

Compactação e Reuso de Espaço

Cristina D. A. Ciferri

Thiago A. S. Pardo

Leandro C. Cintra

M.C.F. de Oliveira

Moacir Ponti Jr.



Manipulação de Dados

- Operações básicas
 - **adição** de registros
 - relativamente simples
 - **remoção** de registros
 - **atualização** de registros
 - eliminação e adição de um registro

Quando um registro é removido, deve-se posteriormente reutilizar o espaço do registro



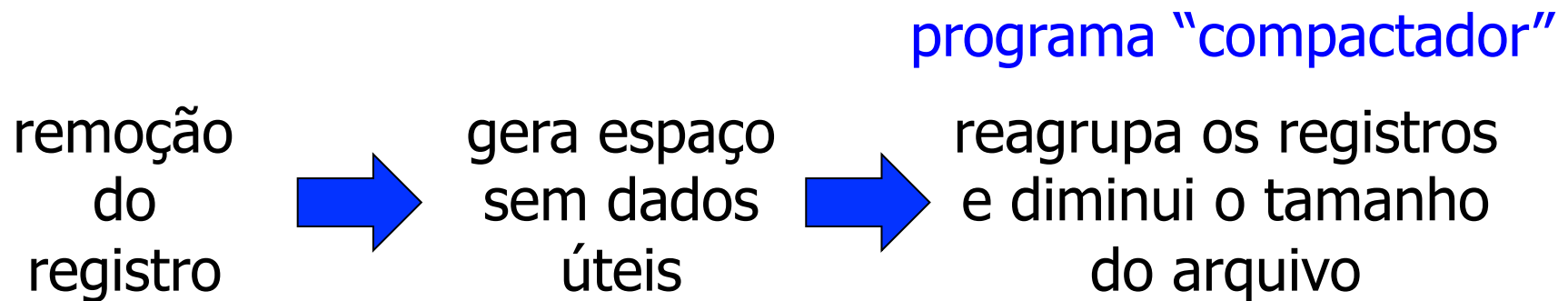
Compactação e Reuso

- Compactação
 - busca por regiões do arquivo que não contêm dados
 - recupera os espaços perdidos
- Reuso
 - insere dados nos espaços perdidos

Abordagem Estática versus
Abordagem Dinâmica



Abordagem Estática



- Técnica para reconhecer registros removidos: remoção lógica
 - atribuir um valor para um campo do registro
 - usar um campo extra



Abordagem Estática

- Não faz nada em um intervalo de tempo Δt
- Durante Δt

remoção
lógica

- registros removidos são marcados, porém não são reaproveitados
- novas inserções são realizadas no final do arquivo
- buscas desconsideram os registros marcados como removidos

- Após Δt

remoção
física

- programa é executado para reconstruir o arquivo
- todos os registros removidos são descartados



Exemplo: Abordagem Estática

- Arquivo original

M A R I A | R U A b 1 | S A O b C A R L O S | b b b b b b b
J O A O | R U A b A | R I O b C L A R O | b b b b b b b b
P E D R O | R U A b X V | S A O b C A R L O S | b b b b b b
A N T O N I A | R U A b X V b D E b M A I O | I B A T E | b
A N A | R U A b A U G U S T O b P A I V A | I B A T E | b b

1. Recupere os dados do registro com **RRN = 1**
2. Remova os registros com **RRN = 1** e **RRN = 3**



Exemplo: Abordagem Estática

- Arquivo após remoção lógica

M A R I A | R U A b 1 | S A O b C A R L O S | b b b b b b b
* O A O | R U A b A | R I O b C L A R O | b b b b b b b b
P E D R O | R U A b X V | S A O b C A R L O S | b b b b b b
* N T O N I A | R U A b X V b D E b M A I O | I B A T E | b
A N A | R U A b A U G U S T O b P A I V A | I B A T E | b b

3. Recupere os dados do registro com **RRN = 1**
4. Compacte o arquivo



Exemplo: Abordagem Estática

- Arquivo compactado

M A R I A | R U A b 1 | S A O b C A R L O S | b b b b b b b
P E D R O | R U A b X V | S A O b C A R L O S | b b b b b b
A N A | R U A b A U G U S T O b P A I V A | I B A T E | b b



Características: Abordagem Estática

- Técnica pode ser aplicada a
 - registros de tamanho fixo
 - registros de tamanho variável
- Frequência para se aplicar a técnica
 - depende da aplicação
 - depende da porcentagem de registros marcados como removidos



