Algoritmos e Estruturas de Dados II – SCC-203

Arquivos: Recuperação de Espaço

Gustavo Batista

Recuperação de espaço

- A organização de um arquivo pode deteriorar durante o seu uso.
- Por exemplo, o que deve ser feito com o espaço liberado por registros removidos?
- E quando registros são atualizados em arquivos com registros de tamanho variável?

Recuperação de espaço

- Atualizações em arquivos são:
  - Adição de registros;
  - Modificação de registros;
  - Remoção de registros.
- Não há deterioração se forem feitas somente adições de registros.
- Modificação de registros de tamanho variável pode ser vista como uma remoção seguida de uma inserção.

Eliminação de registros

- Deve existir um mecanismo que permita reconhecer quando uma área corresponde a um registro que foi eliminado.
- Geralmente, isso é feito colocando um marcador em algum campo do registro removido ou utilizando um campo específico para esse fim.
- Essa estratégia permite recuperar registros removidos.

Compactação

- Compactação consiste na busca por regiões do arquivo que não contém dados, e posterior recuperação desses espaços não utilizados.
- Quando o procedimento de compactação é ativado, o espaço de todos os registros marcados é recuperado de uma só vez.
- A maneira mais simples de compactar é executando um programa de cópia de arquivos que "pule" os registros apagados.

Processo de compactação

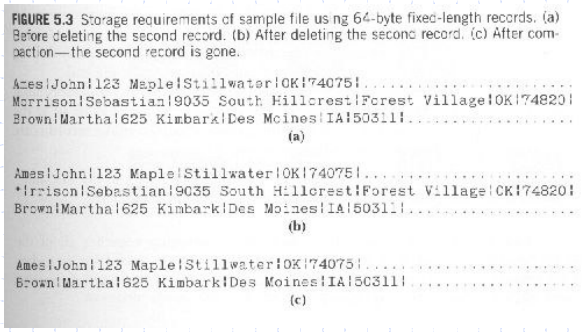


FIGURE 5.3 Storage requirements of sample file using 64-byte fixed-length records. (a) Before deleting the second record. (b) After deleting the second record. (c) After compaction—the second record is gone.

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

Ames|John|123 Maple|Stillwater|OK|74075|.....
*|rrison|Sebastian|9035 South Hillcrest|Forest Village|OK|74820|
Brown|Martha|625 Kimbark|Des Moines|IA|50311|.....
(b)

Ames|John|123 Maple|Stillwater|OK|74075|.....
Brown|Martha|625 Kimbark|Des Moines|IA|50311|.....
(c)
```

Recuperação dinâmica

- O procedimento de compactação é esporádico.
  - Um registro apagado não fica disponível para uso imediatamente.
- E para ter o espaço disponível imediatamente?
  - marcar registros apagados;
  - identificar e localizar os espaços antes ocupados pelos registros apagados, sem buscas exaustivas.

Como localizar os espaços vazios?

- Registros de tamanho fixo:
  - Lista encadeadas de registros eliminados (Dispo);
  - Lista constitui-se de espaços vagos, endereçados por meio de seus RRNs;
  - Cabeça da lista está no *header* do arquivo;
  - Um registro eliminado contém o RRN do próximo registro eliminado;
  - Inserção e remoção ocorrem sempre no início da lista (pilha!).

Registros de tamanho fixo

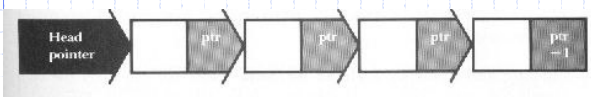
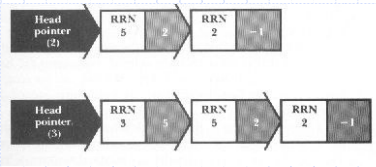


FIGURE 5.4 A linked list.



Pilha antes e depois da inserção do nó correspondente ao registro de RRN 3

Registros de tamanho fixo

- O uso de pilhas para recuperar espaços vagos permite:
  - Saber imediatamente se existem espaços vazios em um arquivo;
  - Pular diretamente para um desses espaços se ele existir.

Registros de tamanho variável

- Pode-se utilizar a mesma técnica de lista de disponíveis utilizada com registros de tamanho fixo.
- Entretanto, não se pode utilizar os RRNs. Deve-se utilizar os *offsets* dos registros.

Remoção de registros

```
HEAD.FIRST_AVAIL: -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|
```

(a)

```
HEAD.FIRST_AVAIL: 43

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

(b)

FIGURE 5.6 A sample file for illustrating variable-length record deletion. (a) Original sample file stored in variable-length format with byte count (header record not included). (b) Sample file after deletion of the second record (periods show discarded characters).

Registros de tamanho variável

- Com tamanho variável, não se pode mais utilizar a estrutura de pilha.
- É necessário uma busca seqüencial na lista para encontrar uma posição com espaço suficiente.
- Por exemplo, a inserção de um registro com 55 bytes:

Remoção da lista Dispo

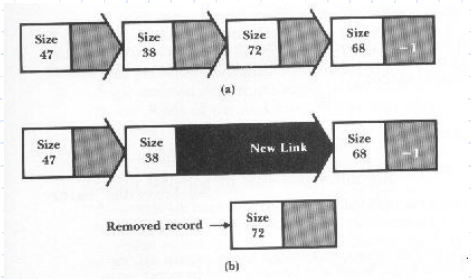


FIGURE 5.7 Removal of a record from an avail list with variable-length records. (a) Before removal. (b) After removal.

