Área do Conhecimento de Ciências Exatas e Engenharias
Bacharelado em Ciência da Computação

Índices Baseados em Listas Encadeadas

Organização de Arquivos

Prof. Daniel Luis Notari

Setembro - 2017

Problematização

"Como as listas encadeadas podem ser auxiliar a busca por informações armazenadas em arquivos de dados físicos?"

Momento 1: Índices Baseados em Listas Encadeadas

Índices baseado em lista encadeada

- A utilização de listas auxilia o acesso a registros por campos que sejam chave secundária.
- Há diversas aplicações em que a recuperação de informações precisa ser realizada com base em vários campos (dados) de um registro.
- Estes vários campos são representados pelas chaves secundárias.
  - pode ser qualquer atributo (campo) de um arquivo pelo qual deseja-se acessar os registros.

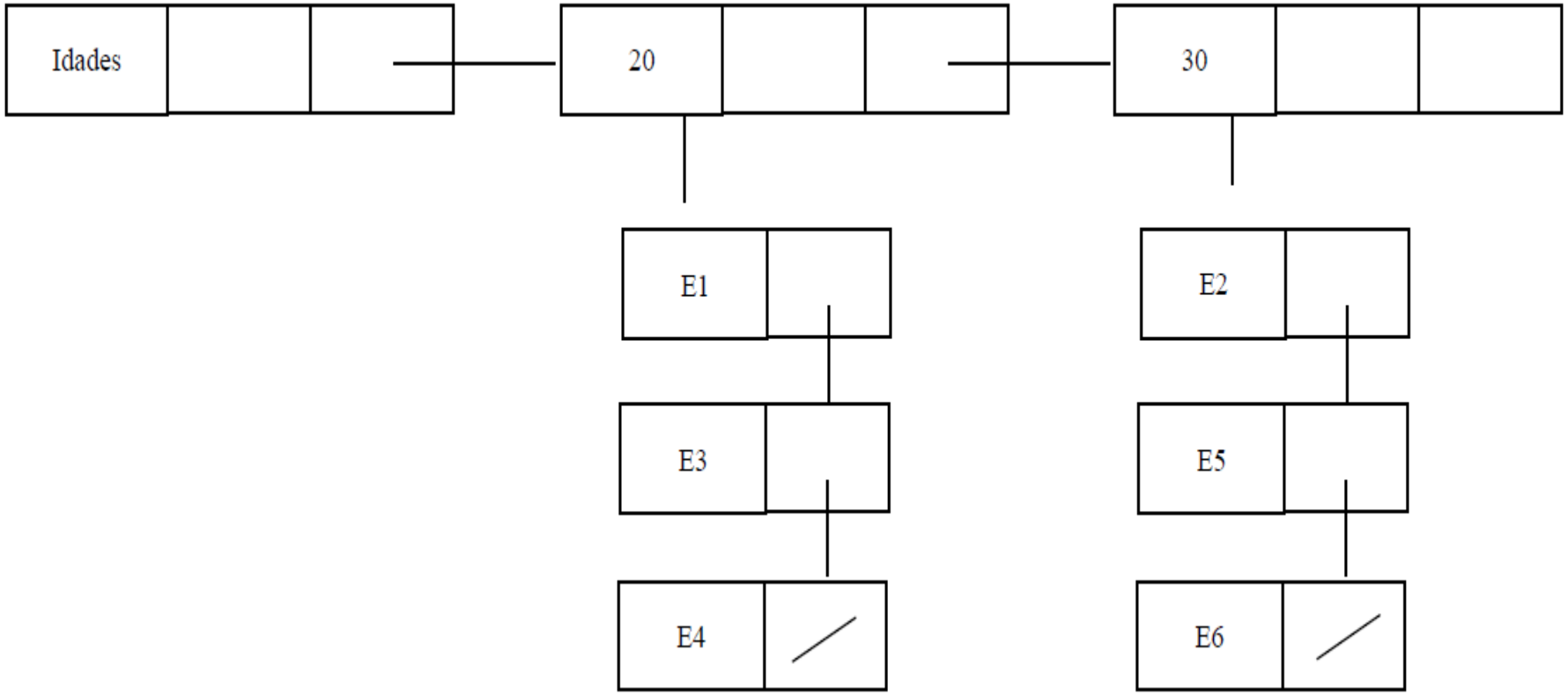
Índices baseado em lista encadeada

- Como um determinado valor de chave secundária pode pertencer a muitos registros,
  - Uma consulta realizada sobre essa chave pode trazer vários registros como resultado,
  - além disso, se a consulta incluir expressões booleanas acaba-se por consultar muito mais registros do que aqueles que farão parte do resultado.
- O tempo gasto com este tipo de consulta pode ser muito grande,
  - por esta razão, um arquivo precisa ser organizado de forma a minimizar o esforço de pesquisa.

Índices baseado em lista encadeada

- De forma genérica pode-se pensar nesta estrutura de lista contendo:
  - uma lista para cada chave secundária, que conterá os valores ordenados que essa chave possui nos registros;
  - cada elemento da lista acima formará uma lista que conterá os endereços dos registros que possuem esse valor de chave secundária.

