Acesso, Manutenção e Reuso de Espaço

Prof. André Backes | @progdescomplicada

Relembrando...

- De modo geral, as informações em arquivos são organizadas logicamente em campos e registros
  - campos e registros são conceitos lógicos
  - possuem associação com o arquivo lógico

Produto	Quantidade
Pão	10
Leite	2
Manteiga	1

Relembrando...

- Campo
  - Menor unidade lógica de informação em arquivo
  - É uma ferramenta conceitual
  - Não está associado a um conceito físico
- Registro
  - É um conjunto de campos agrupados

Produto	Quantidade
Pão	10
Leite	2
Manteiga	1

Registros e campos com tamanho fixo

- Um dos métodos mais comuns de organização de arquivos:
  - É simples e permite acesso direto aos registros por RRN (*Relative Record Number*)
- Porém, pode ser inapropriado...
  - Desperdício de memória secundária – fragmentação

```
struct item{
    char produto[20];
    int quantidade;
};
```

RRN	Produto	Quantidade
0	Pão	10
1	Leite	2
2	Manteiga	1

Registros e campos com tamanho fixo

- Operações / formas de acesso
  - Processamento sequencial
  - Acesso Direto (RRN / Byte Offset)
  - Busca de registro por valor de chave primária (identificador único)
  - Busca de registros por valor de um campo

RRN	Produto	Quantidade
0	Pão	10
1	Leite	2
2	Manteiga	1

Manipulação de Dados

- Operações básicas
  - Inserção de um novo registro
  - Atualização de um registro
  - Remoção de um registro
- Quando um registro é removido, deve-se posteriormente reutilizar o espaço do registro

Inserção de um novo registro

RRN	Produto	Quantidade
0	Pão	10
1	Leite	2
2	Manteiga	1

RRN	Produto	Quantidade
0	Pão	10
1	Leite	2
2	Manteiga	1
3	**Café**	**2**

Atualização de um registro

RRN	Produto	Quantidade
0	Pão	10
1	Leite	2
2	Manteiga	1

RRN	Produto	Quantidade
0	Pão	10
1	Leite	**3**
2	Manteiga	1

Atualização de um registro

- A atualização de um registro pode envolver a eliminação e inserção de um novo registro
  - Ocorre se o novo valor dos dados implicar em crescimento do tamanho do registro
  - Registro de tamanho variável
  - Nesse caso, devemos excluir o registro atual e acrescentar o novo registro ao final

Remoção de um registro

- Formas de implementação
  - Reorganização imediata
  - Reuso estático
  - Reuso dinâmico
  - Reuso dinâmico para registros de tamanho variável

Reorganização imediata

RRN	Produto	Quantidade
0	Pão	10
1	**Leite**	**2**
2	Manteiga	1
3	Café	2

RRN	Produto	Quantidade
0	Pão	10
1	Manteiga	1
2	Café	2

Solução inviável na prática
Custo muito alto da operação

Compactação e Reuso

- Compactação
  - Busca por regiões do arquivo que não contêm dados
  - Recupera os espaços perdidos
- Reuso
  - Insere dados nos espaços perdidos
- Duas abordagem
  - Estática versus Dinâmica

Reuso Estático

- Atribuir um valor para um campo do registro para reconhecer registros removidos
  - Usar um campo extra
- Remoção lógica
  - Remoção do registro
  - Gera espaço sem dados úteis
  - Reagrupa os registros e diminui o tamanho do arquivo

RRN	Ativo	Produto	Quantidade
0	1	Pão	10
1	**0**	**Leite**	**2**
2	1	Manteiga	1
3	1	Café	2

Reuso Estático

- Programa "compactador"
  - 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

Reuso Estático

- Exemplo
  - Recupere os dados do registro com RRN = 1
  - Remova os registros com RRN = 1 e RRN = 3

RRN	Ativo	Produto	Quantidade
0	1	Pão	10
1	**0**	**Leite**	**2**
2	1	Manteiga	1
3	**0**	**Café**	**2**
4	1	Arroz	1

Reuso Estático

- Exemplo
  - Recupere os dados do registro com RRN = 2
  - Insira um novo registro

