

Capítulo 5

Organizando Arquivos para Desempenho



Outline (do capítulo)

1. Compressão de dados
2. Recuperando espaço em disco
3. Achando as coisas mais rapidamente:
 - uma introdução a ordenação interna e busca binária.
4. Ordenação de chaves (tag sort):
 - ordenação de arquivos grandes.



Recuperando Espaço em Disco: Motivação

- Suponha que num arquivo de registros de tamanho variável, um registro foi modificado de modo que o novo registro é maior do que o original.
- O que você faz com os dados extra?
 1. Colocar no final do arquivo e criar um ponteiro a partir do espaço do registro original para a extensão do registro, ou
 2. Poderia reescrever todo o registro no final do arquivo, deixando um “buraco” na posição do registro original.
- Cada solução tem suas desvantagens:
 - Na 1a, o trabalho de processar o registro é esquisito e mais lento do que era originalmente.
 - Na 2a, o arquivo desperdiça espaço.

● ● ● | Recuperando Espaço em Disco

- Nesta seção, veremos as maneiras como a organização do arquivo se deteriora a medida que ele vai sendo modificado. Tais modificações podem ser ocasionadas por:
 - adição de novos registros.
 - atualização de registros.
 - eliminação de registros.



Recuperando Espaço em Disco

- Adicionar registro (ao final do arquivo)
 - Esta operação não deteriora a organização do arquivo.
- As questões de manutenção se tornam mais complicadas quando:
 - Atualizamos registro de comprimento variável
 - O novo registro é menor (atualizar no mesmo lugar gera perda de espaço)
 - O novo registro é maior (é necessário eliminar o registro velho e incluir o novo)
 - Eliminamos um registro de comprimento fixo ou variável
 - a eliminação é LÓGICA.
 - deve-se poder reusar esse espaço.

● ● ● | Recuperando Espaço em Disco

Uma vez que a atualização de um registro pode ser vista como uma eliminação seguida de uma adição, vamos focar na questão da eliminação de um registro.

Depois que um registro é eliminado, desejamos poder reutilizar aquele espaço!!!



Recuperando Espaço em Disco: Eliminação e Compactação

- Estratégia de eliminação (fixo ou variável)
 - Qualquer que seja a estratégia, é necessário que o registro eliminado possa ser identificável.
 - Exemplos
 - Colocar um * ou um \$ num campo do registro, ou
 - Colocar um campo fixo no registro para indicar o status do registro.
- Estratégia de recuperação (fixo ou variável)
 - Deixar o registro eliminado por um tempo:
 - os programas devem ignorar os registros eliminados.
 - E depois, recuperar todos os espaços de uma só vez:
 - copiar os registros válidos para uma nova área de armazenamento em disco e devolver a área antiga, ou
 - compactar no mesmo lugar, lendo e regravando apenas os registros válidos.
 - Ex: arquivo de contabilidade compactado ao final do ano fiscal

● ● ● | Recuperando Espaço Dinamicamente: Eliminando e Reutilizando (Reg. Tam. Fixo)

- Eliminação
 - Marcar o registro eliminado com um *, por exemplo.
- Reutilizando o espaço ao inserir um novo reg.:
 - Varrer o arquivo seqüencialmente antes de adicionar um novo registro, procurando registro por registro, até que um registro eliminado seja encontrado. Se o programa atingir o final do arquivo e nenhum registro eliminado for encontrado, então o novo registro deve ser adicionado no final do arquivo
 - Processo muito lento!!

● ● ● | Recuperando Espaço Dinamicamente: Eliminando e Reutilizando (Reg. Tam. Fixo)

- Para reutilizar o espaço de um registro eliminado, nós precisamos:
 - uma maneira de saber, imediatamente, se existem espaços vazios no arquivo, e
 - uma maneira de pular diretamente para estes espaços, caso existam.



Recuperando Espaço Dinamicamente: Eliminando e Reutilizando (Reg. Tam. Fixo)

○ SOLUÇÃO:

- Utilizar uma lista encadeada contendo todos os registros eliminados.
- **Lista Encadeada:** estrutura de dados na qual cada elemento ou nó contém uma referência ao seu sucessor na lista.
- A maneira mais simples de manusear esta lista, é como uma pilha. Uma pilha é uma lista na qual todas as inserções e remoções dos nós acontece num dos finais da lista.
- Vamos referir a esta pilha, contendo os registros que se tornaram espaços disponíveis, como **PED** (pilha de espaços disponíveis).



