03/02/2023

Dispositivos de Armazenamento:

* Uma imagem contendo Diagrama

  Descrição gerada automaticamenteMemoria primaria x secundaria:
* Obs: como as memorias secundarias são não volateis é possivel representar uma entidade do mundo nela , o que seja usada na diciplina
* Disco Rígido x SSD:
  + Disco 🡪 partes mecânicas, se movem com apoio de motores]
  + Linha do tempo

    Descrição gerada automaticamenteSSD 🡪 eletricamente, e uso do controlador que gerencia troca de dados
  + Algoritmos feitos na disciplina devem levar em consideração armazenamento de dados
  + Interface gráfica do usuário, Texto, Aplicativo, Carta, Email

    Descrição gerada automaticamenteCálculo de um disco rígido:
  + Tela de celular com texto preto sobre fundo branco

    Descrição gerada automaticamenteCálculo da operação em disco(tempo):

08/02/2023

* Leitura e escrita no Disco:
  + Cluster é um conjunto de setores, unidade mínima que o SO acessa
* Tempos de operação:
  + Acesso 🡪 cabeçote ser posicionado no setor desejado
  + Busca 🡪 posicionamento no cilindro correto
  + Transferência 🡪
  + Latência Rotacional 🡪

Manipulando a Memória Secundaria

* Arquivos como vetores de bytes:
  + O arquivo no disco pode ser visto como um vetor de bytes , com posições sequenciais iniciadas em 0
  + O ponto em que os dados serão lidos ou escritos é identificado por um ponteiro
  + Quaisquer valores a serem escritos no arquivo devem ser convertidos em uma sequência de bytes
  + Registro 🡪 sequência de bytes em um arquivo que contêm uma entidade
  + Campo 🡪 sequência de bytes em um arquivo que contêm um atributo

Aula 10/02/2023

Fluxos de entrada e saída

* Diferentes tipos de registros:
  + Registro de tamanho e campo fixo:
    - Desperdício de memória e perda de dados
    - Fácil de manipular(sempre sabe onde cada registro começa, fazendo saltos)
  + Registro de tamanho fixo e campo de tamanho variável
    - Tela de computador com texto preto sobre fundo branco

      Descrição gerada automaticamenteUtilizar delimitador
    - Desperdício ou falta de espaço
    - Quando strings não tem tamanho fixo e existe um delimitador
  + Registros de tamanho variável
    - Arquivo se ajusta ao tamanho necessário de cada campo
    - Tela de computador com texto preto sobre fundo branco

      Descrição gerada automaticamenteSomente o acesso sequencial aos registros é possível, porém o acesso aleatório pode ser feito por índices
  + Qual melhor tipo?
    - Depende do contexto 🡪 objetivo – menor n de operações e menor tamanho de arquivo

Aula dia 15/02

Arquivos Sequenciais:

* Sequencial: Apenas uma direção, começo ao fim em lida só
* Arquivos em que os registros são acessados na ordem em que estão armazenados
* Usados quando há poucas movimentações de registros
* Acesso rápido a um conjunto de registro
* Não são bons para acessos aleatórios
* Ordenação:
  + Momento da criação
  + Chave primária
  + Qualquer chave de ordenação
  + Nenhuma ordem
* Chaves de ordenação:
  + Eventualmente precisamos reordenar para manter a ordem
  + Normalmente arquivo sequencias são usados como arquivos temporários
  + Chave primária 🡪 chaves que identificam cada entidade de forma exclusiva
* Identificadores:
  + Ajudam a estabelecer os relacionamentos entre as entidades
* CRUD:
  + CREATE:
    - Insert , post/put
  + READ:
    - Select e get
  + UPDATE:
    - Update e put.patch
  + DELETE:
    - Delete
  + Lapide 🡪 marca de exclusão, campo de 1 byte que indica se registro foi excluído
  + Exclusão não é físico e sim lógico, ferramentas de recuperação utilizam dessa característica para recuperar dados
  + TP: lapide, tamanho , dados

Aula dia 17/02/2023

Ordenação externa

* Processo de ordenação de dados em arquivos
* Adotado quando os dados a serem ordenados serem maiores que a capacidade de ordenação em memória primaria
* Prioriza o acesso sequencial aos arquivos

Intercalação balanceada:

* Algoritmo que ordena os registros por meio da intercalação de registros de varias fontes balanceados
* Execução:
  + 1 Etapa 🡪 os registros serão lios em blocos de grande tamanho para memoria primaria(QuickSort , Mergesort), e escritos nos primeiros M caminhos, em diferentes arquivos
  + 2 Etapa🡪Compara primeiro registro dos caminhos , salvando em um outro arquivo ( em blocos de tamanho determinado) , faz isso continuamente e manda pra outro arquivo
  + Diagrama

    Descrição gerada automaticamenteSe a fonte for única parar
  + Complexidade:
    - É o numero de passadas por todos os registros
    - Calculo 🡪 tempo da ordenação em memoria primaria é pouco significativo
  + Texto

    Descrição gerada automaticamentePara mais eficiência:
    - Segmento de tamanho variável:
      * Blocos já ordenados, de tamanho diferentes
    - Substituição

Aula dia 24/02

Intercalação com seleção por substituição:

* É gerar segmentos ordenados maiores a fase de distribuição
* Requer o uso de uma fila de prioridades , como um heap
* Diagrama

  Descrição gerada automaticamentePesos diferentes: , caso o numero no arquivo é maior que o numero que vc quer ordenar
* A cada nível de peso será um segmento ordenado guardado em diferentes arquivos

Economia de arquivo temporário:

