

# Sistemas de Arquivos: Otimizações

**Prof. Dr. Márcio Castro**  
marcio.castro@ufsc.br



# Introdução

- O acesso a dados no **disco** é **muito mais lento** do que o acesso a dados em **memória**
- Logo, os sistemas de arquivo implementam **otimizações para melhorar o desempenho geral do sistema**
- **Otimizações**
  - 1 Buffer cache
  - 2 Leitura antecipada de blocos

1

Buffer cache

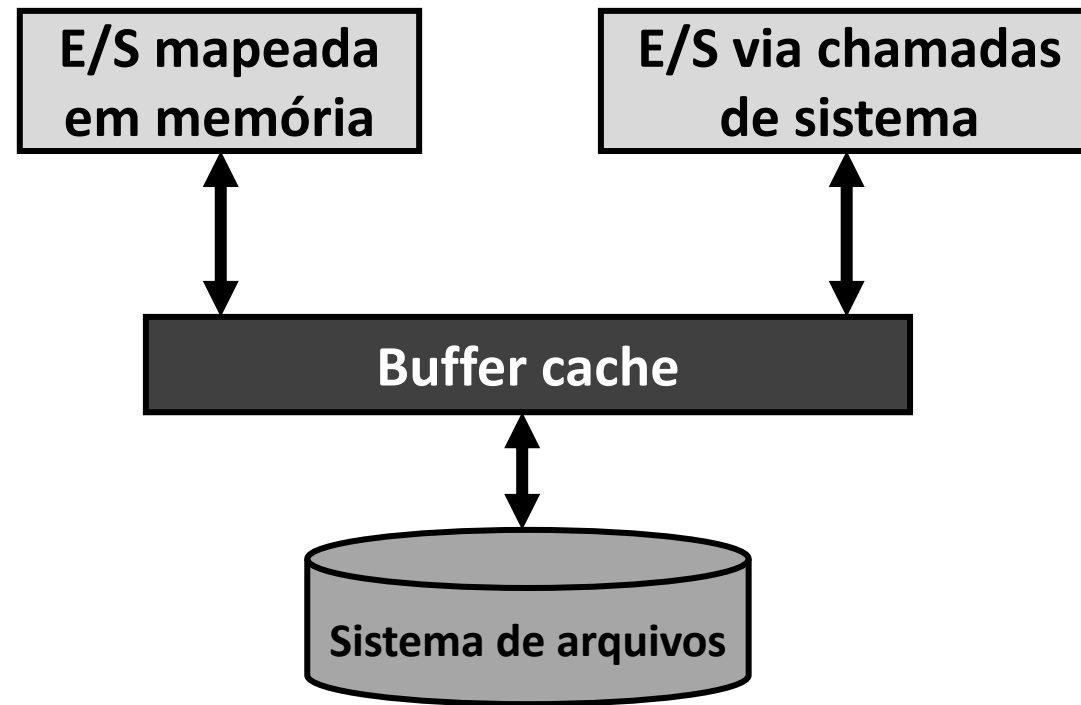
---

# Buffer cache

- **Caches em hardware** são utilizadas em arquiteturas de computadores para **reduzir o custo de acesso** a meios de armazenamento mais lentos
  - Caches de **dados e instruções (L1, L2 e L3)**
  - Caches para **tradução de endereços (TLBs)**
- A **mesma estratégia** é adotada em sistemas de arquivos, para evitar o alto **custo envolvido no acesso ao disco**
  - Porém, a cache é implementada **em software**

# Buffer cache

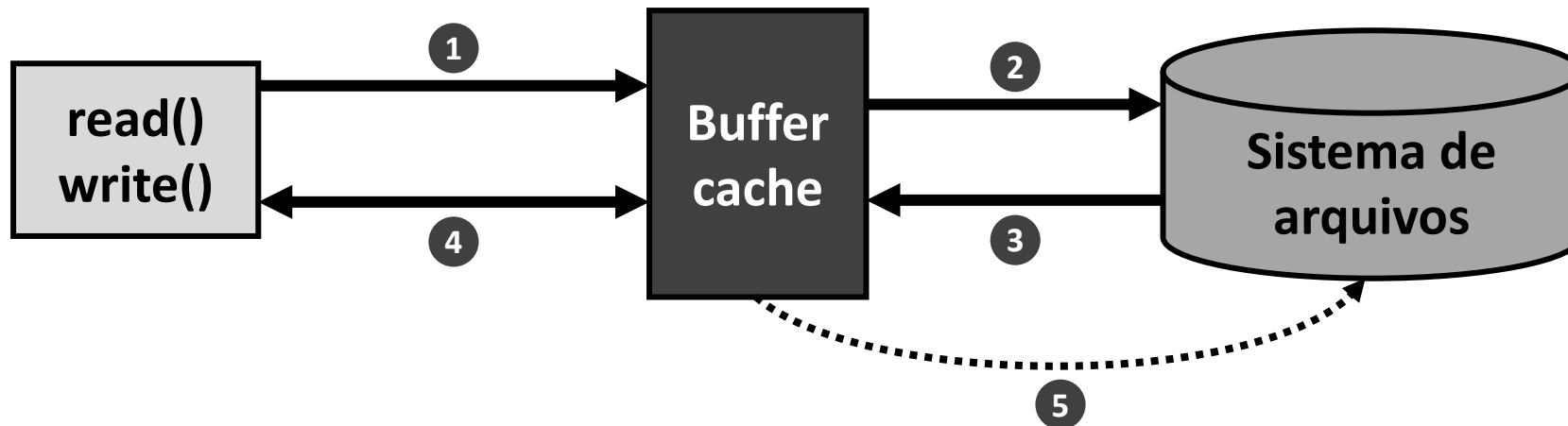
- É uma estrutura de dados que mantém em memória os blocos do disco que estão sendo mais utilizados no momento



# Buffer cache

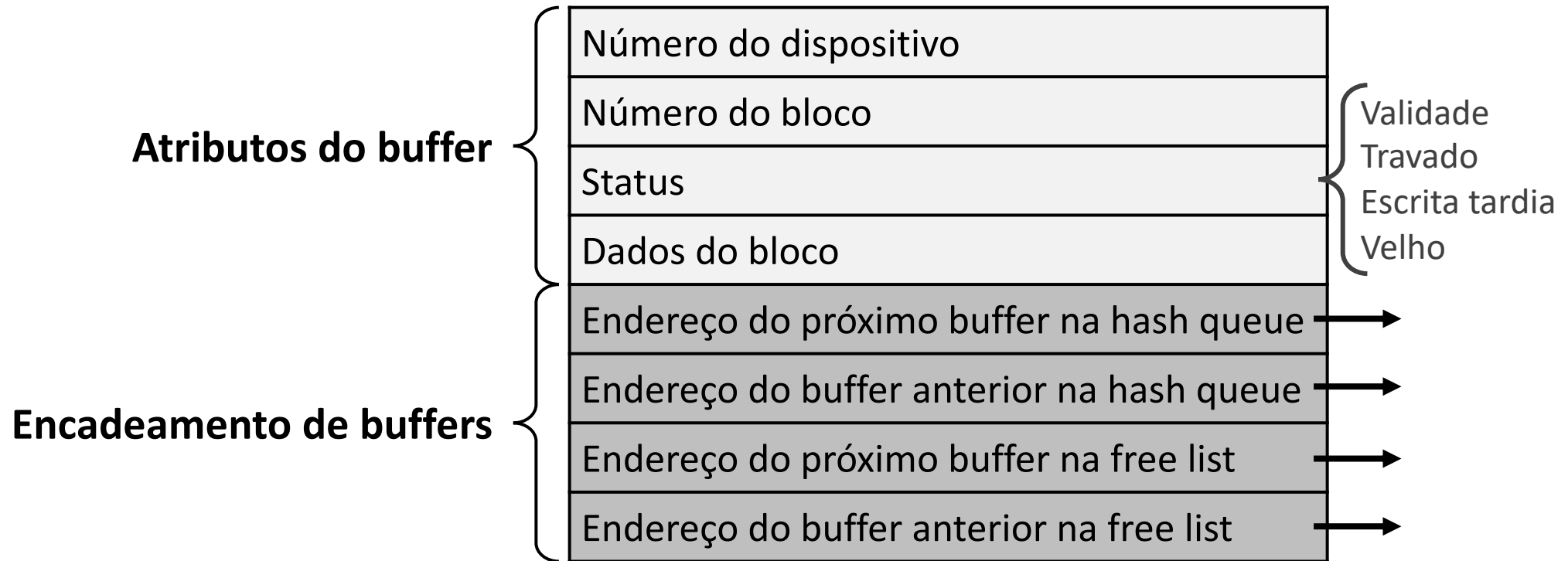
## ▪ Funcionamento geral da buffer cache

- 1 Uma operação de **leitura** ou **escrita** em um bloco do disco é requisitada à **buffer cache**
- 2 Se os dados do bloco **não estão na buffer cache**, uma **solicitação** é feita ao disco
- 3 Os **dados** do bloco são **copiados** para um **buffer** da buffer cache
- 4 Operações de escrita e leitura são realizadas **nos buffers em memória**
- 5 Buffers com conteúdo **modificado** são eventualmente **escritos de volta no disco**