Recuperando Espaço Dinamicamente: Eliminando e Reutilizando (Reg. Tam. Fixo)

- PED em arquivo com registro cabeçalho
 - O cabeçalho é o primeiro registro do arquivo, contendo:
 - o topo da PED
 - Contém o NRR do último registro eliminado.
 - Se $\text{topo}(\text{PED}) < 0$, então a pilha está vazia (-1 representa um ponteiro nulo).
 - Quando um registro é eliminado, ele é marcado como eliminado e inserido na PED (ou seja, terá um ponteiro para o registro eliminado antes dele). O espaço do registro está na mesma posição que antes, mas logicamente foi inserido na PED.
 - Implementa, no arquivo, uma lista encadeada dos registros eliminados.

Recuperando Espaço Dinamicamente: Eliminando e Reutilizando (Reg. Tam. Fixo)

Exemplo

Topo da PED -> 5

0	1	2	3	4	5	6
João ...	Pedro ...	Luiz ...	*-1	Paula ...	*3	Rui ...

Topo da PED -> 1

0	1	2	3	4	5	6
João ...	*5	Luiz ...	*-1	Paula ...	*3	Rui ...

Topo da PED -> -1

0	1	2	3	4	5	6
João ...	1º novo reg	Luiz ...	3º novo reg	Paula ...	2º novo reg	Rui ! ² . .



Recuperando Espaço em Disco: Eliminando e Reutilizando (Reg. Tam. Var.)

- Para dar suporte a reutilização de registros através de uma lista de espaço disponível (LED), nós precisamos de:
 - Uma maneira interligar os registros eliminados na LED (ou seja, um lugar para colocar os apontadores).
 - Um algoritmo para incluir na LED novos registros eliminados.
 - Um algoritmo para achar e remover da LED os espaços disponíveis quando se vai reutilizá-los.

Recuperando Espaço em Disco: Eliminando e Reutilizando (Reg. Tam. Var.)

- LED e espaços livres

- A cabeça/topo da LED fica no registro cabeçalho.

- Estrutura dos registros normais (com reuso):

<tamanho registro> | <c₁> | <c₂> | ... | <c_n> |

onde: indica a fragmentação interna do registro (espaço perdido no registro, menor que o original liberado).

- Registro eliminado:

<tamanho registro> * <ponteiro>

onde: * indica que o registro está livre (foi eliminado logicamente).

<ponteiro> é binário e aponta para o primeiro byte do próximo registro livre.

..... é o espaço restante do registro livre.

14

Não podemos usar o
NRR como ponteiro!!!

● ● ● | Recuperando Espaço em Disco: Eliminando e Reutilizando (Reg. Tam. Var.)

○ Exemplo

Topo da LED: -1

```
40 Ames|John|123 Maple|Stillwater|OK|74075|64 Morrison|Sebastian  
|9035 South Hillcrest|Forest Village|OK|74820|45 Brown|Martha|62  
5 Kimbark|Des Moines|IA|50311|
```



Após a eliminação
do 2o registro.

Topo da LED: 43

```
40 Ames|John|123 Maple|Stillwater|OK|74075|64 *| -1.....  
.....45 Brown|Martha|62  
5 Kimbark|Des Moines|IA|50311|
```



Recuperando Espaço em Disco: Eliminando e Reutilizando (Reg. Tam. Var.)

- Incluir e Remover Registros da LED:
 - Na inclusão, trata-se a LED como uma PED:
 - adiciona-se o registro livre na cabeça/topo da LED.
 - Na remoção, trata-se a LED como uma lista:
 - busca-se o primeiro espaço na LED, tal que
 - $|\text{reg-novo}| \leq |\text{espaço}|$
 - Pode-se pesquisar a LED inteira, sem achar o *espaço adequado*.
 - Se tal espaço existe, então ele é removido da LED e reusado.
 - Se não o reg-novo é colocado no fim do arquivo.