Abordagem Dinâmica

- Indicada para aplicações interativas que acessam arquivos altamente voláteis
- Desafios
 - **marcar** registros como logicamente removidos
 - **identificar** se existem registros marcados como logicamente removidos, ou seja, se existem espaços a serem reaproveitados
 - **localizar** os espaços ocupados por esses registros logicamente removidos sem realizar buscas exaustivas



Registros de Tamanho Fixo: Abordagem Dinâmica

- Solução
 - **lista encadeada** de registros eliminados
- Características
 - **lista**: constitui-se dos **RRNs** dos registros marcados como logicamente removidos
 - **cabeça da lista**: armazenada no registro de cabeçalho do arquivo
 - **inserção e reuso de espaço**: ocorrem sempre no início da lista

Implementação: **PILHA**

Registros de Tamanho Fixo: Abordagem Dinâmica

- Arquivo original

- registro de cabeçalho: `TopoPilha` → `-1 (NIL)`
- registros de dados:

```
M A R I A | R U A b 1 | S A O b C A R L O S | b b b b b b b
J O A O | R U A b A | R I O b C L A R O | b b b b b b b b
P E D R O | R U A b X V | S A O b C A R L O S | b b b b b b
A N T O N I A | R U A b X V b D E b M A I O | I B A T E | b
A N A | R U A b A U G U S T O b P A I V A | I B A T E | b b
```

1. Remova o registro com `RRN = 1`

Registros de Tamanho Fixo: Abordagem Dinâmica

- Arquivo após remoção lógica
 - registro de cabeçalho: **TopoPilha** → 1
 - registros de dados:

```
M A R I A | R U A b 1 | S A O b C A R L O S | b b b b b b b
*-1 A O | R U A b A | R I O b C L A R O | b b b b b b b b
P E D R O | R U A b X V | S A O b C A R L O S | b b b b b b
A N T O N I A | R U A b X V b D E b M A I O | I B A T E | b
A N A | R U A b A U G U S T O b P A I V A | I B A T E | b b
```

2. Remova o registro com **RRN = 3**

Registros de Tamanho Fixo: Abordagem Dinâmica

- Arquivo após remoção lógica
 - registro de cabeçalho: **TopoPilha** → 3
 - registros de dados:

M A R I A		R U A	b 1		S A O	b C A R L O S		b b b b b b b	
* -1 A O		R U A	b A		R I O	b C L A R O		b b b b b b b b	
P E D R O		R U A	b X V		S A O	b C A R L O S		b b b b b b b	
* 1 T O N I A		R U A	b X V	b D E	b M A I O		I B A T E		b
A N A		R U A	b A U G U S T O	b P A I V A		I B A T E		b b	

3. Remova o registro com **RRN = 0**

Registros de Tamanho Fixo: Abordagem Dinâmica

- Arquivo após remoção lógica
 - registro de cabeçalho: `TopoPilha` → 0
 - registros de dados:

```
* 3 R I A | R U A b 1 | S A O b C A R L O S | b b b b b b b
*-1 A O | R U A b A | R I O b C L A R O | b b b b b b b b
P E D R O | R U A b X V | S A O b C A R L O S | b b b b b b
* 1 T O N I A | R U A b X V b D E b M A I O | I B A T E | b
A N A | R U A b A U G U S T O b P A I V A | I B A T E | b b
```

3. Insira um novo registro

Registros de Tamanho Fixo: Abordagem Dinâmica

- Arquivo após inserção
 - registro de cabeçalho: **TopoPilha** → 3
 - registros de dados:

```
C A T A R I N A | A V E N I D A b X | R I O b C L A R O | b
*-1 A O | R U A b A | R I O b C L A R O | b b b b b b b b
P E D R O | R U A b X V | S A O b C A R L O S | b b b b b b
* 1 T O N I A | R U A b X V b D E b M A I O | I B A T E | b
A N A | R U A b A U G U S T O b P A I V A | I B A T E | b b
```




Registros de Tamanho Variável: Abordagem Dinâmica

- Solução
 - **lista encadeada** de registros eliminados
- Características
 - **lista**: constitui-se dos *byte offsets* dos registros marcados como logicamente removidos
 - **cabeça da lista**: armazenada no registro de cabeçalho do arquivo
 - **inserção**: ocorre sempre no início da lista (como uma pilha)