# Buffer cache

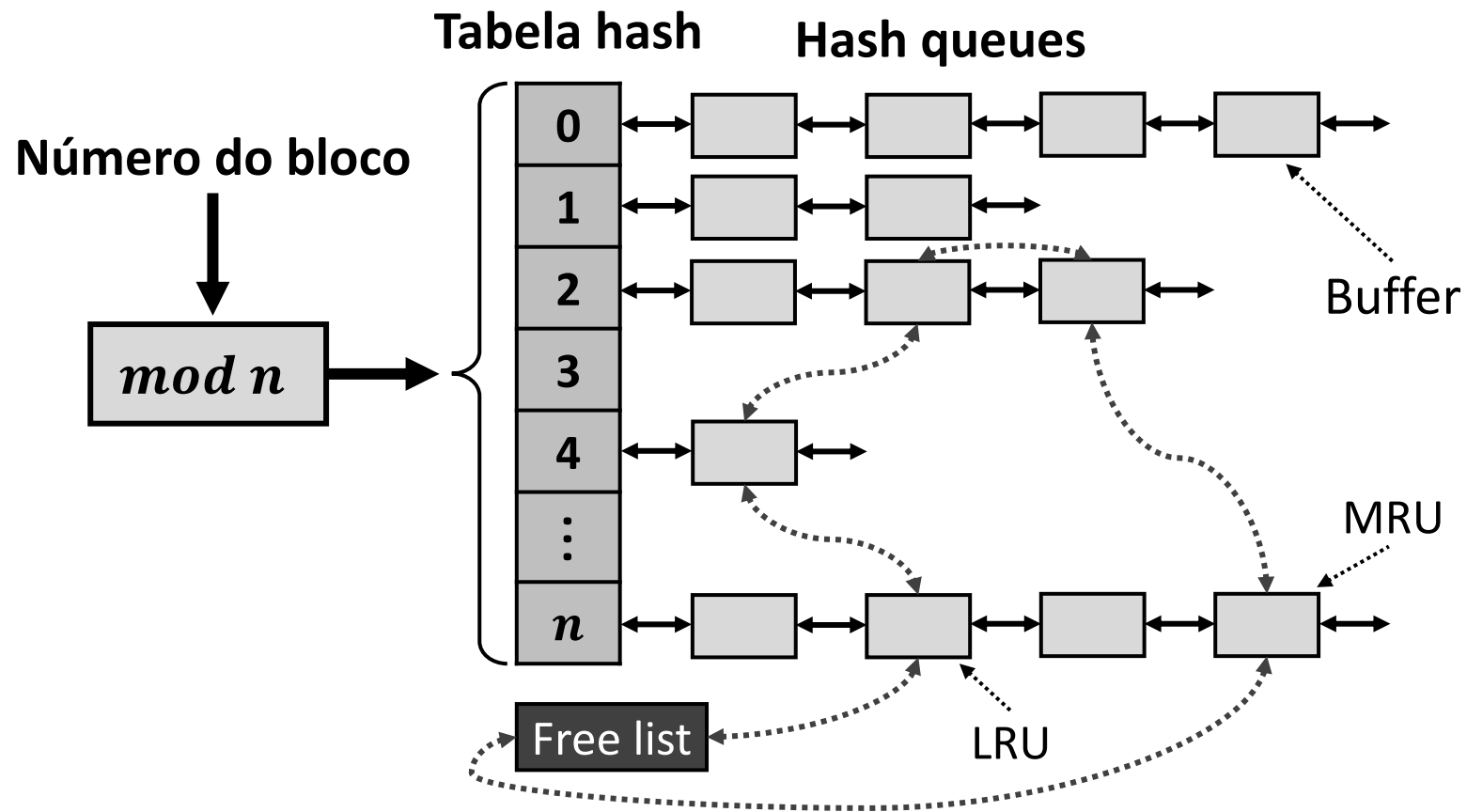
## Estrutura interna do buffer



# Buffer cache

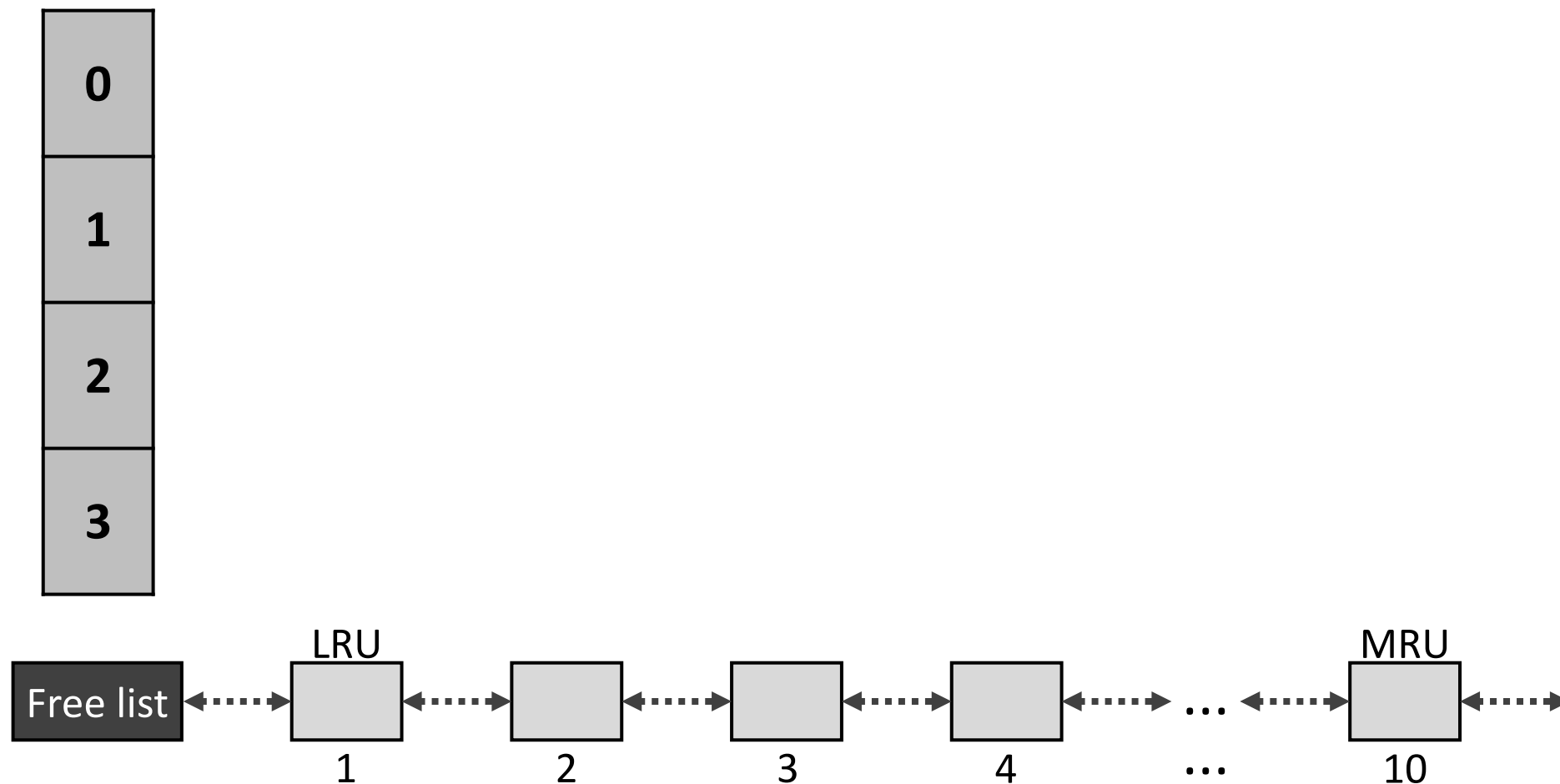
- **Estrutura de dados da buffer cache**

- Uma **tabela hash** com lista duplamente encadeada de colisões (**hash queues**)
- **Lista duplamente encadeada** de buffers livres (**free list**)

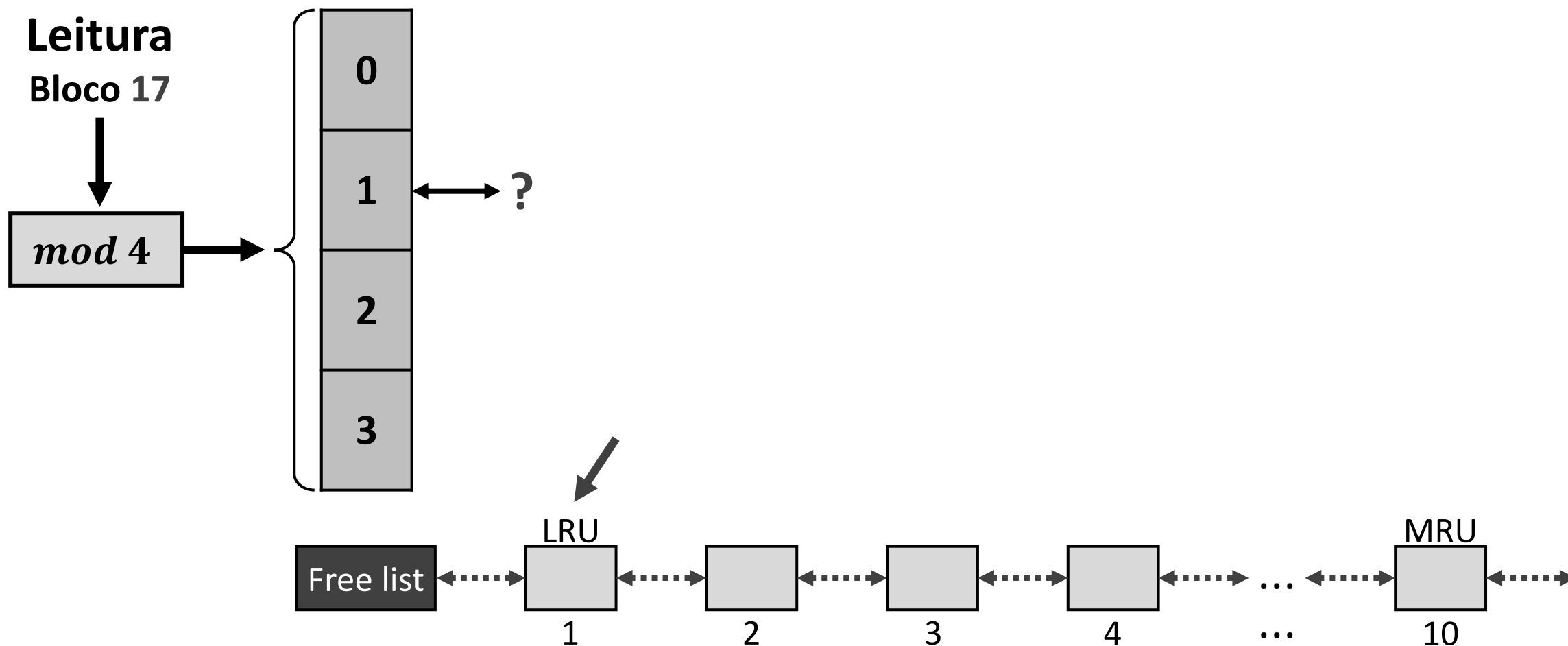




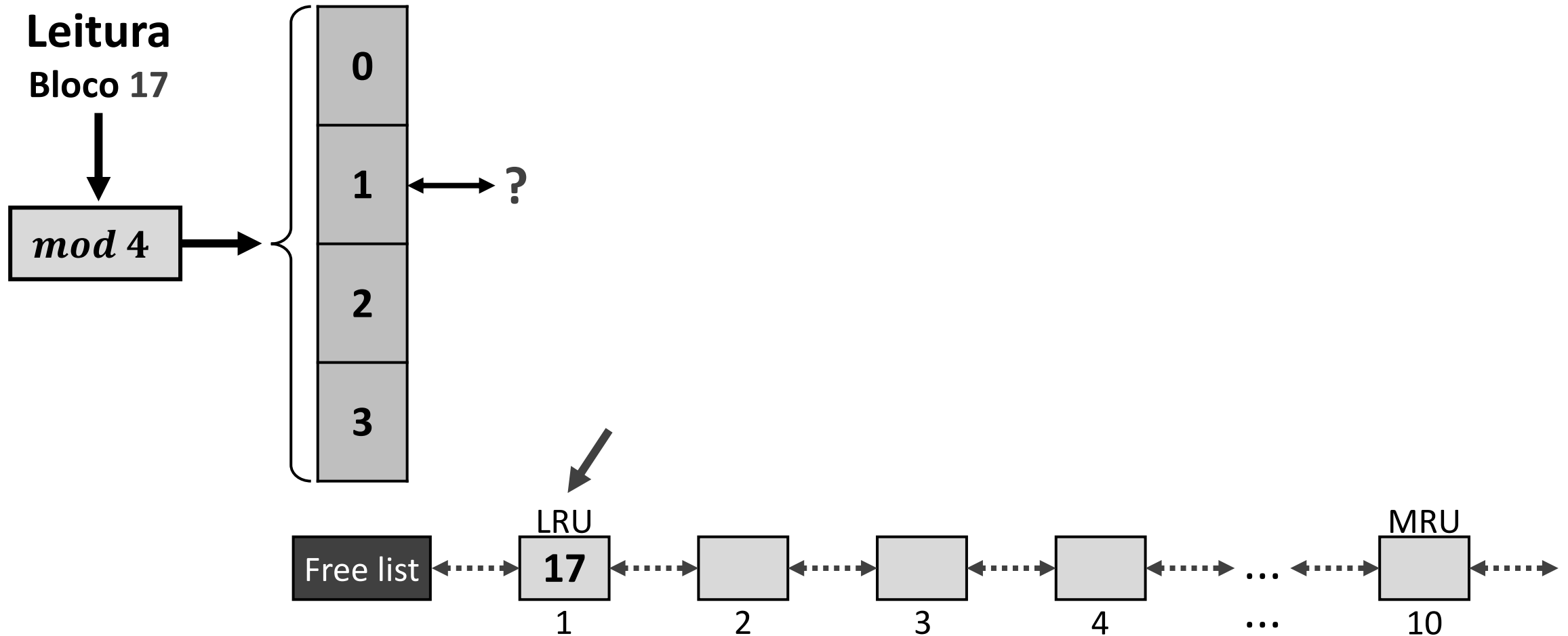
# Exemplo de alocação inicial de buffers



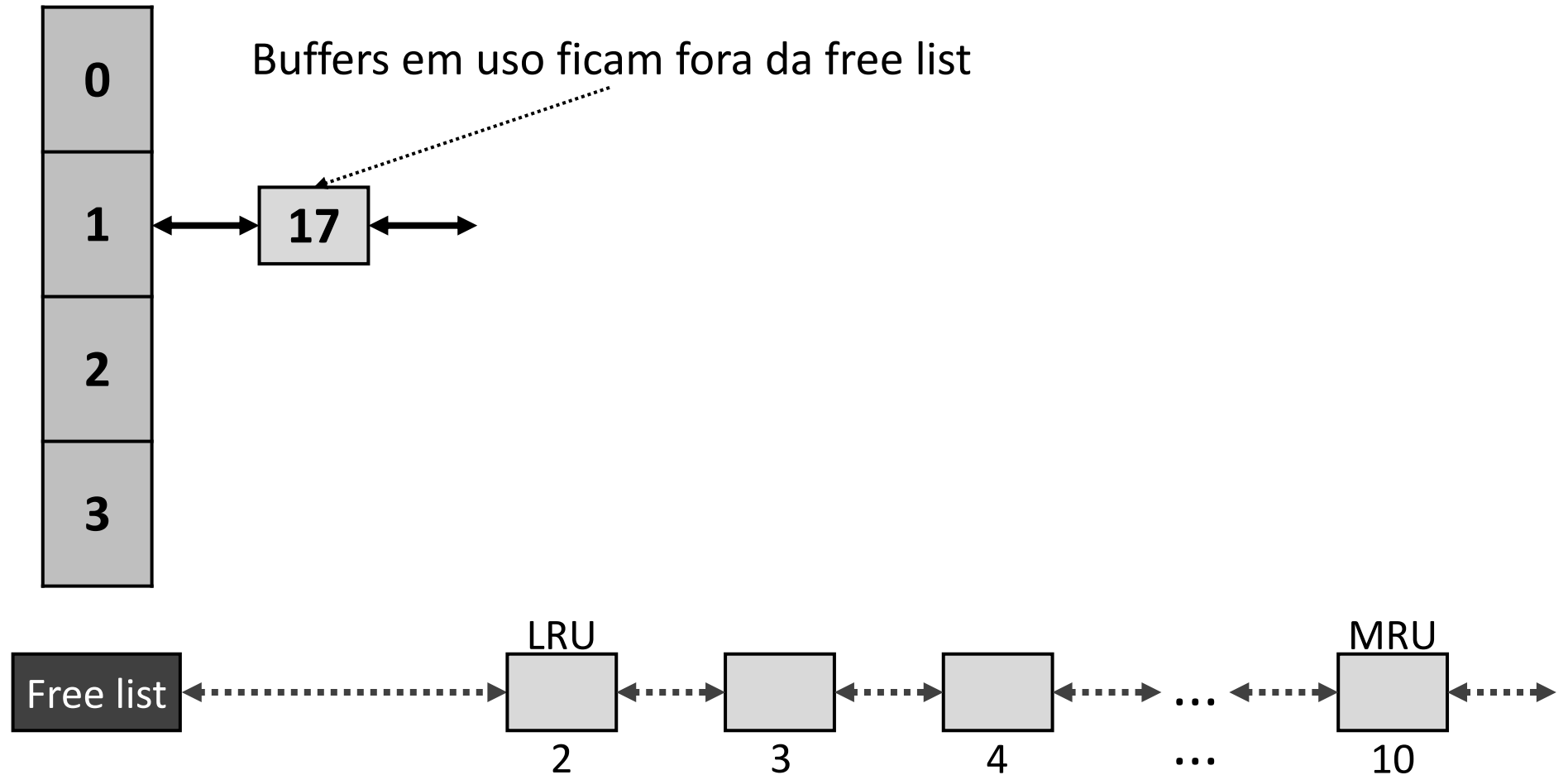
# Exemplo de alocação inicial de buffers