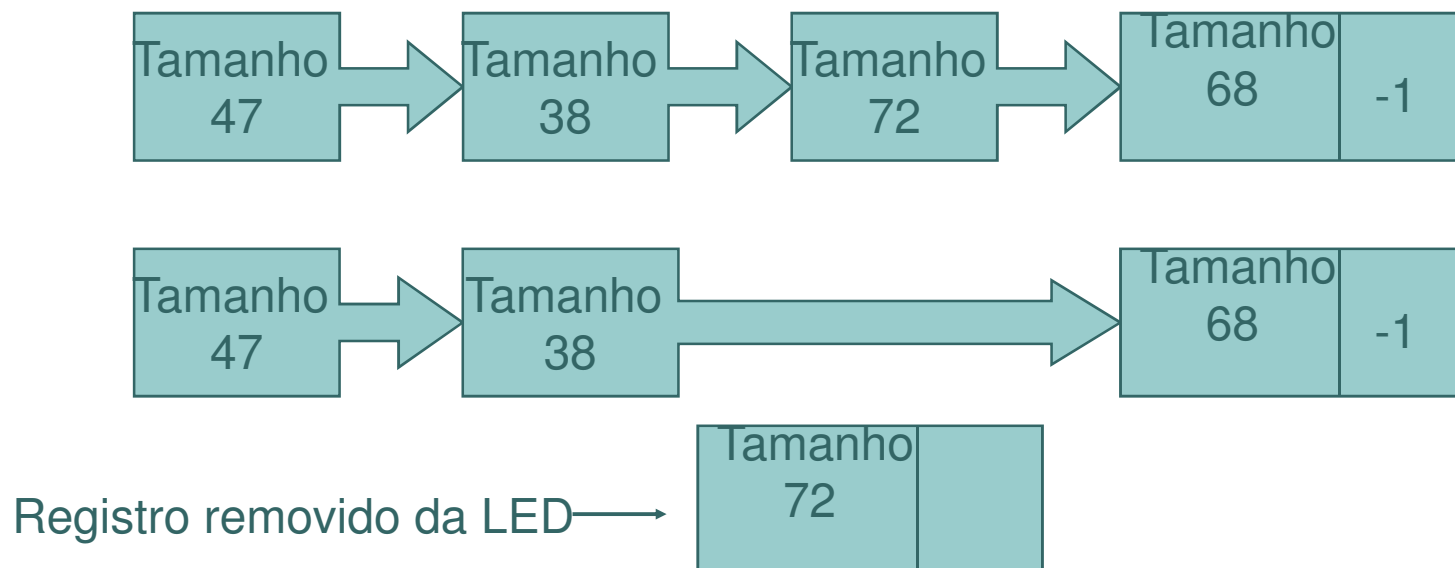
- Essa estratégia pode levar a uma alta fragmentação interna dos registros.

Fragmentação interna é a perda de espaço ao nível de cada registro (ocorre por que o espaço do registro não é totalmente utilizado).

Recuperando Espaço em Disco: Eliminando e Reutilizando (Reg. Tam. Var.)

Exemplo

- Remoção de um registro da LED para armazenar um novo registro que requer 55 bytes de espaço.



Recuperando Espaço em Disco: Fragmentação Interna

- Em registros de tamanho fixo:

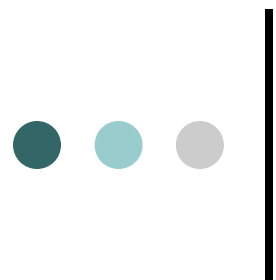
- Ex: registros de tamanho fixo de 64 bytes

```
Ames|John|123 Maple|Stillwater|OK|74075|.....  
Morrison|Sebastian|9035 South Hillcrest|Forest Village|OK|74820|  
Brown|Martha|625 Kimbark|Des Moines|IA|50311|.....
```

Fragmentação
Interna

- Alternativas:

- permitir que os campos possam variar
 - provoca perdas no final do registro.
- dimensionar cada campo pelo maior tamanho da informação
 - provoca perdas ao nível de campo.
- ambas provocam fragmentação interna.



Recuperando Espaço em Disco: Fragmentação Interna

- Em registros de tamanho variável:

- Ex: registros de tamanho variável com indicador de tamanho

```
40 Ames|John|123 Maple|Stillwater|OK|74075|64 Morrison|Sebastian  
|9035 South Hillcrest|Forest Village|OK|74820|45 Brown|Martha|62  
5 Kimbark|Des Moines|IA|50311|
```

- A fragmentação interna é eliminada na criação dos registros:
 - usa-se só o espaço requerido.
- Nos casos de eliminação de registro e reuso de espaços disponíveis por registros menores do que os originais:
 - volta o problema da **fragmentação interna**.