Necessário guardar também o **tamanho do registro**



Registros de Tamanho Variável: Abordagem Dinâmica

- Arquivo original

- registro de cabeçalho: `TopoLista` → `-1 (NIL)`
- registros de dados:

```
M A R I A | R U A b 1 | S A O b C A R L O S | # J O A O | R
U A b A | R I O b C L A R O | # P E D R O | R U A b X V | S
A O b C A R L O S | # A N T O N I A | R U A b X V b D E b M
A I O | I B A T E | # A N A | R U A b A U G U S T O b P A I
V A | I B A T E | #
```

1. Remova o registro relativo ao JOAO



Registros de Tamanho Variável: Abordagem Dinâmica

- Arquivo original

- registro de cabeçalho: `TopoLista` → 24
- registros de dados:

```
M A R I A | R U A b 1 | S A O b C A R L O S | # * 22 -1 O | R
U A b A | R I O b C L A R O | # P E D R O | R U A b X V | S
A O b C A R L O S | # A N T O N I A | R U A b X V b D E b M
A I O | I B A T E | # A N A | R U A b A U G U S T O b P A I
V A | I B A T E | #
```

2. Remova o registro relativo à ANA



Registros de Tamanho Variável: Abordagem Dinâmica

- Arquivo original
 - registro de cabeçalho: `TopoLista` → 101
 - registros de dados:

```
M A R I A | R U A b 1 | S A O b C A R L O S | # * 22 -1 O | R
U A b A | R I O b C L A R O | # P E D R O | R U A b X V | S
A O b C A R L O S | # A N T O N I A | R U A b X V b D E b M
A I O | I B A T E | # * 29 24 | R U A b A U G U S T O b P A I
V A | I B A T E | #
```

3. Remova o registro relativo à MARIA

Registros de Tamanho Variável: Abordagem Dinâmica

- Arquivo original

- registro de cabeçalho: `TopoLista` → 0
- registros de dados:

```
* 24 101  A | R U A b 1 | S A O b C A R L O S | # * 22 -1  O | R
U A b A | R I O b C L A R O | # P E D R O | R U A b X V | S
A O b C A R L O S | # A N T O N I A | R U A b X V b D E b M
A I O | I B A T E | # * 29 24 | R U A b A U G U S T O b P A I
V A | I B A T E | #
```



Registros de Tamanho Variável: Abordagem Dinâmica

- Reuso de espaço
 - realiza uma **busca sequencial** na lista
 - se encontrou espaço disponível no **tamanho adequado**
 - então **reaproveita** o espaço para armazenar o novo registro, usando uma **estratégia de alocação**
 - senão **insere** o novo registro **no final do arquivo**

O tamanho do registro que foi removido deve ser do **tamanho adequado**, ou seja, “grande o suficiente” para que os dados do novo registro usem aquele espaço



Estratégias de Alocação

- First-Fit
 - utiliza o **primeiro** espaço que servir
- Best-Fit
 - escolhe o espaço **mais justo** possível
- Worst-Fit
 - escolhe o **maior espaço** possível



Reuso de Espaço: Estratégia First-Fit

- Arquivo original
 - registro de cabeçalho: `TopoLista` → 0
 - registros de dados:

```
* 24 101  A | R U A b 1 | S A O b C A R L O S | # * 22 -1  O | R
U A b A | R I O b C L A R O | # P E D R O | R U A b X V | S
A O b C A R L O S | # A N T O N I A | R U A b X V b D E b M
A I O | I B A T E | # * 29 24 | R U A b A U G U S T O b P A I
V A | I B A T E | #
```

4. Insira um novo registro de tamanho 18 usando a estratégia `First-Fit`

Reuso de Espaço: Estratégia First-Fit

- Arquivo original

- registro de cabeçalho: `TopoLista` → 101
- registros de dados:

```
R I T A | R U A b X | I B A T E | A R L O S | # * 22 -1 O | R
U A b A | R I O b C L A R O | # P E D R O | R U A b X V | S
A O b C A R L O S | # A N T O N I A | R U A b X V b D E b M
A I O | I B A T E | # * 29 24 | R U A b A U G U S T O b P A I
V A | I B A T E | #
```

fragmentação
interna



Reuso de Espaço: Estratégia Best-Fit

- Arquivo original
 - registro de cabeçalho: **TopoLista** → 0
 - registros de dados:

```
* 24 101  A | R U A b 1 | S A O b C A R L O S | # * 22 -1  O | R
U A b A | R I O b C L A R O | # P E D R O | R U A b X V | S
A O b C A R L O S | # A N T O N I A | R U A b X V b D E b M
A I O | I B A T E | # * 29 24 | R U A b A U G U S T O b P A I
V A | I B A T E | #
```