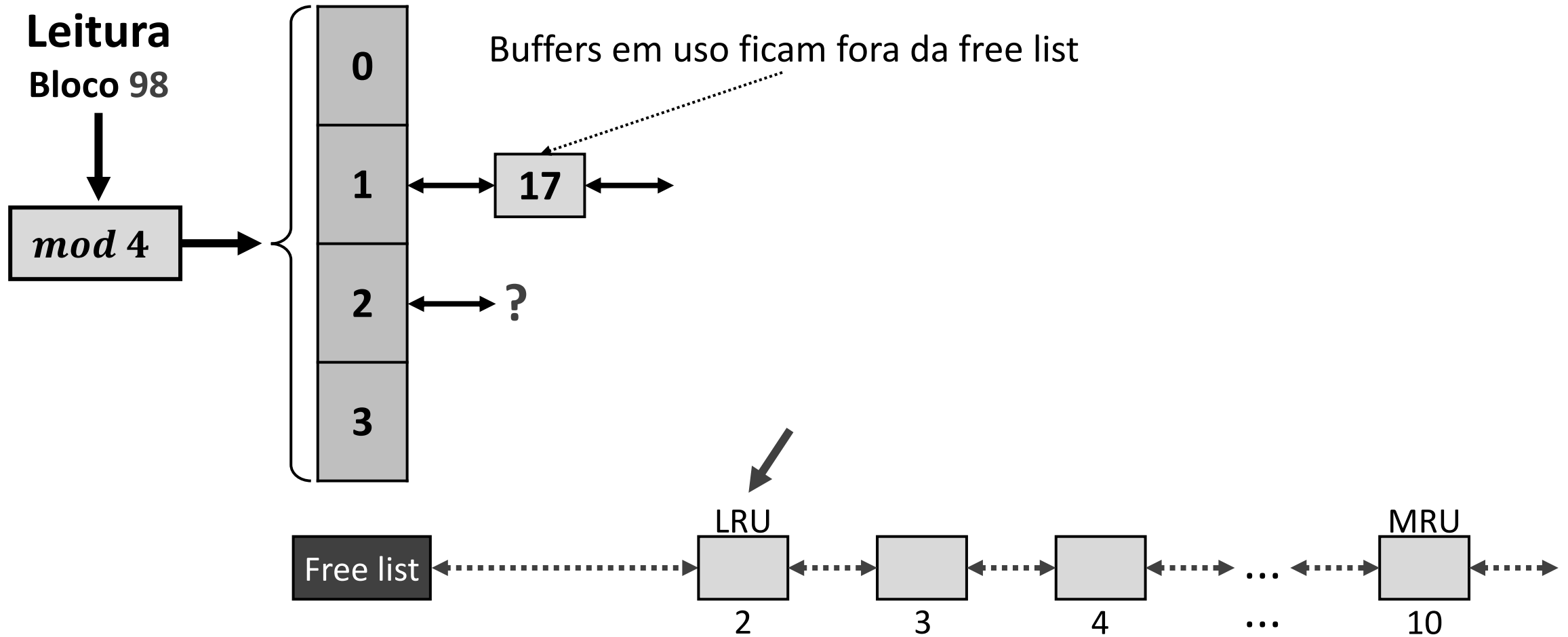
# Exemplo de alocação inicial de buffers



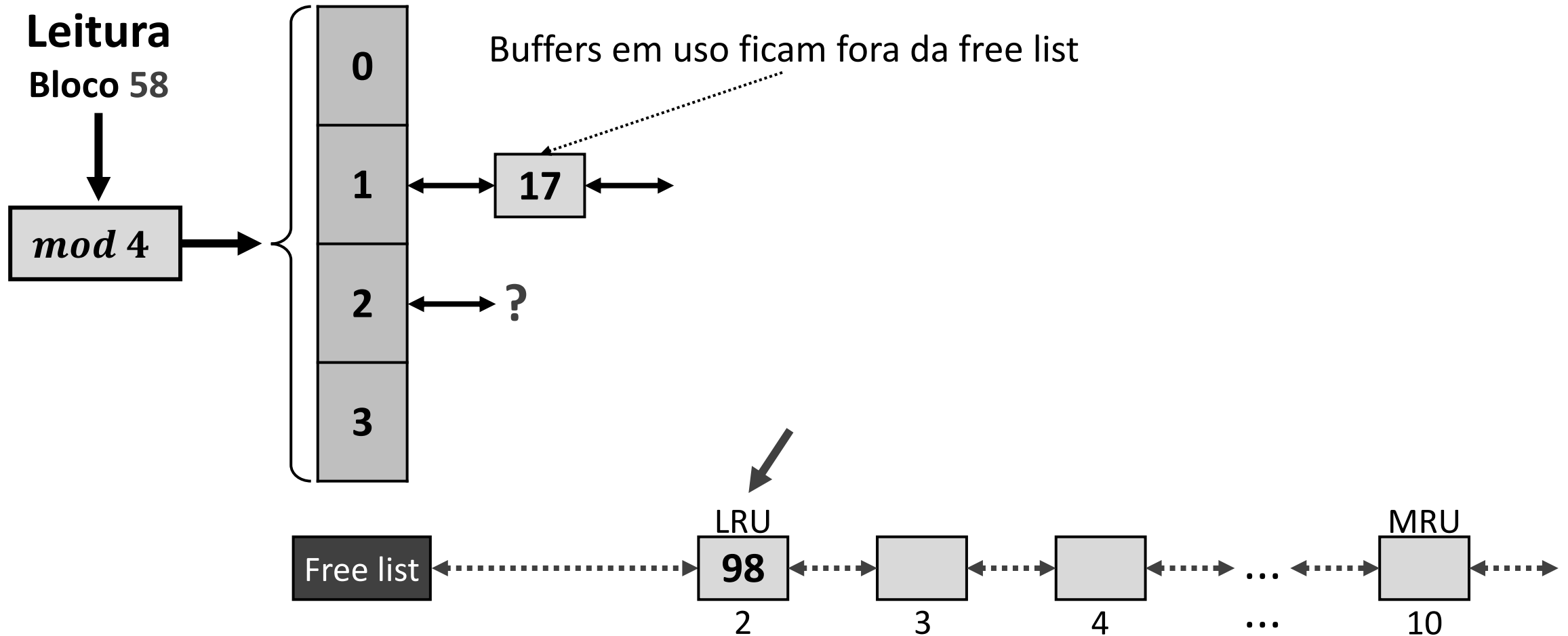
# Exemplo de alocação inicial de buffers



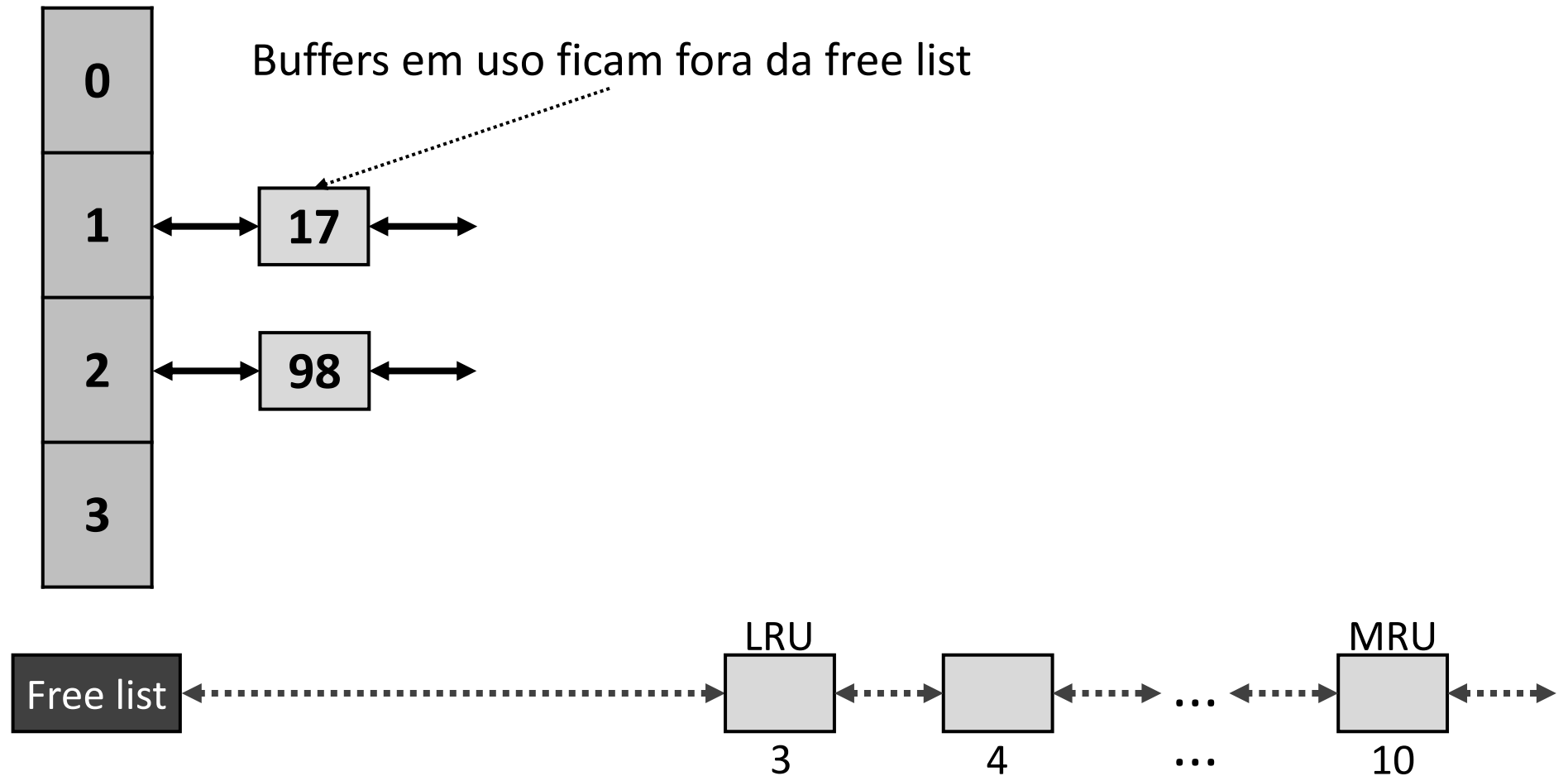
# Exemplo de alocação inicial de buffers



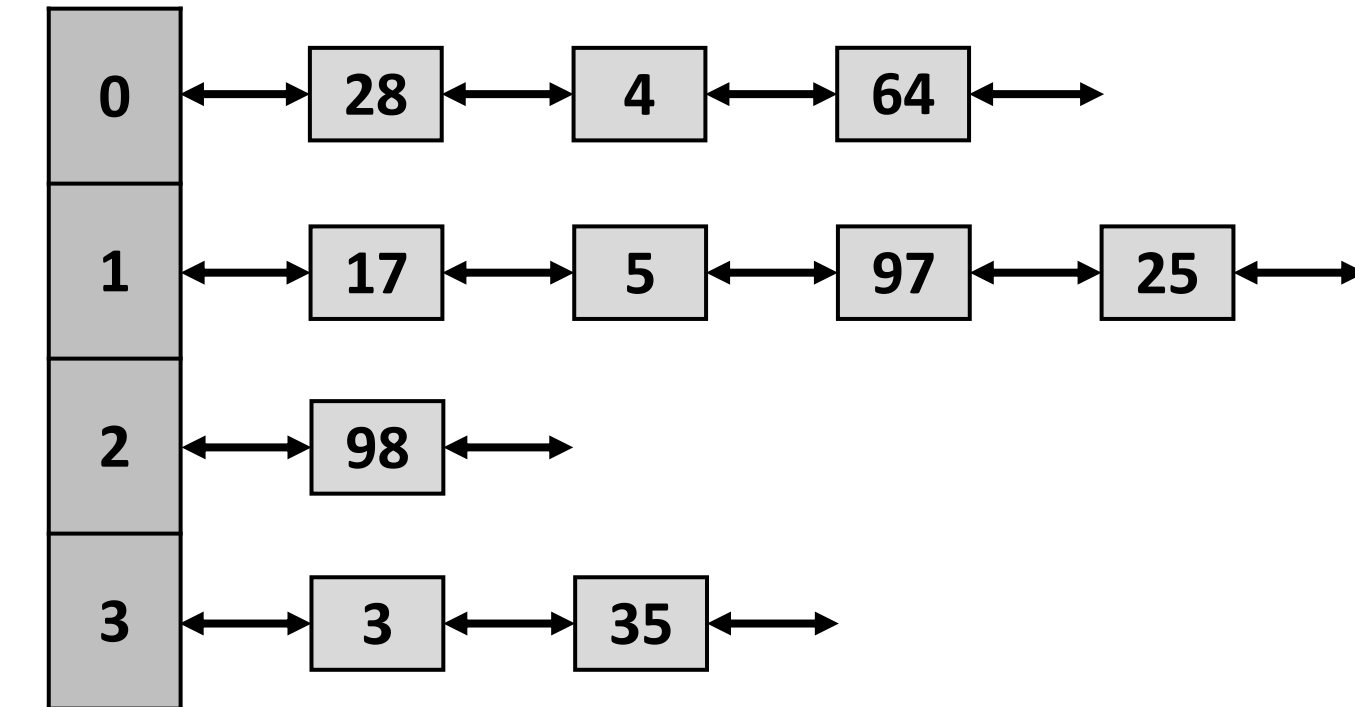
# Exemplo de alocação inicial de buffers