Fragmentação

- Fragmentação interna é o espaço não utilizado dentro dos registros.
- Frequentemente está associada a registros de tamanho fixo.
- Entretanto, pode ocorrer com registros de tamanho variável. No caso do exemplo anterior, um registro de 55 bytes é inserido em um espaço de 72 bytes. Sobram 17 bytes.

Fragmentação

FIGURE 5.10 Illustration of fragmentation with variable-length records. (a) After deletion of the second record (unused characters in the deleted record are replaced by periods). (b) After the subsequent addition of the record for Al Ham.

```
HEAD.FIRST_AVAIL: 43

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

(a)

```
HEAD.FIRST_AVAIL: -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|
```

(b)

Fragmentação

- Uma forma de combater a fragmentação interna é colocar o espaço não utilizado na lista de disponíveis.

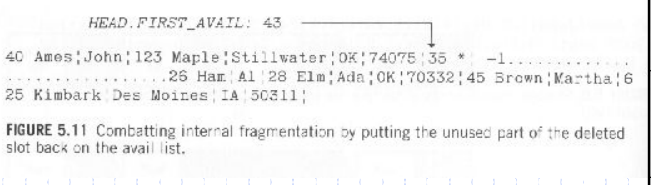


FIGURE 5.11 Combatting internal fragmentation by putting the unused part of the deleted slot back on the avail list.

Fragmentação

- Entretanto, qual é a probabilidade de encontrar um ou mais registros que ocupem exatamente o espaço restante?
- Por exemplo, se o espaço restante de 35 bytes for ocupado por um registro de 33 bytes, um espaço de apenas 2 bytes ficará livre.
- Esse problema é conhecido como **fragmentação externa**.

Fragmentação

- Para combater a fragmentação externa pode-se utilizar:
  - Uma compactação do arquivo de tempos em tempos;
  - Se dois espaços não ocupados são fisicamente adjacentes, combinar esses espaços. Esse método é conhecido como **coalizão**;
  - Minimizar a fragmentação por meio da escolha do espaço a ser ocupado. Conhecido como estratégia de colocação.

Estratégia de posicionamento

- As principais estratégias de posicionamento são:
  - *First-fit*: a lista de disponíveis não tem ordem. Não importa se o espaço livre é muito maior que o registro;
  - *Best-fit*: lista de disponíveis em ordem crescente de espaço. Procura-se o espaço livre mais próximo do registro;
  - *Worst-fit*: lista de disponíveis em ordem decrescente. Procura-se pelo maior espaço livre disponível.

Estratégia de posicionamento

- Caracteristicas das estratégias de posicionamento:
  - *Best-fit* requer percorrer a lista de disponíveis para encontrar um espaço livre e para inserir espaços livres.
  - *Worst-fit* requer percorrer a lista de disponíveis apenas para inserir espaços livres;
  - *Best-fit* faz com que os espaços de fragmentação externa sejam muito pequenos.
  - *Worst-fit* faz com que o espaço de fragmentação externa seja o maior possível, aumentando as chances dele ser utilizado posteriormente.

Busca e Ordenação

- Busca:
  - **Seqüencial:** funciona para qualquer organização de arquivo. Lenta *O(n)*.
  - **Binária:** mais complexa para arquivos com registros de tamanho variável (requer estrutura adicional). Requer que o arquivo esteja ordenado. Mais eficiente *O(log n)*.
- Para realizar uma busca binária é necessário ordenar o arquivo.

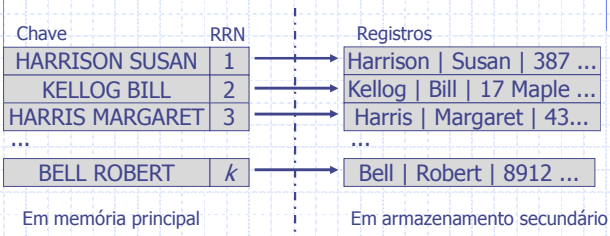
Ordenação

- Se o arquivo é pequeno, pode-se carregá-lo em memória principal, ordená-lo e gravá-lo.
- Entretanto, deve-se encontrar um método de ordenação capaz de ordenar grandes arquivos.
- Um método de ordenação interna diretamente adaptado para ordenação externa poderia fazer um número de acesso ao disco muito grande.

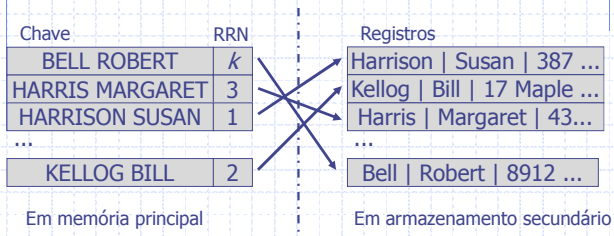
Ordenação: Keysorting

- *Keysorting* ou *tag sort* é baseado na idéia que o importante é ordenar as chaves e portanto o arquivo não precisa ser integralmente carregado em memória.
- Dessa forma, somente as chaves e RRNs (supondo um arquivo de registros de tamanho fixo) são carregados em memória principal e ordenados com um método de ordenação interna.

Algoritmo Keysorting



Algoritmo Keysorting



Limitações Keysorting

- Principais limitações:
  - Necessidade de ler todo o arquivo de entrada duas vezes;
  - A segunda leitura não é seqüencial, mas aleatória: custo de posicionamento das cabeças é grande.
  - A gravação do arquivo de saída é aparentemente seqüencial. Mas os *seeks* fazem com que o custo de gravar o arquivo seja alto.

Agradecimentos

- Slides da profa. Cristina Ferreira.

28