4. Insira um novo registro de tamanho 18 usando a estratégia **Best-Fit**

Reuso de Espaço: Estratégia Best-Fit

- Arquivo original

- registro de cabeçalho: `TopoLista` → 0
- registros de dados:

```
* 24 101  A | R U A b 1 | S A O b C A R L O S | # R I T A | R
U A b X | I B A T E | A R O | # P E D R O | R U A b X V | S
A O b C A R L O S | # A N T O N I A | R U A b X V b D E b M
A I O | I B A T E | # * 29 -1 | R U A b A U G U S T O b P A I
V A | I B A T E | #
```

fragmentação
interna



Reuso de Espaço: Estratégia Worst-Fit

- Arquivo original
 - registro de cabeçalho: **TopoLista** → 0
 - registros de dados:

```
* 24 101  A | R U A b 1 | S A O b C A R L O S | # * 22 -1  O | R
U A b A | R I O b C L A R O | # P E D R O | R U A b X V | S
A O b C A R L O S | # A N T O N I A | R U A b X V b D E b M
A I O | I B A T E | # * 29 24 | R U A b A U G U S T O b P A I
V A | I B A T E | #
```

4. Insira um novo registro de tamanho 18 usando a estratégia **Worst-Fit**

Reuso de Espaço: Estratégia Worst-Fit

- Arquivo original
 - registro de cabeçalho: `TopoLista` → 0
 - registros de dados:

```
* 24 24  A | R U A b 1 | S A O b C A R L O S | # * 22 -1 O | R
U A b A | R I O b C L A R O | # P E D R O | R U A b X V | S
A O b C A R L O S | # A N T O N I A | R U A b X V b D E b M
A I O | I B A T E | # R I T A | R U A b X | I B A T E | A I
V A | I B A T E | #
```

fragmentação
interna



Fragmentação Interna

- Definição
 - espaço que sobra dentro de um registro
 - pode ocorrer com qualquer estratégia de alocação
- Solução
 - colocar o espaço que sobrou na lista encadeada como um registro eliminado



Reuso de Espaço: Estratégia First-Fit

- Arquivo original

- registro de cabeçalho: `TopoLista` → 0
- registros de dados:

```
* 24 101  A | R U A b 1 | S A O b C A R L O S | # * 22 -1  O | R
U A b A | R I O b C L A R O | # P E D R O | R U A b X V | S
A O b C A R L O S | # A N T O N I A | R U A b X V b D E b M
A I O | I B A T E | # * 29 24 | R U A b A U G U S T O b P A I
V A | I B A T E | #
```

4. Insira um novo registro de tamanho 18 usando a estratégia `First-Fit`, *com tratamento de fragmentação interna*



Reuso de Espaço: Estratégia First-Fit

- Arquivo original

- registro de cabeçalho: `TopoLista` → 0
- registros de dados:

```
* 6 101  A # R I T A | R U A b X | I B A T E | # * 22 -1 O | R
U A b A | R I O b C L A R O | # P E D R O | R U A b X V | S
A O b C A R L O S | # A N T O N I A | R U A b X V b D E b M
A I O | I B A T E | # * 29 24 | R U A b A U G U S T O b P A I
V A | I B A T E | #
```




Fragmentação Externa

- Definição

- o espaço que sobrou dentro de um registro foi colocado na lista encadeada como um registro eliminado
- o espaço é muito pequeno, e não pode armazenar nenhum dado
- pode ocorrer com qualquer estratégia de alocação



Fragmentação Externa

- Como combater
 - junção de espaços vazios adjacentes (*coalescimento*)
 - combinação de dois espaços vazios na lista que são fisicamente adjacentes em um espaço único maior
- Dificuldade
 - a adjacência de registros na lista é lógica, não física, o que requer a busca por registros adjacentes



Observações

- Estratégias de alocação
 - usadas somente com registros de tamanho variável
- Recomendações
 - se o espaço está sendo desperdiçado devido à **fragmentação interna**
 - então escolha entre First-Fit e Best-Fit
 - se o espaço está sendo desperdiçado devido à **fragmentação externa**
 - então escolha Worst-Fit