# Exemplo de alocação inicial de buffers



# Exemplo de alocação inicial de buffers

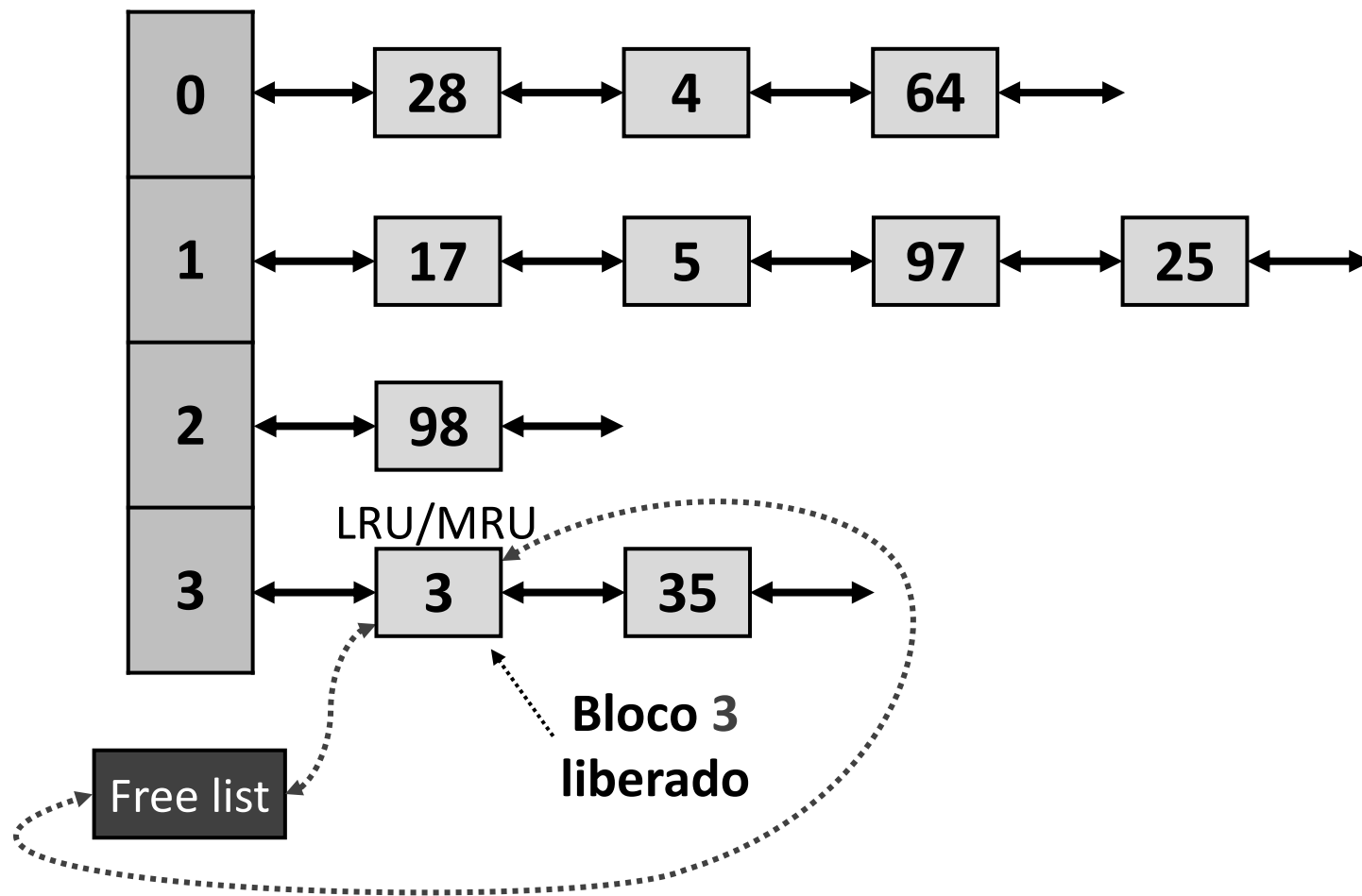


Free list

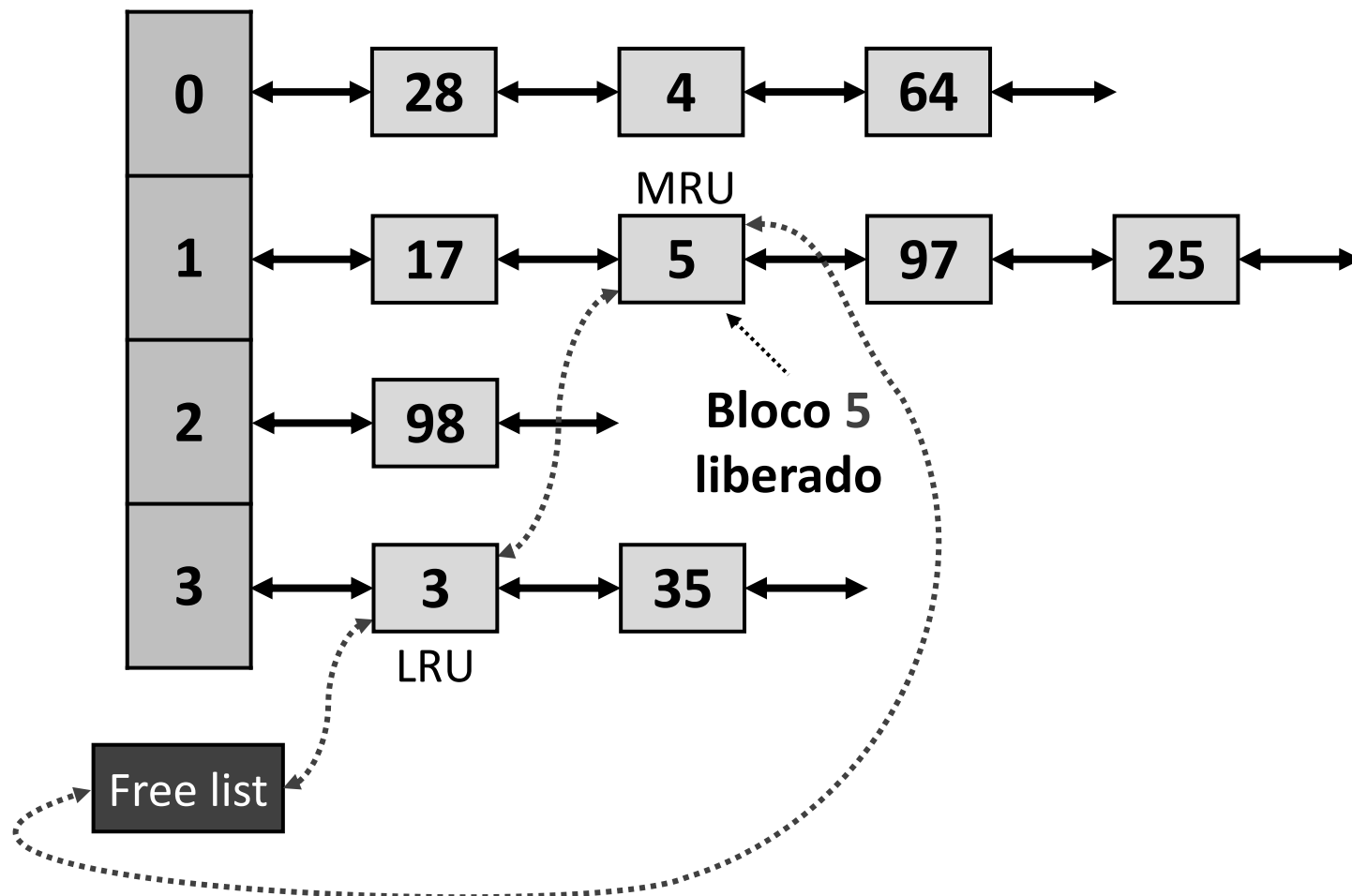
**Free list vazia:**  
Todos os buffers estão em uso no momento!