Índices baseado em lista encadeada



Índices baseado em lista encadeada

- As listas para acesso a chave secundária podem ser construídas quando se abre o arquivo de dados,
  - ou pode-se construir a partir de arquivos de índices que já existam para cada uma das chaves secundárias.
- A vantagem da segunda solução é que esta já possui os valores ordenados e permite que,
  - se necessário apenas parte dos valores dessa chave secundária sejam trazidos para a memória.
- A desvantagem dessa solução é a manutenção das estruturas e índices na memória externa.

Organização de Arquivos com Lista Circular

- Este método está orientado a um eficiente processamento de subconjuntos de registros.
- Um subconjunto é definido como um grupo de registros,
  - os quais contém algum valor de atributo comum,
  - por exemplo, todos os alunos que cursam a disciplina organização de arquivos.
- Subconjuntos de registros são explicitamente mantidos juntos através do uso de ponteiros,
  - os quais definem alguma ordem para os membros do subconjunto.

Organização de Arquivos com Lista Circular

- Implementação com listas circulares:
  - Uma lista circular para cada chave secundária que conteria os valores que cada chave possui e um apontador para a lista circular com os endereços dos registros que possuem este valor de chave.

Valor	Ponteiro p/ endereços	Ponteiro p/ próximo valor

Momento 2: Mapeamento de Bits

Mapeamento de Bits

- Esta técnica é útil quando
  - a chave secundária utilizada possui um conjunto bem definido de valores,
  - de forma que possam ser mapeados para bits.
- Além disso,
  - esta técnica permite uma eficiente realização de consultas booleanas que envolvam a utilização de chaves secundárias.

Mapeamento de Bits

- Para mapear-se os valores de chaves secundárias em bits procede-se da seguinte maneira:
  - Verifica-se a quantidade máxima de valores N que uma determinada chave primária K pode possuir.
  - O Mapa de Bits da chave K terá N bits, cada um correspondendo a um valor de chave.
  - Cria-se um nodo para cada valor de bits acrescentando-se o endereço no arquivo de dados.

Mapeamento de Bits

- Sugere-se utilizar conjuntos de 8 bits (ou seja, N deve ser múltiplo de 8).
- Entretanto,
  - esta pode não ser a realidade da maior parte das chaves secundárias que serão mapeadas para bits
  - e desta forma frequentemente teremos a utilização de bits ociosos.
- Para resolver este problema
  - pode-se concatenar todos bits que são utilizados para mapear todas as chaves secundárias de um arquivo,
  - obtendo desta forma um melhor aproveitamento da estrutura.
  - Para tanto, deve-se saber quais bits representam os valores de cada chave secundária.

Mapeamento de Bits

- Este armazenamento
  - concatenado das chaves secundárias
  - permite a realização de comparações booleanas com um único teste.

Mapeamento de Bits

- Exemplo uma aplicação contêm as seguintes informações:
  - Código,
  - Nome,
  - Cor dos olhos (preto, castanho, azul, verde, indefinido claro, indefinido escuro),
  - Cor dos cabelos (preto, ruivo claro, ruivo escuro, castanho
  - claro, castanho escuro, loiro),
  - Cor da pele (branca, preta, amarela, albino),
  - Estado civil (solteiro, casado, viúvo, separado, divorciado, ajuntado),
  - Nacionalidade (brasileira, estrangeira),
  - Idade.

Mapeamento de Bits

- O campo código é a chave primária,
- Os campos Nome e Idade são atributos não chave e os outros campos são chaves secundárias.
- O índice de inversão possuirá 24 bits, assim dispostos:
  - Cor dos olhos: 6 bits
  - Cor dos cabelos: 6 bits
  - Cor da pele: 4 bits
  - Estado civil: 6 bits
  - Nacionalidade: 2 bits

Mapeamento de Bits

- Cada bit correspondendo a um valor possível na chave secundária.
- Assim, se uma pessoa tivesse
  - olhos verdes,
  - cabelos loiros,
  - pele branca,
  - fosse casada e brasileira
  - possuiria a seguinte sequencia de bits associada a seu registro:
  - 000100 0000011000001000010.

Exercício Mapeamento de Bits

- Defina os valores binários para:
  - Cor dos olhos (preto, castanho, azul, verde, indefinido claro, indefinido escuro),
  - Cor dos cabelos (preto, ruivo claro, ruivo escuro, castanho
  - claro, castanho escuro, loiro),
  - Cor da pele (branca, preta, amarela, albino),
  - Estado civil (solteiro, casado, viúvo, separado, divorciado, ajuntado),
  - Nacionalidade (brasileira, estrangeira)

Exercício Mapeamento de Bits

- Gere o código de mapeamento de bits para:
  - Olhos pretos, cabelos pretos, pele branca, solteiro, estrangeira
  - Olhos azuis, cabelos pretos, pele branca, solteiro, brasileiro,
  - Olhos azuis, cabelos ruivo, pele escura, casado, brasileiro
  - Crie mais uns 15 exemplos
- Crie uma lista encadeada com estes valores

Momento 3: Exercícios

Problematização

Qual é a resposta ?

“Como as listas encadeadas podem ser auxiliar a busca por informações armazenadas em arquivos de dados físicos?”

Exercícios

1. Porque utilizar estruturas de listas como índices?
2. A estrutura de índice deve ser manipulada em arquivo ou em memória?
3. Qual a relação entre o tamanho do arquivo e o tamanho do índice (e, por consequência, o tamanho de memória a ser utilizado)?
4. Qual o tipo de índice que deve ser utilizado em listas?
5. O uso de registros de tamanho fixo ou variável interfere na criação do índice baseado em lista?