● ● ● | Recuperando Espaço em Disco: Fragmentação Interna

- Ilustração: Fragmentação em registros de tamanho variável

Topo da LED: 43

40 Ames|John|123 Maple|Stillwater|OK|74075|64 *| -1.....
.....45 Brown|Martha|62
5 Kimbark|Des Moines|IA|50311|



Após a ocupação do espaço do
2o registro por um novo registro
menor que o anterior.

Topo da LED: -1


40 Ames|John|123 Maple|Stillwater|OK|74075|64 Ham|Al|28 Elm|Ada|
OK|70332|.....45 Brown|Martha|62
5 Kimbark|Des Moines|IA|50311|

 **Fragmentação Interna**²⁰

● ● ● | Recuperando Espaço em Disco: Fragmentação Interna

- Solução: LED com reuso e com liberação de espaço
 - Pega-se da LED um espaço disponível:
 - ocupa-se a parte final do espaço com o novo registro.
 - altera-se, na LED, o tamanho do espaço inicial.
 - Essa entrada é mantida na LED para uso futuro.
 - Exceto pelo tamanho desse espaço, a LED não é alterada.

Topo da LED: 43



```
40 Ames|John|123 Maple|Stillwater|OK|74075|35 *| -1.....
.....26 Ham|Al|28 Elm|Ada|OK|70332|45 Brown|Martha|6
25 Kimbark|Des Moines|IA|50311|
```

Recuperando Espaço em Disco: Fragmentação Interna

- A parte que fica na LED pode ser muito pequena:
 - se tal espaço não puder ser usado por nenhum outro registro, caracteriza a **fragmentação externa**.

Topo da LED: 43

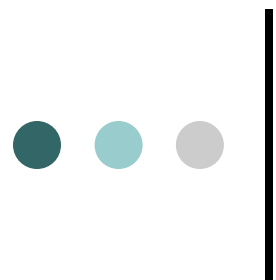
40 Ames|John|123 Maple|Stillwater|OK|74075|35 *| -1.....
.....26 Ham|Al|28 Elm|Ada|OK|70332|45 Brown|Martha|6
25 Kimbark|Des Moines|IA|50311|



Após a ocupação de mais um registro
de 25 bytes na área disponível na LED.

Topo da LED: 43

40 Ames|John|123 Maple|Stillwater|OK|74075|8 *| -1...25 Lee|Ed|R
t 2|Ada|OK|74820|26 Ham|Al|28 Elm|Ada|OK|70332|45 Brown|Martha|6
25 Kimbark|Des Moines|IA|50311|



Recuperando Espaço em Disco: Fragmentação Externa

- Meios de combater a Fragmentação Externa
 1. Gerar um novo arquivo (quando a fragmentação ficar intolerável)
 - eliminando os espaços dos registros eliminados.
 - devolvendo a área do arquivo antigo ao sistema.
 2. Concatenar espaços adjacentes na LED
 - Combiná-los para fazer deles um único registro maior.
 3. Tentar minimizar a fragmentação antes que ela ocorra, adotando uma das seguintes estratégias de reuso:
 - primeiro ajuste.
 - melhor ajuste.
 - pior ajuste.

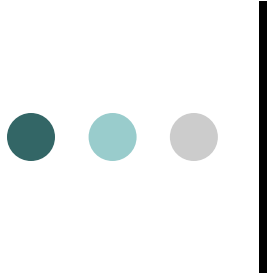
● ● ● | Recuperando Espaço em Disco: Estratégias de Reuso: Primeiro Ajuste

- A LED não está ordenada:
 - os espaços gerados pela eliminação de registros são incluídos na cabeça/topo da lista.
- Quando se quer introduzir um registro novo:
 - a LED é percorrida até que $|\text{reg}| \leq |\text{espaço}_i|$ ou que o fim da lista seja atingido.
 - se um espaço_i foi encontrado, grave o registro nele e libere o espaço restante.
 - Se não, inclua o registro no fim do arquivo.