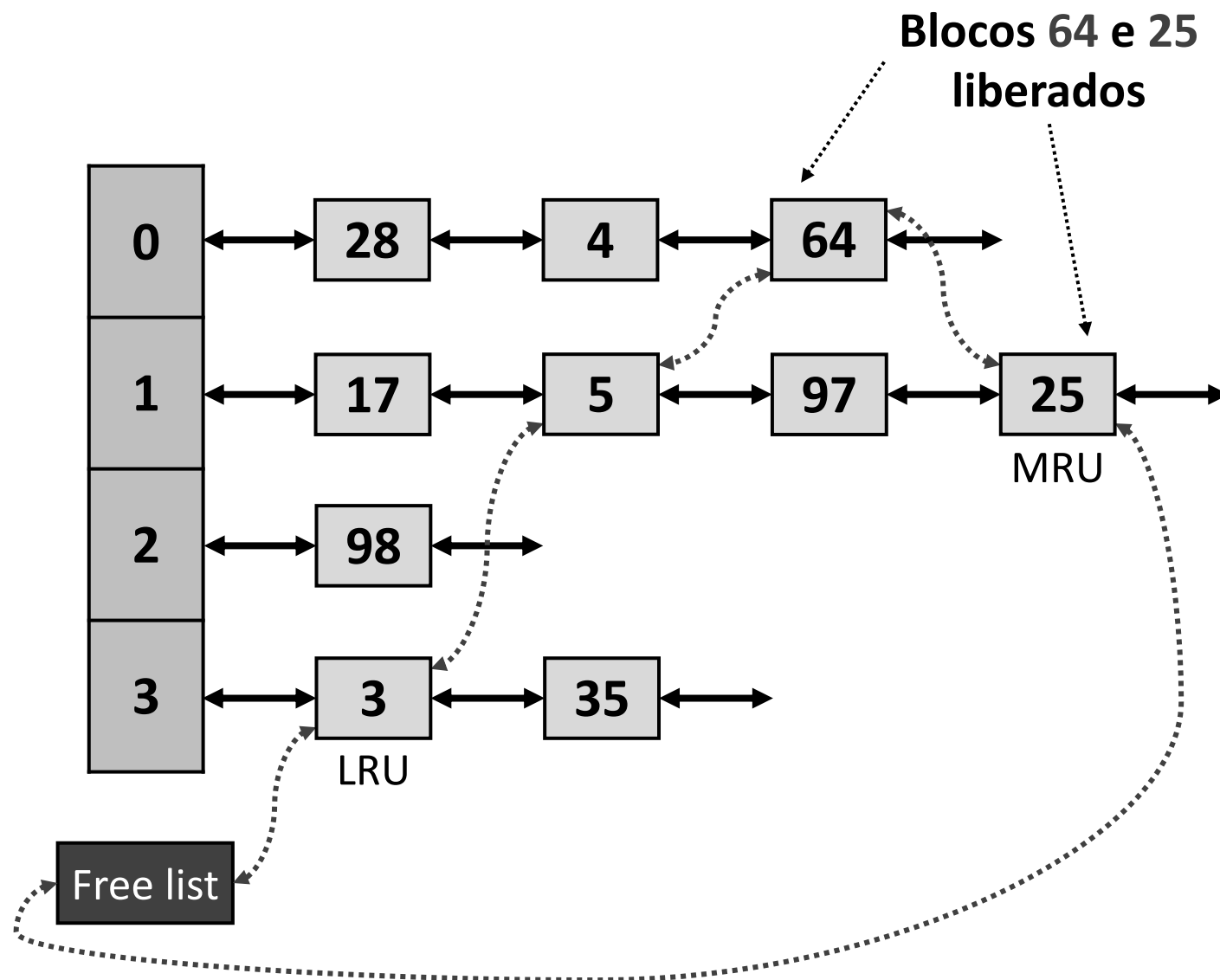
# Exemplo de liberação de buffers



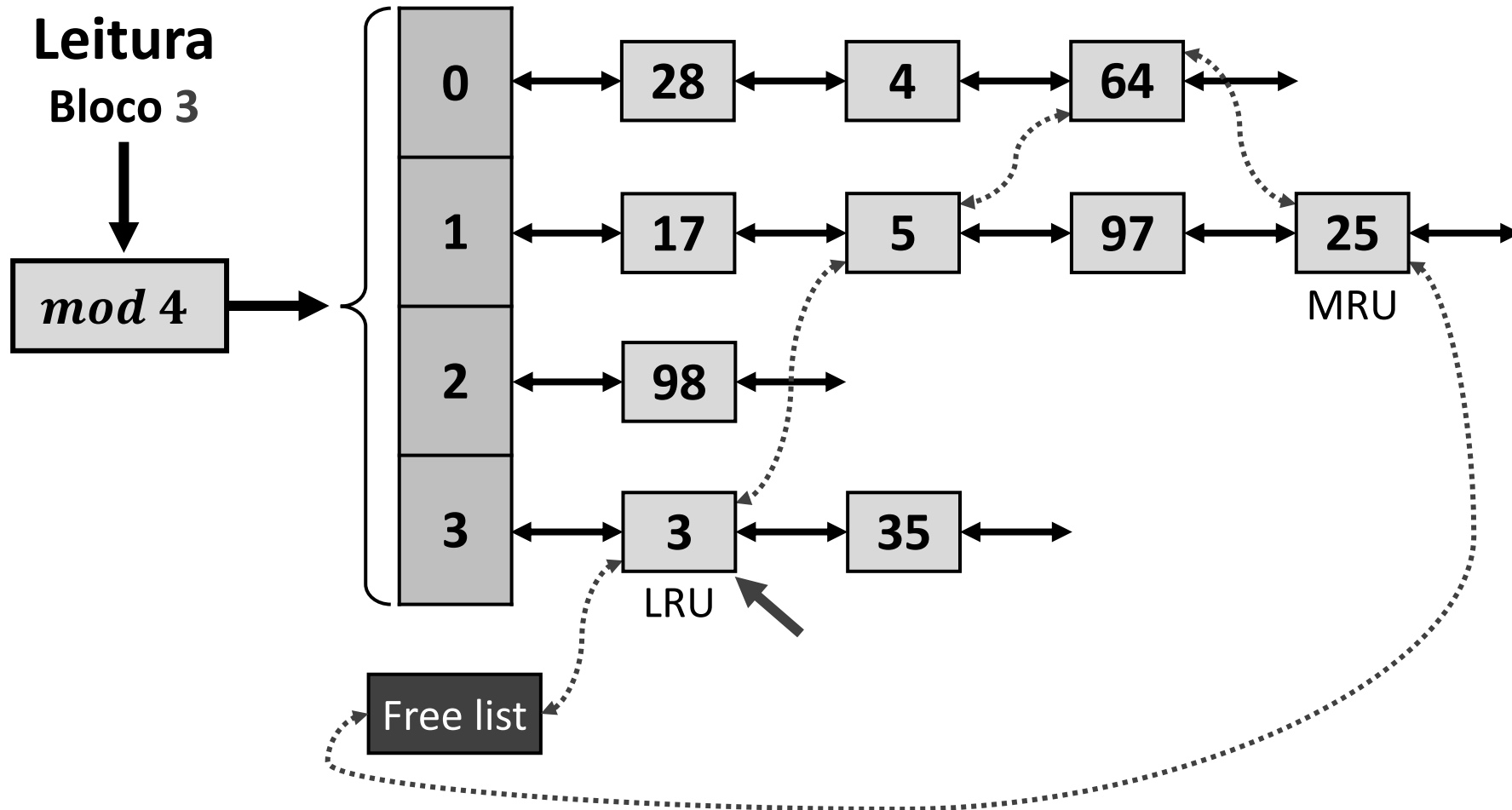
# Exemplo de liberação de buffers



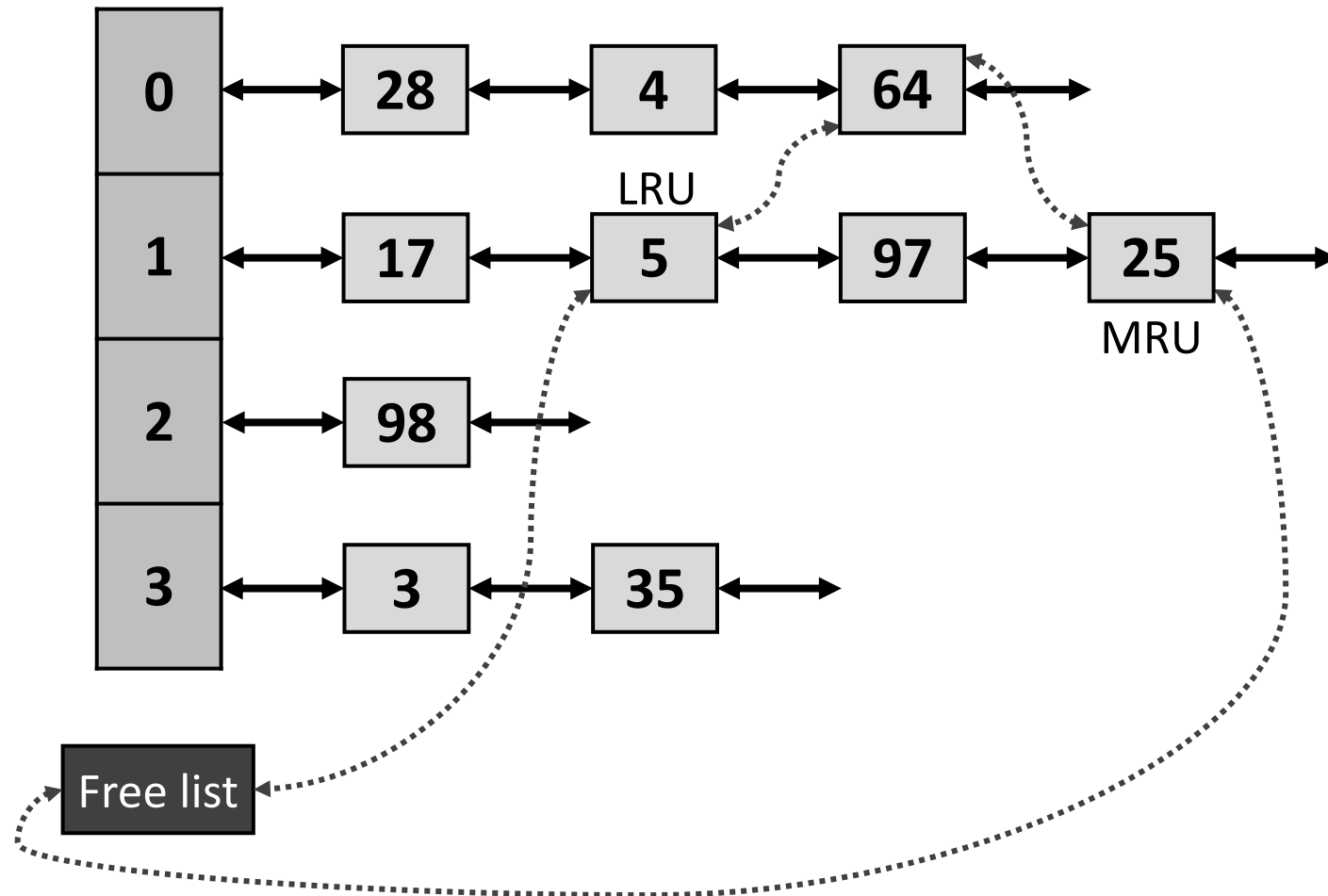
# Exemplo de liberação de buffers



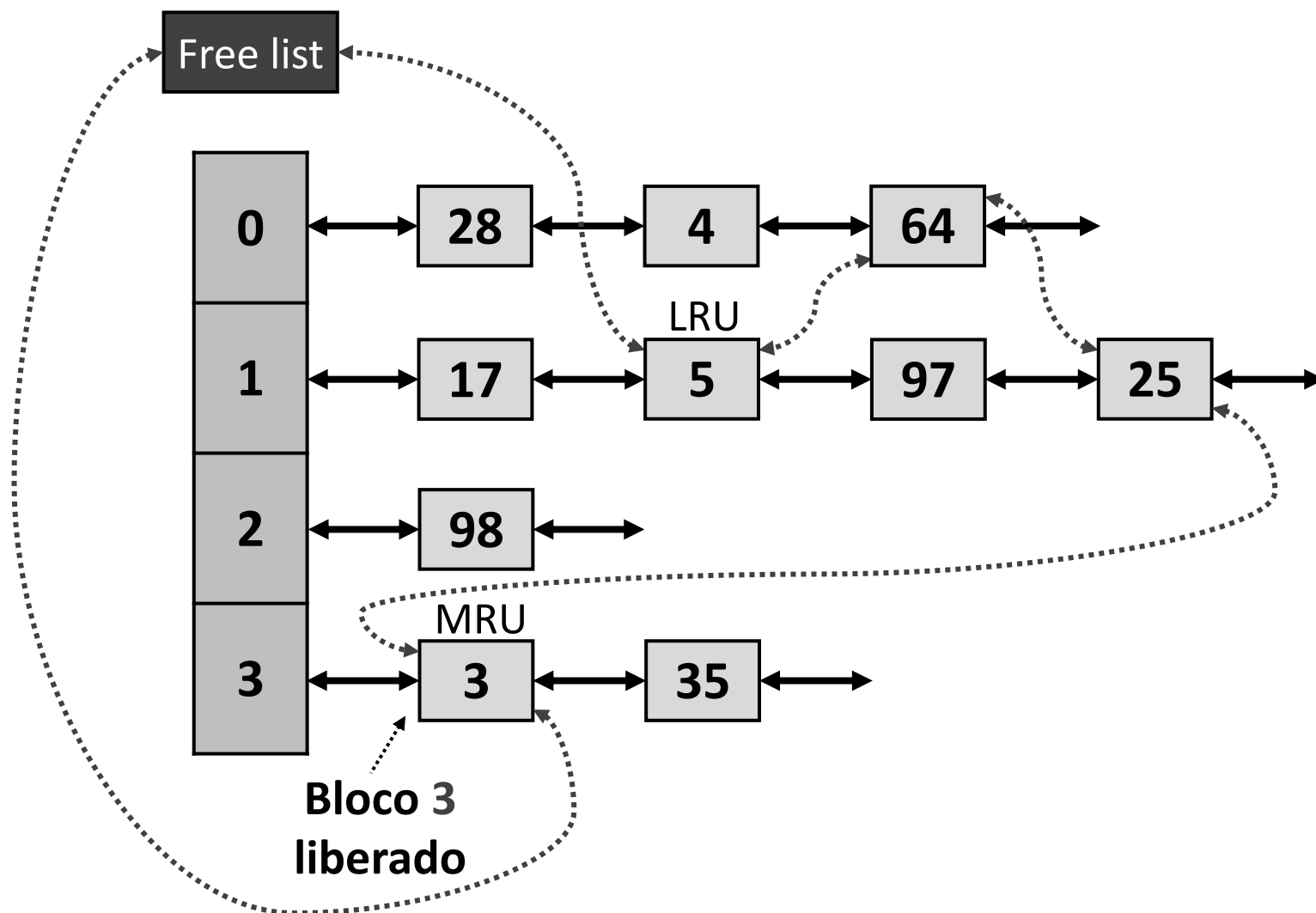
# Exemplo de reutilização de buffers



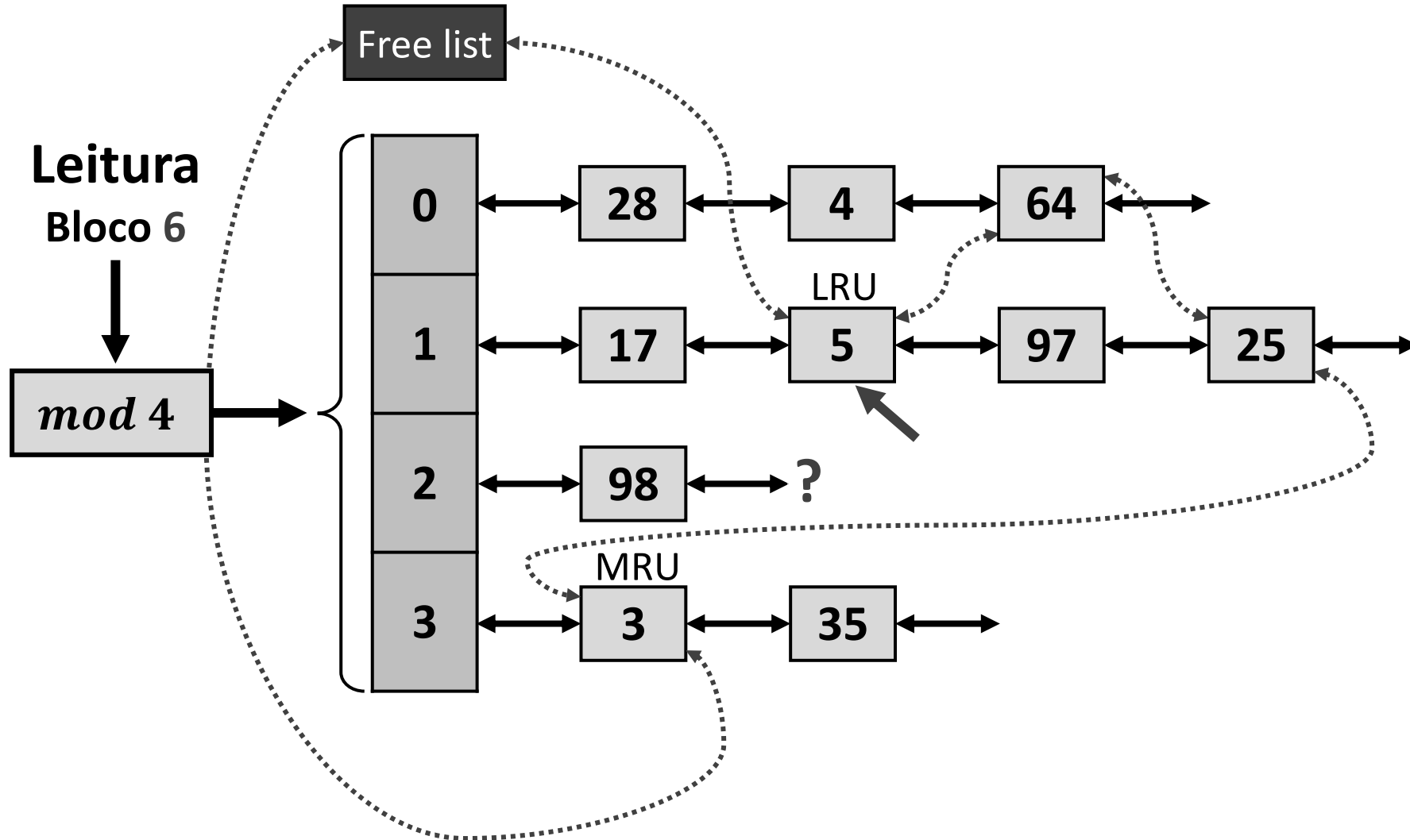
# Exemplo de reutilização de buffers



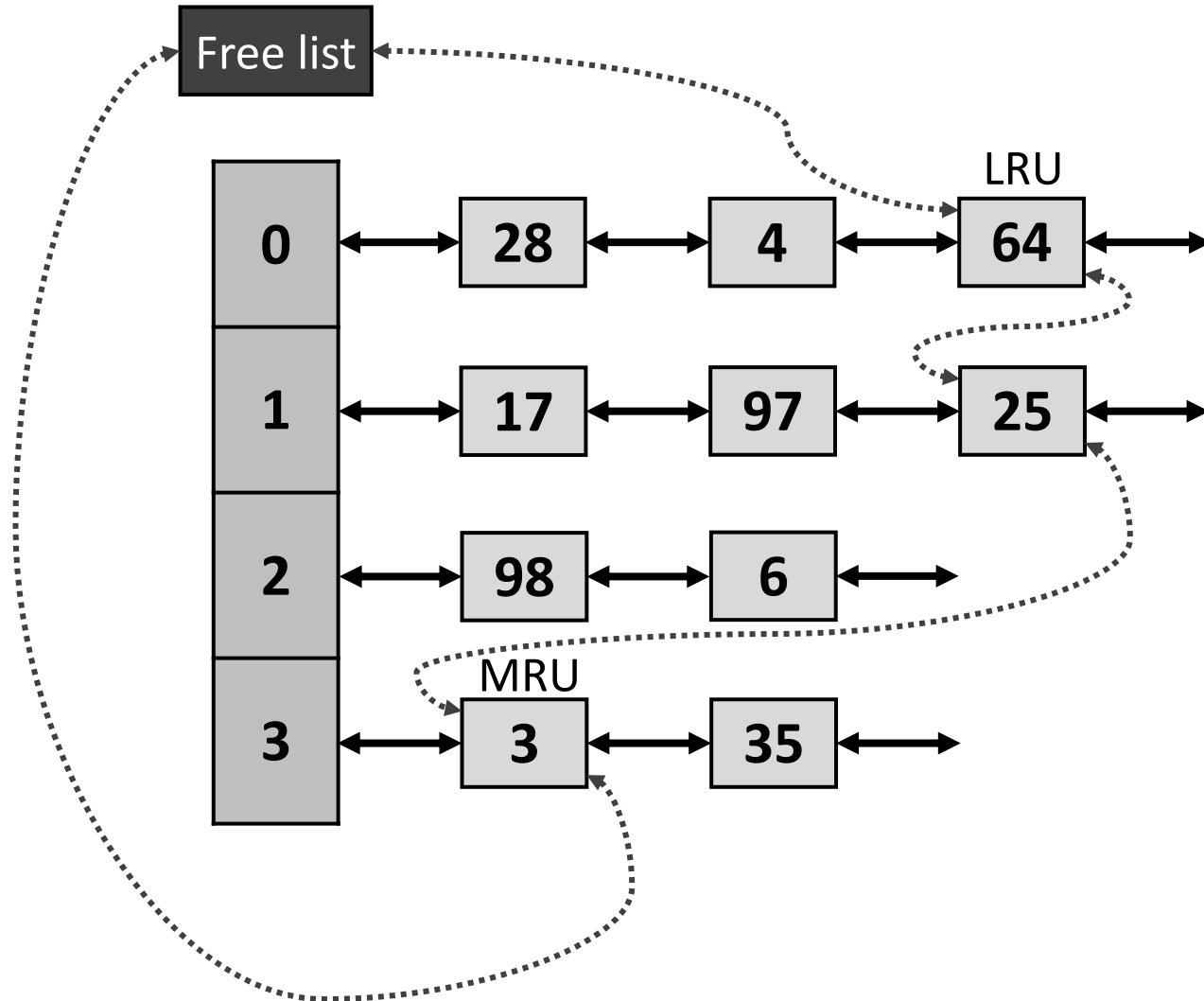
# Exemplo de reutilização de buffers



# Exemplo de substituição de blocos

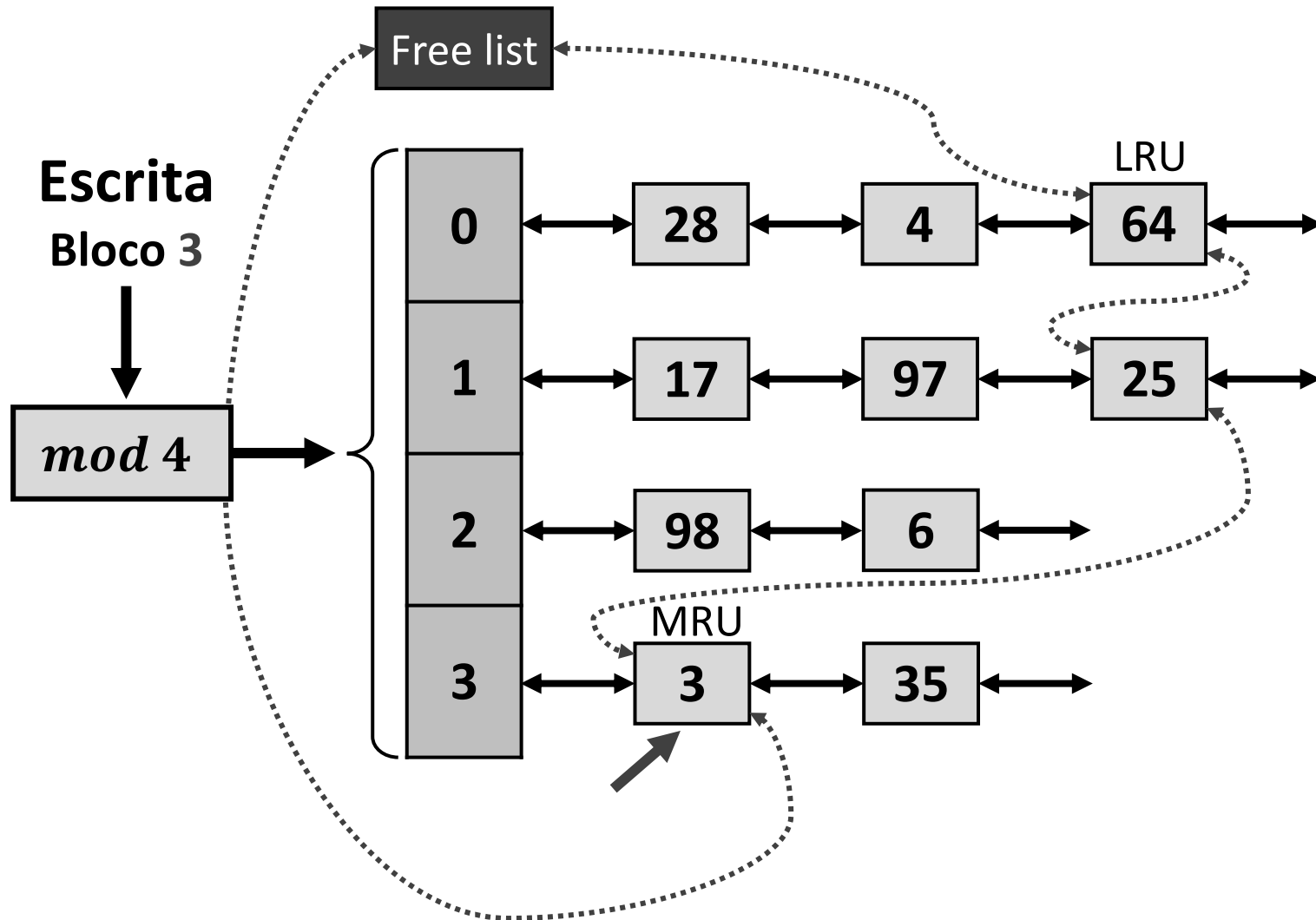


# Exemplo de substituição de blocos

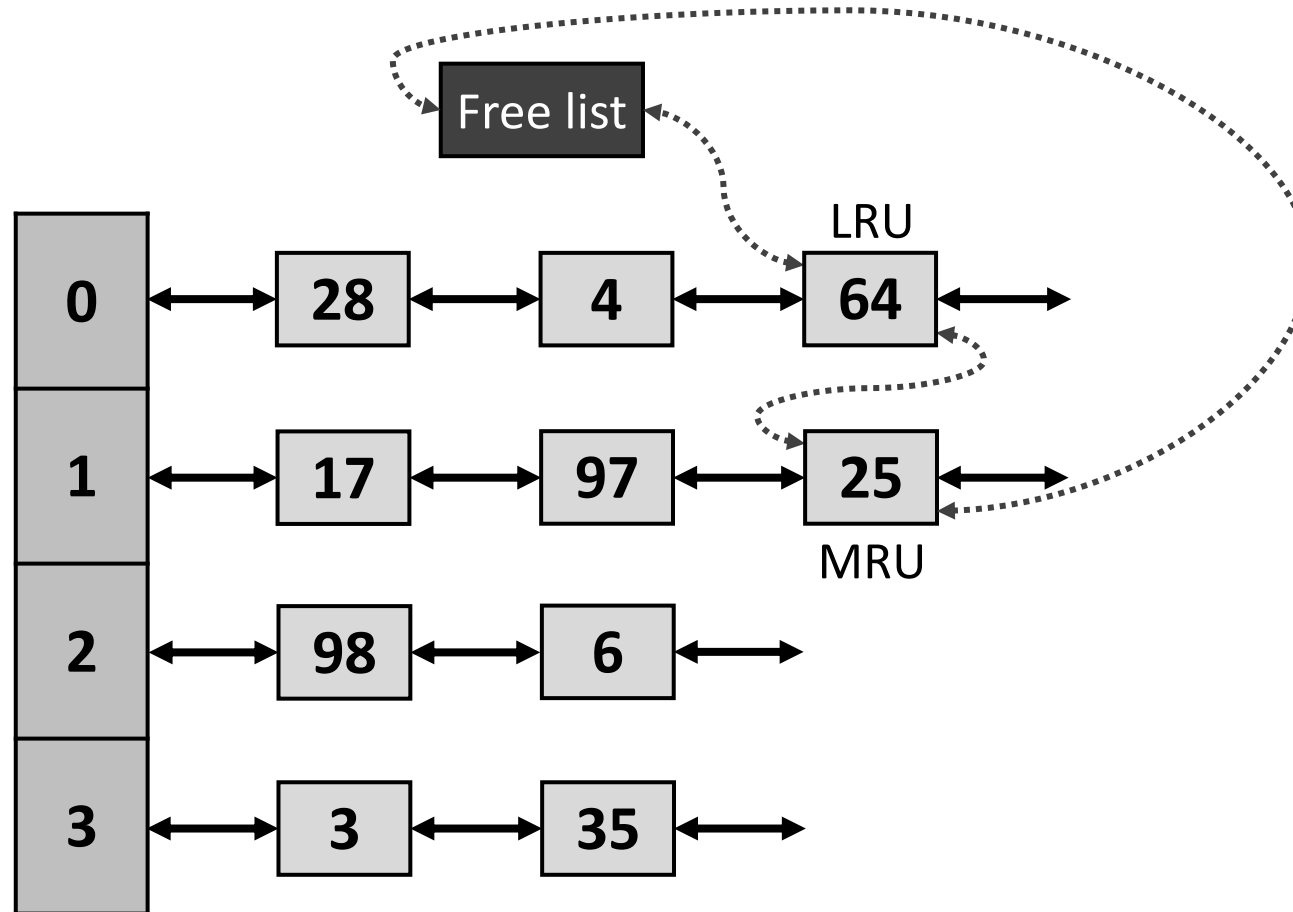




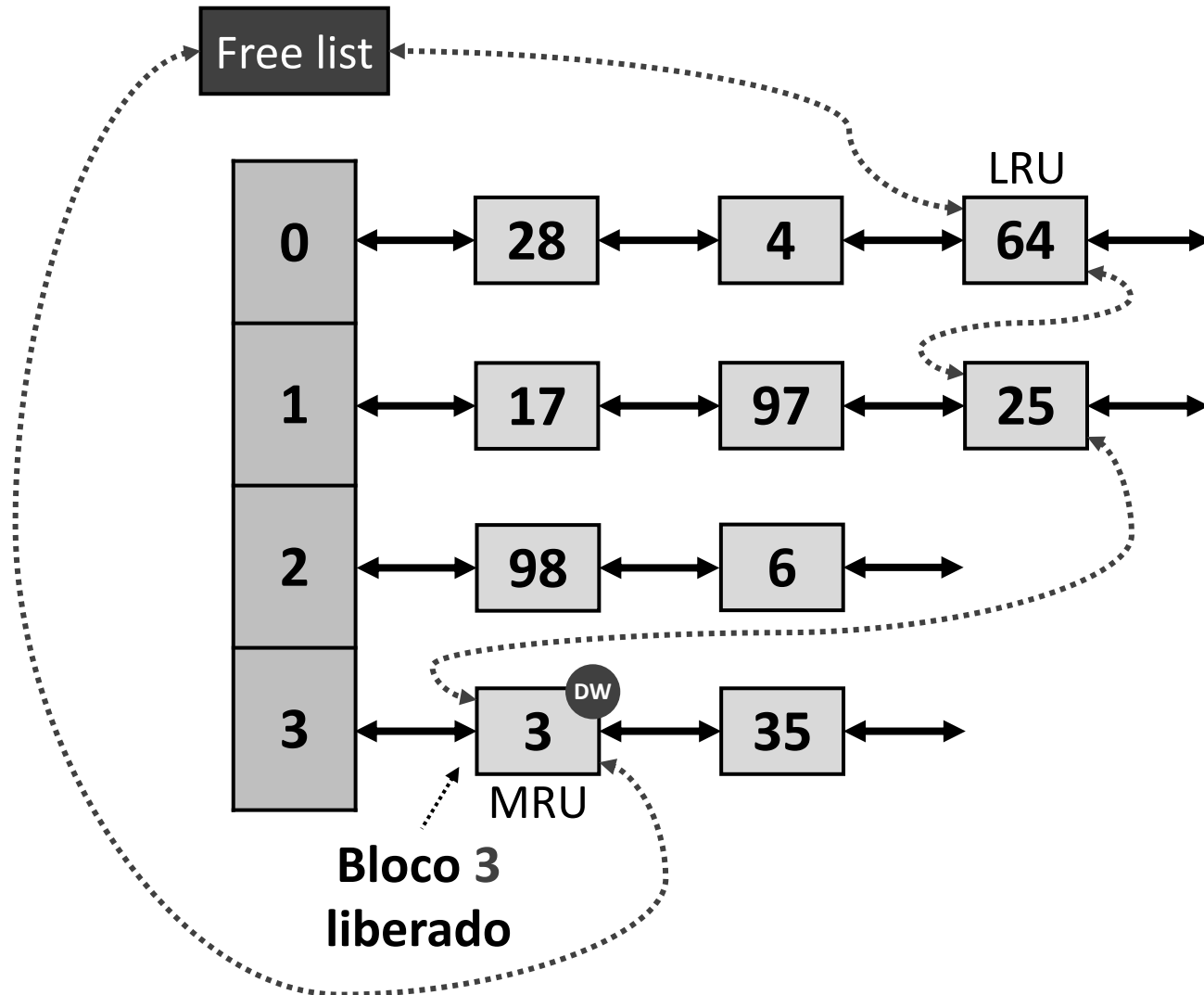
# Exemplo de escrita tardia de buffers



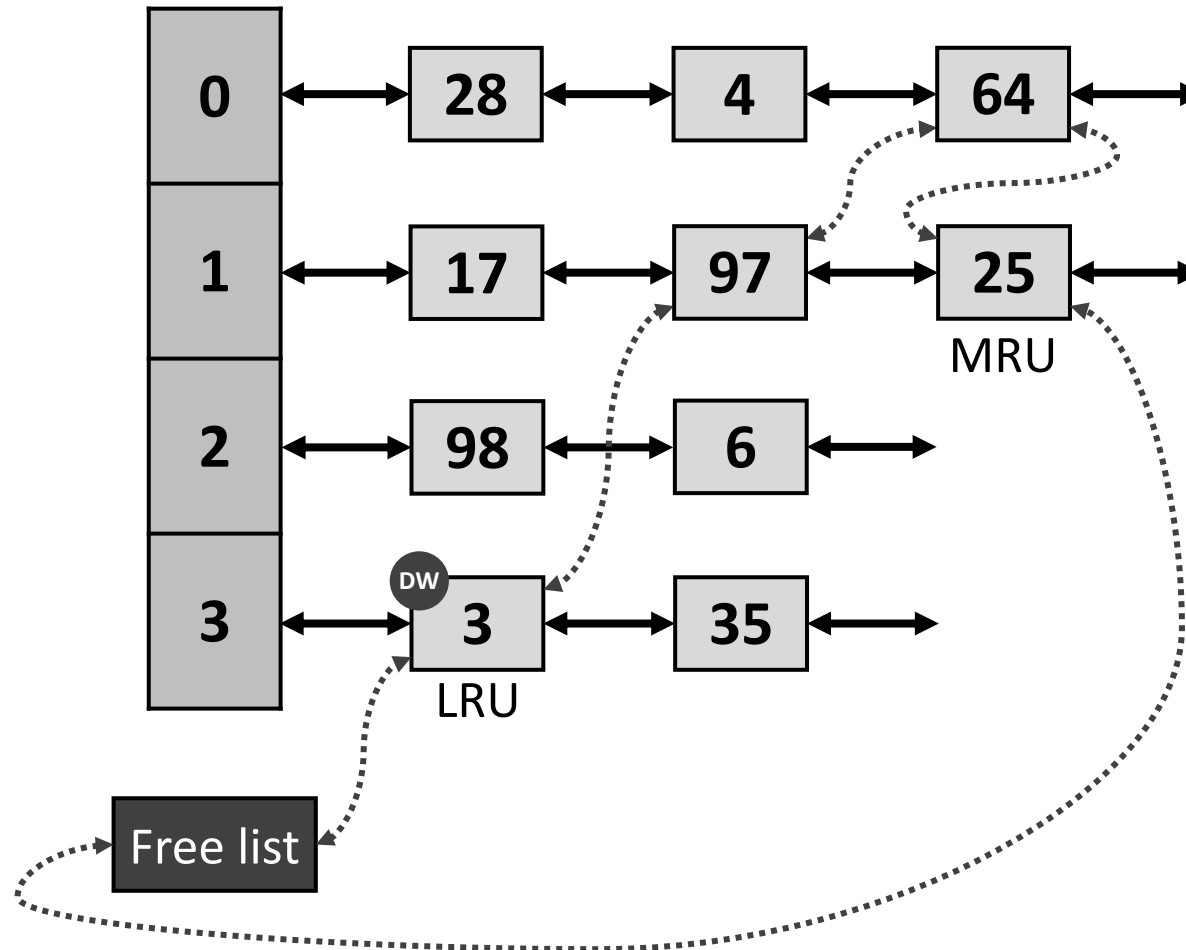
# Exemplo de escrita tardia de buffers



# Exemplo de escrita tardia de buffers



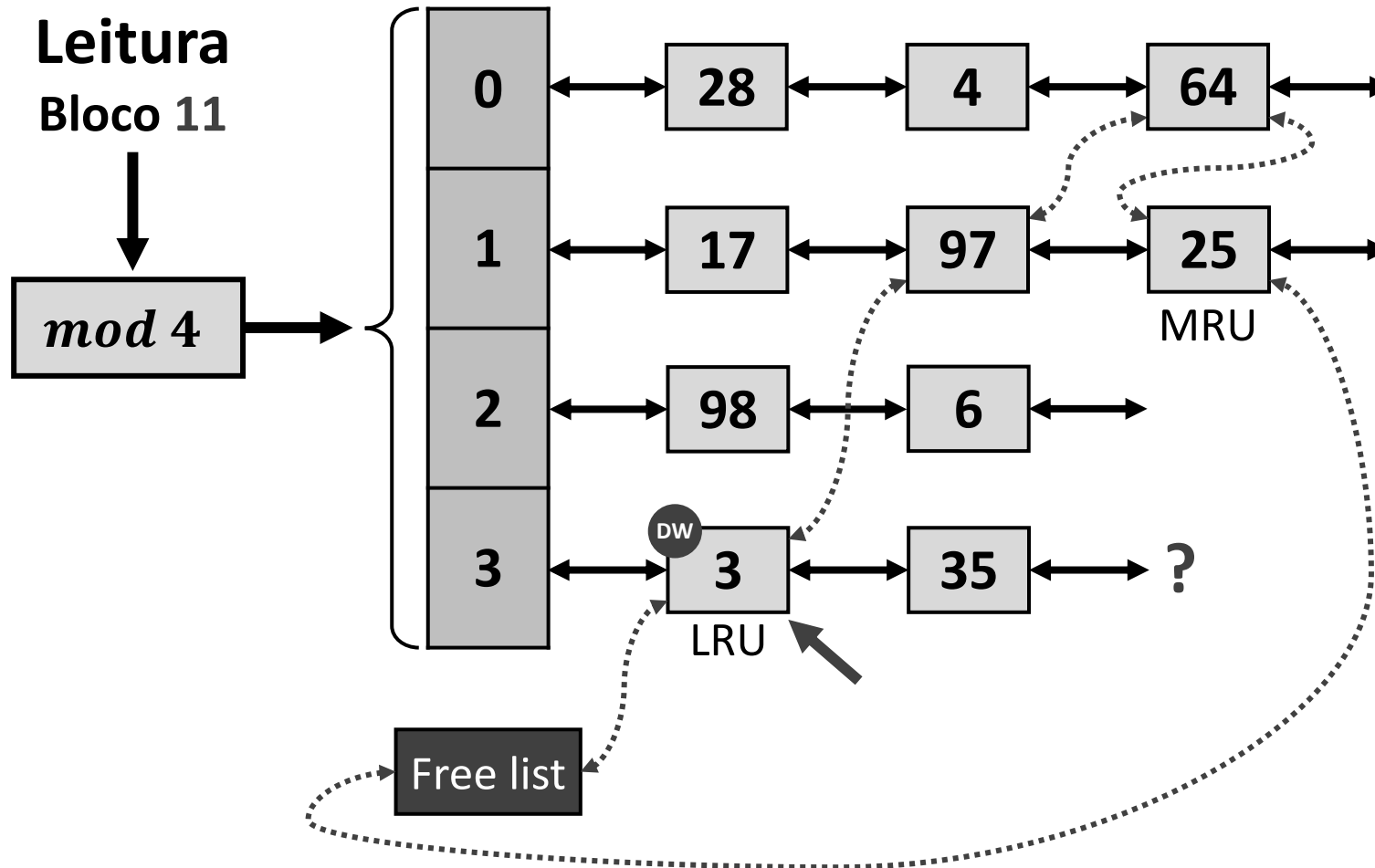
# Exemplo de escrita tardia de buffers



**Após os seguintes eventos:**

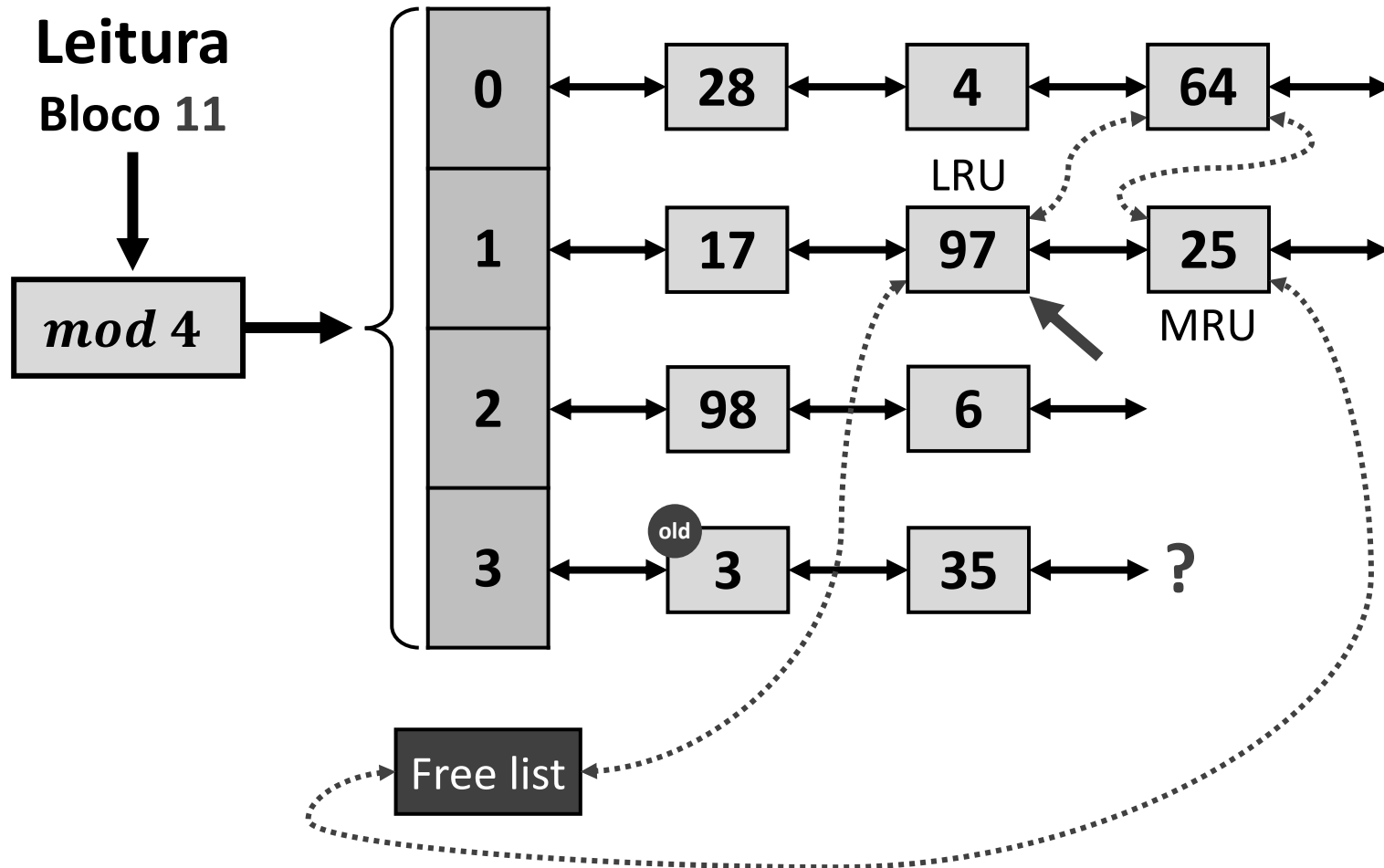
1. Liberação do buffer **97**
2. Leitura/liberação do buffer **64**
3. Leitura/liberação do buffer **25**

# Exemplo de escrita tardia de buffers

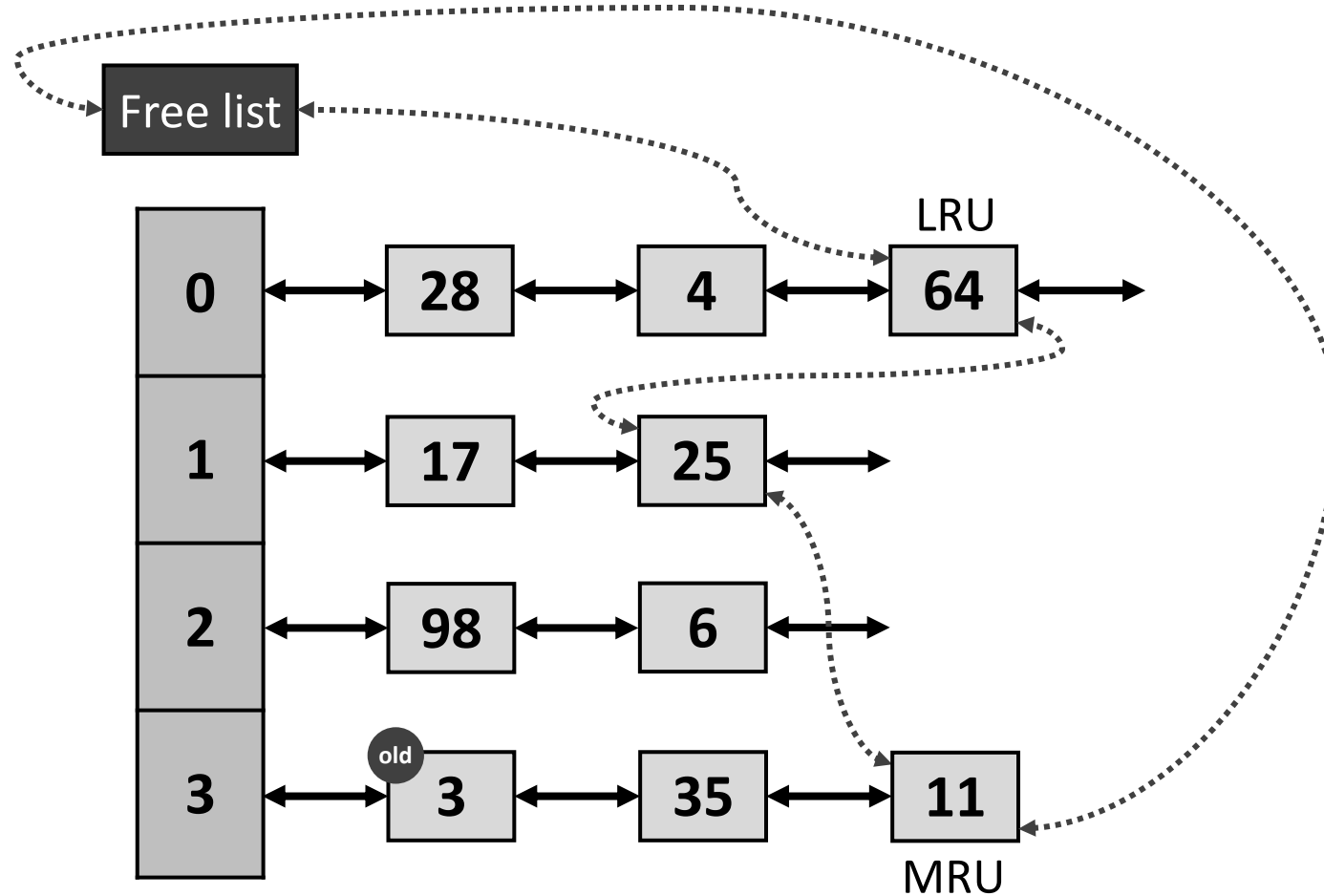


**Escrita tardia:**  
Requisição assíncrona!

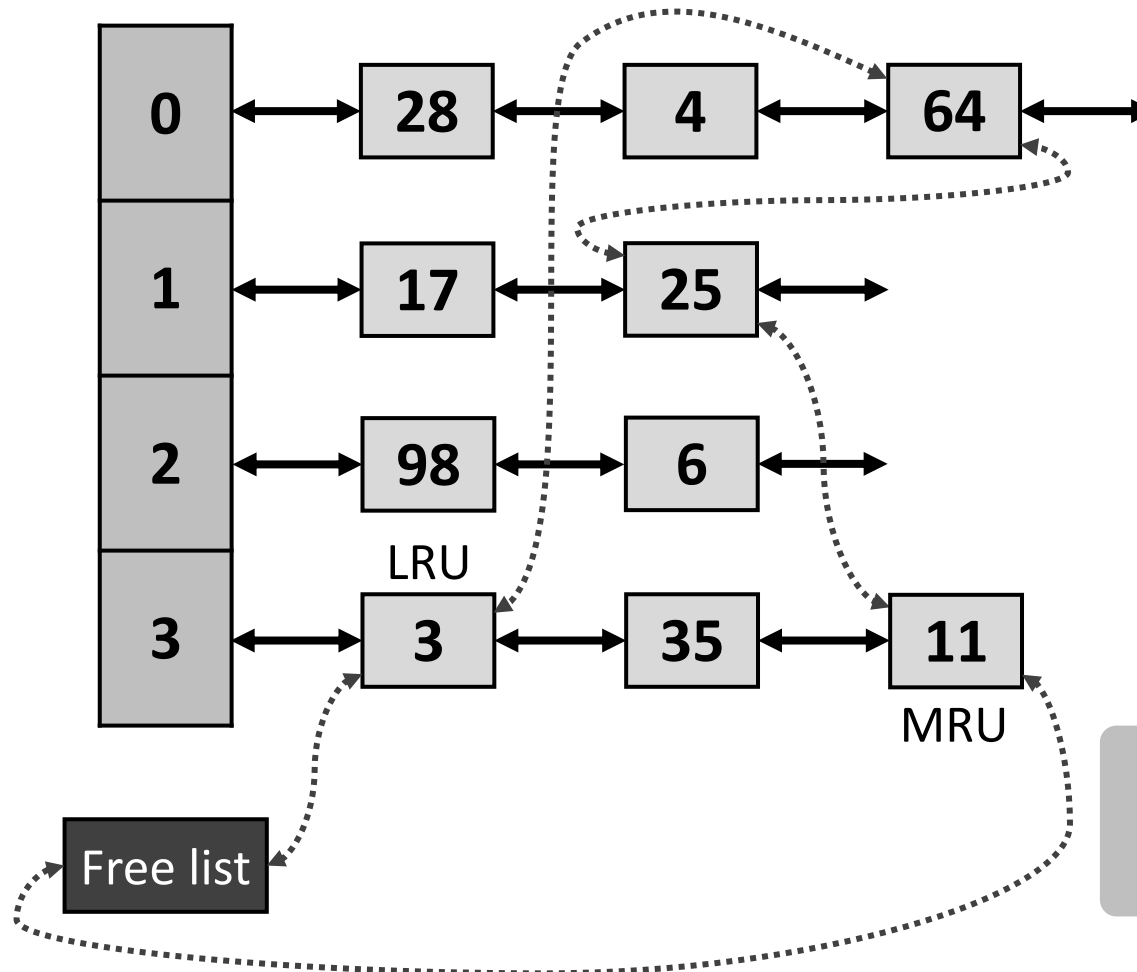
