Operações em arquivos sequenciais



Interface do CRUD

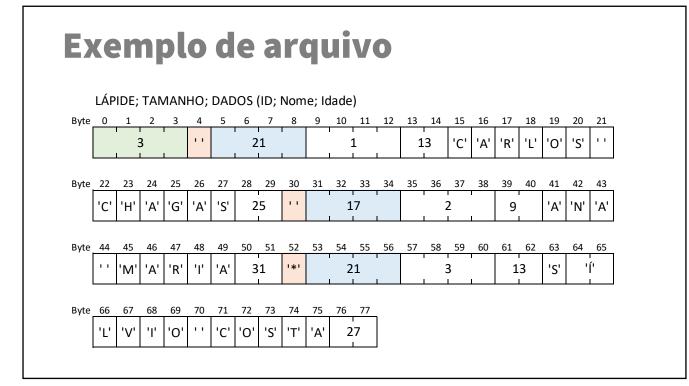
O ID passado nesse novo objeto será ignorado (usar -1)

- ID ← arquivo.create(novo_objeto)
- objeto ← arquivo.read(ID)
- ok ← arquivo.update(objeto_atualizado)
- ok ← arquivo.delete(ID)

O ID deve permanecer o mesmo, pois IDs nunca são alterados.

Lápide

- Marca de exclusão
 - Campo (1 byte) que indica se o registro foi excluído ou permanece válido
- Os registros só são realmente apagados do arquivo quando há uma reordenação



Create (inclusão)

• Arquivo sem ordenação específica

```
01: algoritmo create(objeto)
        mover o ponteiro para início do arquivo (cabeçalho)
02:
03:
        <u>ler</u> últimoID
        objeto.ID ← últimoID + 1
04:
05:
        mover o ponteiro para início do arquivo
        escrever objeto.ID
06:
        criar registro para o objeto
07:
08:
        mover para o fim do arquivo
09:
        escrever registro
10: fim-algoritmo
```

Read (leitura)

Algoritmo de busca sequencial

```
01: algoritmo read(ID)
02:
         mover o ponteiro para o primeiro registro (após o cabeçalho)
03:
         enquanto não atingir o fim do arquivo
             <u>ler</u> próximo registro
04:
05:
             <u>se</u> registro.lapide ≠ '*'
                  então extrair objeto do registro
06:
                         se objeto.ID = ID
07:
                             então retornar objeto e terminar
09:
                         <u>fim-se</u>
             <u>fim-</u>se
10:
11:
         fim-enquanto
         retornar objeto vazio // null
13: fim-algoritmo
```

Read (leitura)

• Variação: Leitura de conjunto de objetos

```
01: algoritmo read(critérios)
02:
        criar conjunto vazio
        mover o ponteiro para o primeiro registro (após o cabeçalho)
03:
04:
        enquanto não atingir o fim do arquivo
05:
             <u>ler</u> próximo registro
             se registro.lapide ≠ '*'
                 então extrair objeto do registro
07:
                       se registro atender aos critérios
08:
09:
                            então adicionar objeto ao conjunto
10:
                       fim-se
            fim-se
11:
12:
        fim-enquanto
13:
        retornar conjunto
14: fim-algoritmo
```

Delete (exclusão)

```
01: algoritmo delete(ID)
02:
         mover o ponteiro para o primeiro registro (após o cabeçalho)
03:
         enquanto não atingir o fim do arquivo
04:
             pos ← posição do ponteiro
05:
             ler próximo registro
             <u>se</u> registro.lapide ≠ '*'
06:
07:
                  então extrair objeto do registro
08:
                         se objeto.ID = ID
09:
                             então mover para pos
                                    escrever lápide como excluído
10:
                                    <u>retornar</u> verdadeiro e <u>terminar</u>
11:
12:
                         fim-se
13:
             <u>fim-se</u>
14:
         fim-enquanto
15:
         retornar falso
16: fim-algoritmo
```

Update (atualização)

- Se a alteração não implicar em mudança de tamanho
 - Escrever o registro alterado na mesma posição
- Se o registro diminuir de tamanho
 - Escrever o registro na mesma posição, mas mantendo o mesmo indicador de tamanho
- Se o registro aumentar de tamanho
 - Marcar o registro atual como excluído e criar um novo registro no fim do arquivo

```
01: algoritmo update(novoObjeto)
          mover para o primeiro registro do arquivo (após cabeçalho)
02:
          enquanto não atingir o fim do arquivo
03:
               pos \leftarrow posição do ponteiro
04:
05:
               <u>ler</u> próximo registro
               <u>se</u> registro.lapide ≠ '*'
                    então extrair objeto do registro
07:
08:
                           se objeto.ID = novoObjeto.ID
09:
                              então criar novoRegistro para novoObjeto
                                     se novoRegistro.tamanho ≤ registro.tamanho
10:
11:
                                        então mover para pos
                                               escrever novoRegistro mantendo ind.tam.
12:
13:
                                        senão mover para pos
14:
                                               escrever lápide como excluído
                                               mover para fim do arquivo
15:
16:
                                               escrever novoRegistro
17:
                                     fim-se
                                <u>retornar</u> verdadeiro e <u>terminar</u>
18:
19:
                           fim-se
20:
               fim-se
21:
          fim-enquanto
          retornar falso
23: fim-algoritmo
```

Considerações finais

- Arquivos sequenciais dependem de acesso sequencial, o que significa que eles não são bons para quaisquer operações de acesso aleatório
- Os espaços deixados pelos registros excluídos são espaços que podem ser reaproveitados (desde que exista uma lógica para esse reaproveitamento)
- O arquivo deve ser reordenado sempre que necessário (ordenação externa)