RRN	Ativo	Produto	Quantidade
0	1	Pão	10
1	**0**	**Leite**	**2**
2	1	Manteiga	1
3	**0**	**Café**	**2**
4	1	Arroz	1
5	1	Feijão	3

Reuso Estático

- Exemplo
  - Recupere os dados do registro com RRN = 4
  - Compacte o arquivo

RRN	Ativo	Produto	Quantidade
0	1	Pão	10
1	1	Manteiga	1
2	1	Arroz	1
3	1	Feijão	3

Reuso Estático

- Técnica pode ser aplicada a
  - Registros de tamanho fixo (delimitador)
  - Registros de tamanho variável
  - Aplicações sem tanta volatilidade nos dados
- Algoritmo de Reorganização:
  - Copiar não marcados para novo arquivo (mais simples)
  - In loco

Reuso Estático

- Frequência para se aplicar a técnica
  - Depende da aplicação
  - Depende da porcentagem de registros marcados como removidos
- Período de tempo para reorganização depende
  - Degradação (percentual de registros marcados)
  - Tempo ocioso
  - Aplicação

Reuso Dinâmico

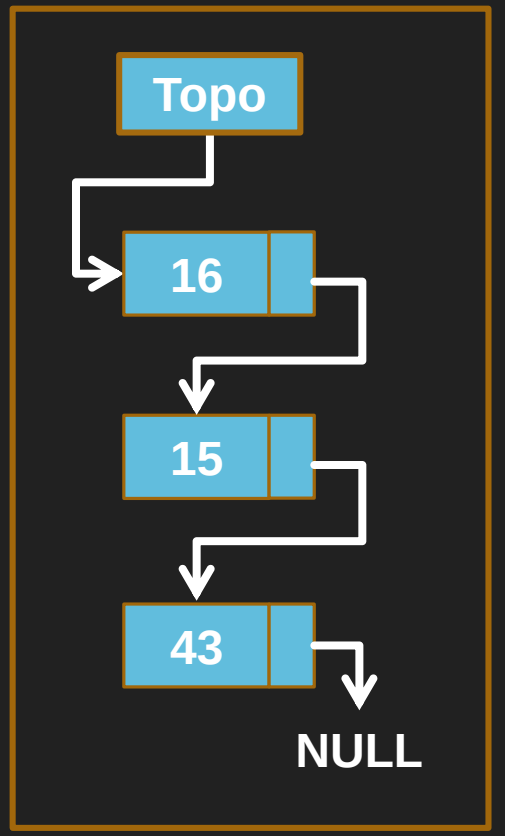
- Indicada para aplicações interativas que acessam arquivos altamente voláteis
  - Situações com alterações constantes nos dados (exclusões, inserções, alterações)
- Estratégia Geral
  - Na remoção, registros são marcados
  - Na inserção, antes de inserir no final do arquivo, verificar se há um registro marcado e reaproveita-lo

Reuso Dinâmico

- 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
  - Como ter acesso imediato ao registros marcados?

Reuso Dinâmico: registros de tamanho fixo

- **Solução**: lista encadeada de registros eliminados
  - Lista composta pelos 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



Reuso Dinâmico: registros de tamanho fixo

- Inicialização da lista de marcados
  - inicialmente, vazia (RRN = -1)
- Remoção de um registro
  - Marca o registro como logicamente removido
  - Insere o registro na lista de registros logicamente removidos (empilha)

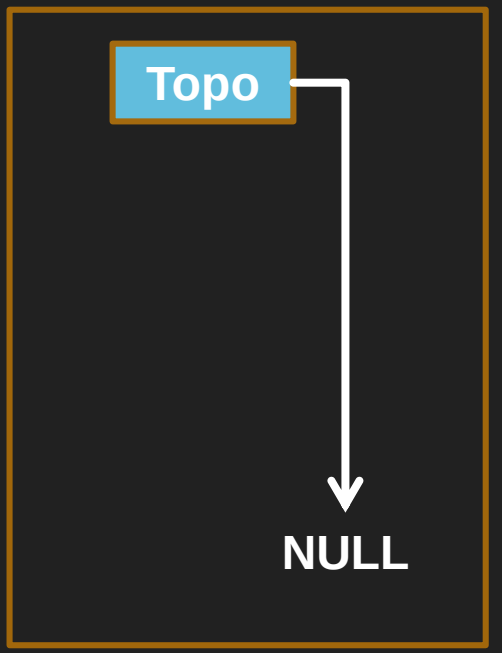
Reuso Dinâmico: registros de tamanho fixo