# Exemplo de escrita tardia de buffers



# Exemplo de escrita tardia de buffers



# Exemplo de escrita tardia de buffers



**Escrita do bloco 3 finalizada:  
Inserção na cabeça da free list!**



# Buffer cache

**bread ( )**: Retorna um buffer contendo os dados do bloco requisitado

```
bread() {  
    get buffer for block; ← getblk()  
  
    if (buffer valid)  
        return buffer;  
  
    initiate disk read into the buffer;  
    sleep(process, event: I/O is complete);  
    return buffer;  
}
```

# Buffer cache

**getblk() : Retorna um buffer da buffer cache**

```
getblk() {  
    while (buffer not found) {  
        if (block in hash queue) {  
            if (buffer locked) {  
                sleep(process, event: buffer becomes free);  
                continue;  
            }  
            lock buffer;  
            remove buffer from free list;  
            return buffer;  
        }  
        else {...}  
    }  
}
```

## `getblk()`: Retorna um buffer da buffer cache

```
} else {  
    if (there is no buffer on free list) {  
        sleep(process, event: any buffer becomes free);  
        continue;  
    }  
    lock buffer;  
    remove buffer from free list;  
    if (buffer marked as delayed write) {  
        asynchronous write buffer to disk;  
        continue;  
    }  
    remove buffer from old hash queue (if needed);  
    mark buffer as not valid;  
    put buffer in new hash queue (if needed);  
    return buffer;  
}  
}
```

# Buffer cache

**brelease() : Libera um buffer da buffer cache**

```
brelease() {  
    wakeup all procs: waiting for any buffer to be free;  
    wakeup all procs: waiting for this buffer to be free;  
  
    if (buffer valid and not old)  
        insert buffer at end of free list (MRU);  
    else  
        insert buffer at head of free list (LRU);  
  
    unlock buffer;  
}
```

# Buffer cache

**bwrite() : Escreve os dados do buffer em um bloco no disco**

```
bwrite() {  
    initiate disk write;  
    if (I/O sync)  
        sleep(process, event: I/O is complete);  
    release buffer; ← brelease()  
else {  
    if (buffer marked as delayed write)  
        mark buffer as old;  
    insert buffer at head of f.l. (LRU) when op. finished;  
}  
}
```

Feito pelo tratador de interrupção!

2

## Leitura antecipada de blocos

---

# Leitura antecipada de blocos

- Técnica que **trazer blocos** do disco para a buffer cache de maneira **assíncrona antes deles serem requisitados** pelos processos