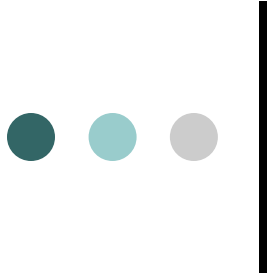
Recuperando Espaço em Disco: Estratégias de Reuso: Primeiro Ajuste

- Vantagens:
 - a inclusão dos registros eliminados é rápida.
 - realizada pela cabeça da LED.
- Desvantagens
 - A cada inclusão de um novo registro no arquivo a LED precisa ser percorrida.
 - as vezes sem sucesso!
 - Não importa o tamanho do espaço disponível, desde que ele comporte o tamanho do novo registro.
 - pode-se ter uma sobra pequena demais ou muito grande!



Recuperando Espaço em Disco: Estratégias de Reuso: Melhor Ajuste

- A LED aqui está ordenada em ordem crescente de tamanho dos espaços.
 - Os espaços gerados são incluídos na posição correta da LED.
 - esse esforço pode ser significativo!
- Na inclusão de um novo registro:
 - percorre-se a LED até que $|\text{registro}| \leq |\text{espaço}_i|$
 - grava-se o registro em espaço_i
 - essa sobra pode ser pequena demais e não ser reusável por nenhum outro registro.
 - gerando fragmentação
 - Obs: Aqui o espaço encontrado na LED é o menor espaço possível disponível para o novo registro.



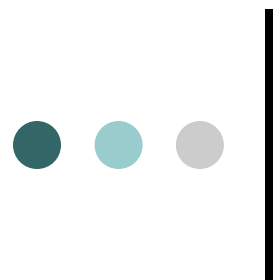
Recuperando Espaço em Disco: Estratégias de Reuso: Melhor Ajuste

○ Vantagens

- Como esse é o menor espaço existente na LED que suporta o registro novo, o desperdício é o menor possível.
 - mesmo que ele gere fragmentação externa.

○ Desvantagem

- Os primeiros elementos da LED, de tão pequenos, podem não ser aproveitáveis:
 - aumentando o tamanho/tempo da busca até se encontrar o espaço adequado.
- A inclusão dos espaços na LED pode ser lenta:
 - requerendo um tempo significativo.



Recuperando Espaço em Disco: Estratégias de Reuso: Pior Ajuste

- A LED aqui está ordenada em ordem decrescente de tamanho dos espaços:
 - os espaços gerados são incluídos na posição correta da LED.
 - o que gera um esforço significativo!
- Na inclusão de um novo registro no arquivo:
 - usa-se o registro no topo da LED (vista como pilha)
 - grava-se o registro nesse espaço:
 - essa sobra é a maior possível.
 - aumentando as chances de sua reutilização!



Recuperando Espaço em Disco: Estratégias de Reuso: Pior Ajuste

- Vantagens

- A busca pelo espaço na LED é rápida:
 - pega-se o primeiro elemento da LED.
 - Se o primeiro elemento da LED não for grande o bastante para o novo registro, nenhum dos outros elementos serão.
- Diminui a chance de fragmentação externa.
 - os espaços gerados pela reutilização são os maiores possíveis.

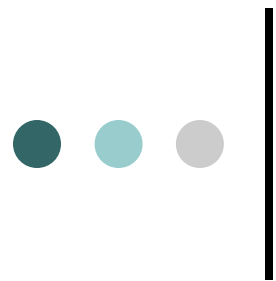
- Desvantagens

- A inclusão do espaço na LED ordenada pelo tamanho é custosa.



Recuperando Espaço em Disco: Estratégias de Reuso: Conclusão

- A estratégia de reuso somente se aplica a:
 - arquivos voláteis.
 - arquivos de registro com comprimento variável.
- Se o espaço é perdido por fragmentação interna (ou seja, estão sobrando muitos espaços dentro dos registros):
 - a escolha recai sobre uma das estratégias:
 - primeiro ajuste.
 - melhor ajuste.
- Se a perda é por fragmentação externa (ou seja, estão sobrando muitos espaços que não utilizáveis)
 - então considere a estratégia do “pior ajuste”.



Próxima Aula

- Recuperação rápida de arquivos
 - Ordenação por Chaves - *Keysorting*