- Inserção de um registro de dados
  - Se a lista estiver vazia, insere no final do arquivo
  - Caso contrário, remove o registro da lista de registros logicamente removidos (desempilha)
  - Insere os dados no espaço do registro desempilhado
- Atualização de um registro de dados
  - Ocorre no próprio registro

Reuso Dinâmico: registros de tamanho fixo

- Exemplo
  - Arquivo com dados
  - Lista inicialmente está vazia

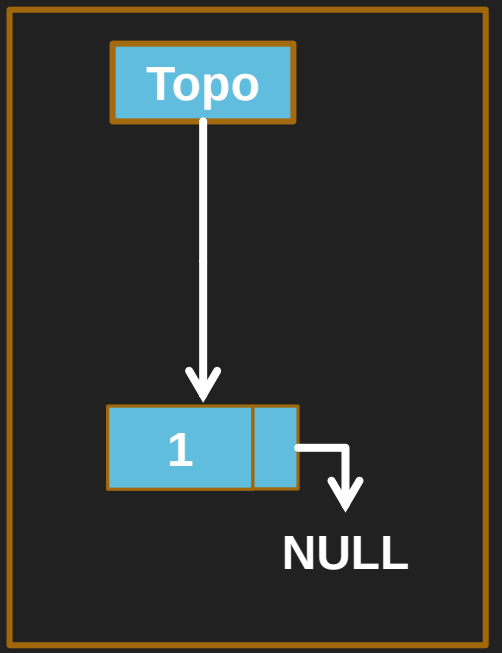
RRN	Ativo	Produto	Quantidade
0	1	Pão	10
1	1	Leite	2
2	1	Manteiga	1
3	1	Café	2



Reuso Dinâmico: registros de tamanho fixo

- Remova o registro com RRN = 1
- Registro é inserido na pilha

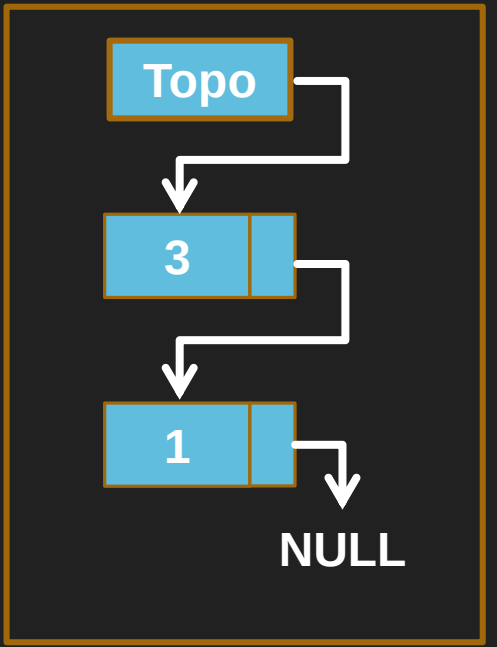
RRN	Ativo	Produto	Quantidade
0	1	Pão	10
1	**0**	**Leite**	**2**
2	1	Manteiga	1
3	1	Café	2