  - O objetivo é **aumentar a taxa de hits** na buffer cache
- A leitura antecipada é útil **quando o padrão de acesso aos blocos é sequencial**
  - O processo requisita um bloco  $k$  à buffer cache
  - O sistema de arquivos escalona uma requisição do bloco  $k + 1$

# Leitura antecipada de blocos

## **abread ( ) : leitura antecipada de bloco**

```
abread() {  
  if (first block not in cache) { getblk()  
    get buffer for first block (b1);  
    initiate disk read into b1;  
  }  
  if (second block not in cache) { getblk()  
    get buffer for second block (b2);  
    initiate asynchronous disk read into b2;  
  }  
  sleep(process, event: I/O on b1 is complete);  
  return b1  
}
```



# Leitura antecipada de blocos

## ■ Problema

- A leitura antecipada pode piorar o desempenho do sistema se o acesso aos blocos **não for sequencial**

## ■ Solução

- Monitorar o **padrão de acesso** aos blocos
- **Desabilitar** a leitura antecipada de blocos se o padrão acesso **deixou de ser sequencial**
- **Habilitar** novamente quando o padrão de acesso **voltar a ser sequencial**

# ! Obrigado pela atenção!



**Dúvidas? Entre em contato:**

- [marcio.castro@ufsc.br](mailto:marcio.castro@ufsc.br)
- [www.marciocastro.com](http://www.marciocastro.com)

