

Aula 4

Revisão

- O que influencia na escolha do método de acesso
- Métodos de acesso
 - Seqüencial
 - Direto
 - Indexado

Métodos de indexação

- Lista encadeada – um registro aponta para o outro
- Lista duplamente encadeada – um registro aponta para o próximo e para o anterior
- Pesquisa binária – método muito rápido e eficiente de pesquisa
- Árvore binária
 - Se bem criada é tão eficaz como a pesquisa binária
 - No pior caso se comporta como uma lista encadeada

Arquivos indexados em árvore B

- Os métodos anteriores apresentam queda de performance com o crescimento do banco de dados
- As árvores B requerem menos reindexações
- Uma árvore B de ordem m deve satisfazer as condições:
 - Todo nó tem no máximo m descendentes.
 - Todo nó, exceto a raiz e os nós terminais tem no mínimo $m/2$ (arredondando para mais) descendentes.
 - A raiz tem pelo menos 2 descendentes a menos que seja nó terminal.
 - Todos os nós terminais aparecem no mesmo nível e não tem nenhuma informação.
 - Um nó interno com k descendentes contém $k-1$ valores de chave.
- Pesquisa se inicia na raiz até encontrar um valor em um nó folha
- Inserção é feita das “folhas” para a raiz dividindo nós “cheios”
- A árvore cresce para cima e não para baixo
- A exclusão pode resultar em *underflow* (nó muito vazio)
- Ocorrem trocas para juntar nós irmãos (troca intermediada pelo pai)

Indexação por hashing

- Um subconjunto de valores de chave é utilizado
- Existe uma função de hashing que “transforma” o valor de chave em um endereço físico
 - A função deve ser escolhida de forma a dar o menor número de conflitos possível
 - Uma função perfeita mapeia cada valor de chave em um endereço

Tópicos Especiais em Análise de Sistemas

- A pior função mapeia todas as chaves para um único endereço
- Inserção: é feita uma pesquisa para encontrar onde o registro deve ser inserido e o acrescenta
- Se houver conflito, o registro é acrescentado ao final da lista daquele endereço
- Exclusão: é feita a pesquisa para encontrar o registro e então ele é removido
- A função de hash pode ser estática ou dinâmica
 - Estática é definida no início e não pode ser alterada
 - Dinâmica pode ser alterada conforme o banco de dados se altera

Comparação entre hash e árvore

- Árvore trabalha bem para pesquisas por faixa de valores e o hashing para consultas simples (valor definido)
- A árvore tem menor eficiência, porém é mais fácil de implementar
- A inclusão e exclusão é mais fácil utilizando hashing
- O processamento seqüencial é mais fácil na árvore B
- A maioria dos bancos de dados utilizam a indexação em árvores B
- No PostgreSQL o projetista do banco de dados pode escolher entre árvore ou hash