Reuso Dinâmico: registros de tamanho fixo

- Remova o registro com RRN = 3
- Registro é inserido na pilha

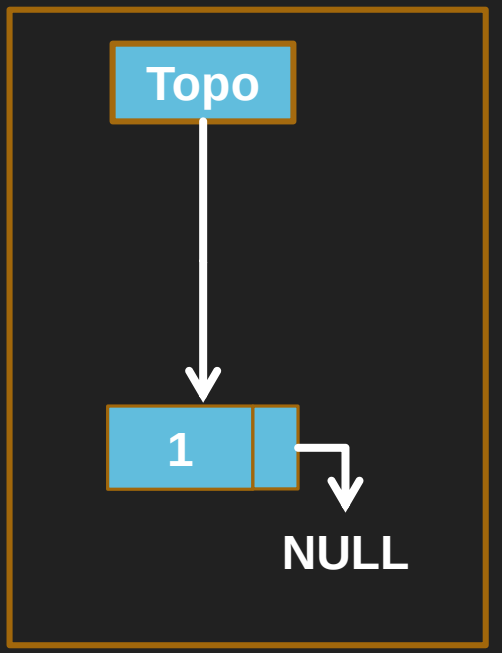
RRN	Ativo	Produto	Quantidade
0	1	Pão	10
1	**0**	**Leite**	**2**
2	1	Manteiga	1
3	**0**	**Café**	**2**



Reuso Dinâmico: registros de tamanho fixo

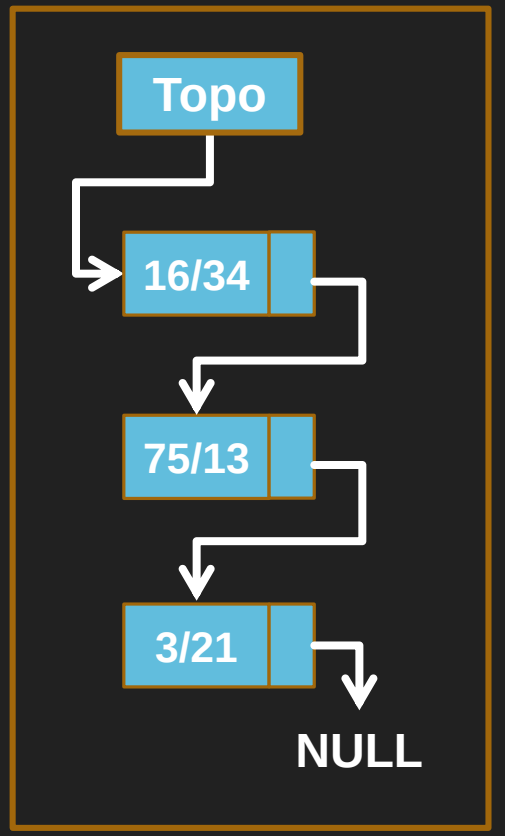
- Insira um novo registro
- Registro é removido da pilha e utilizado

RRN	Ativo	Produto	Quantidade
0	1	Pão	10
1	**0**	**Leite**	**2**
2	1	Manteiga	1
3	1	Arroz	5



Reuso Dinâmico: registros de tamanho variável

- **Solução**: lista encadeada de registros eliminados
  - Lista composta pelos RRNs dos registros marcados como logicamente removidos
  - **Necessário guardar também o tamanho do registro**
  - Cabeça da lista: armazenada no registro de cabeçalho do arquivo
  - **Inserção ocorre no início da lista**
  - Implementação: PILHA



Reuso Dinâmico: registros de tamanho variável

- Inicialização da lista de marcados
  - inicialmente, vazia (RRN = -1)
- Remoção de um registro
  - Marca o registro como logicamente removido
  - Insere o registro (posição e tamanho) na lista de registros logicamente removidos (empilha)

Reuso Dinâmico: registros de tamanho variável

- Inserção de um registro de dados
  - Se a lista estiver vazia, insere no final do arquivo
  - Caso contrário, procura na lista de registros logicamente removidos um com tamanho adequado
  - Insere os dados no espaço do registro selecionado
- Atualização de um registro de dados
  - Se tamanho do registro é suficiente, ocorre no próprio registro; senão, remove e inclui

Reuso Dinâmico: registros de tamanho variável

- 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

Reuso Dinâmico: registros de tamanho variável

- Estratégias de Alocação
  - First-Fit
    - Utiliza o primeiro espaço que tiver tamanho suficiente
  - Best-Fit
    - Escolhe o espaço mais justo possível
  - Worst-Fit
    - Escolhe o maior espaço possível

Reuso Dinâmico: registros de tamanho variável

- Exemplo: inserção de um novo registro de tamanho 10
  - First-Fit: RRN = 252
  - Best-Fit: RRN = 75
  - Worst-Fit: RRN = 113

RRN	Tamanho
252	34
75	13
13	21
113	51

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

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
  - 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