.

Bibliografia

“The Design of the UNIX Operating System”, M. J. Bach, Prentice-Hall, 1996.