Texto

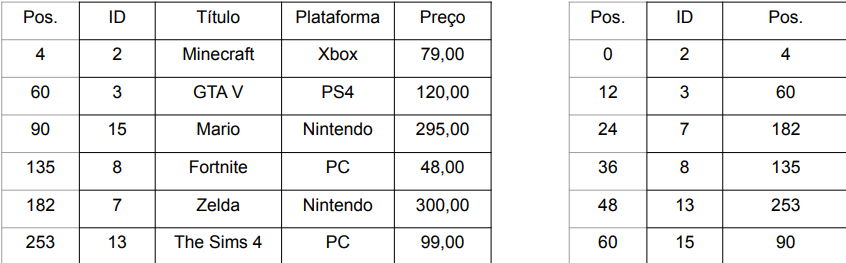
Descrição gerada automaticamenteEstudo Individual 25/02/2023

Arquivos Sequenciais:

* Acessados na ordem em que estão armazenados ( pouca ou nenhuma movimentação de registros)
* Chave de ordenação:
  + Conjunto de critérios que estabelece a ordem
* Identificadores:
  + Característica de um bom identificador:
    - Numérico 🡪 maior possibilidade
    - Sequencial 🡪 1,2,3,4...
    - Exclusivos 🡪 Não pode ter ambiguidade
    - Não significativos 🡪 Não pode contém valores significativos
* Area de extensão:
  + Otimiza o processo de inclusão quando o arquivo é muito grande
  + Combina acesso sequencial com lista encadeadas
  + Periodicamente o arquivo é reorganizado
  + Arquivo não pode ficar travado 🡪 acesso concorrente
* Ordenação Externa:
  + Usado quando a ordenação interna não da conta
  + Intercalação balanceada:
    - Algoritmo que ordena os registros por meio da intercalação de N caminhos balanceados – caminhos são as fontes
    - Etapas:
      * 1- Distribuição de blocos de registros em N caminhos
      * 2-Intercalação sucessivas dos N caminhos

01/03/2023 – Arquivo Indexado

* São arquivos em que os registros são acessados de forma aleatória. Para que a busca seja eficiente , esses arquivos contam com índices:



* Tipos de índices:
  + Primários ou secundários
    - Primário – Segue a mesma ordem do arquivo de dados
    - Secundário – Não seguem a mesma ordem
  + Diretos ou indiretos
    - Direto – Apontam diretamente para a posição do registro no arquivo de dados
    - Indireto – Apontam para um índice direto
  + Densos ou esparsos
    - Densos – possuem uma entrada para cada registro no arquivo de dados
    - Esparsos – possuem entradas para apenas alguns registros
* CRUD:
  + As operações realizadas em arquivos indexados são as mesmas realizadas nos arquivos sequenciais.
  + A principal diferença aqui, porém, é que as buscas por quaisquer entidades não serão feitas sequencialmente no arquivo dados, mas serão feitas em um dos índices.
  + Para que isso funcione corretamente, os índices devem sempre estar atualizados

Arvore B

* Tem altura menor que arvores binarias
* São baixas sendo necessário poucos acessos em disco até chegar ao ponteiro para o bloco
* Ideais para uso com índice de arquivos em disco
* Conceitos:
  + Ordem : diz qual o número máximo de filhos em cada página
  + Página : No da arvore B
* Regras da árvore B:
  + Cada página (exceto a raiz) deve ter pelo menos 50% de ocupação
  + O número de filhos de cada página (exceto as folhas) deve ser igual ao seu número de chaves mais um
  + Todas as folhas estão no mesmo nível (a árvore cresce para cima).Diagrama

    Descrição gerada automaticamente

Aula dia 03/03/2023 – Arvore B

* Tela de celular com texto preto sobre fundo branco

  Descrição gerada automaticamente com confiança médiaRemoção:

Aula dia 15/03/2023 – hashing dinamico

* Um hash dinâmico é uma estrutura de dados que usa tabela hash para armazenar de forma dinâmica
* Tabela hash dinâmico pode cescer ou diminuir de tamanho conforme necessário para elementos adicionados e removidos 🡪 índice precisa ser ajustado
  + Apenas elementos do bucket precisa ser reposicionados
* Inserindo chaves
  + Caso bucket esteja cheio🡪 aumentamos a profundidade e criamos ponteiros extras para os que não explodiram, assim duplicando o diretório
* Deletando chaves:
  + Precisamos reduzir o total de elementos para caber em somente 1 bucket
* Conclusões:
  + O diretório cresce, sem precisarmos reposicionar todos os registros (do índice)
  + O índice (lista de buckets) cresce de acordo com a necessidade
  + Como não há encadeamento dos buckets, não há perda de eficiência

Multilistas e listas invertida

* Indices secundários:
  + Vários problemas apresentam chaves com valores idênticos(eles não distinguem unicamente os registros)
  + As buscas resultam em subconjuntos de registros

Multilistas:

* Encadear registros que possuem mesmo índices secundários
* Indice invertido 🡪é um índice em que uma parte do conteúdo de um registro

Aula 17/03/2023 – Inserção e Remoção Arvore B

* Tela de computador com texto preto sobre fundo azul

  Descrição gerada automaticamente com confiança médiaRepresentação de arvore B em arquivo:

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente29/03/2023 – Compressão de dados - Introdução

* Objetivo: reduzir armazenamento e velocidade de transmissão
* Antes da compressão:
  + Eliminar itens redundantes
  + Uso de notação codificada
  + Codificação de texto
  + Supressão de espaços
* Classificação:
  + Simetria – mesmo esforço 🡪 mesmo tempo (tele medicina) – origem e destino mesmo peso
  + Perda – ocorre perdas de dados 🡪 mp3
  + Adaptabilidade -- Tabela própria que caracteriza do dado, dados recebidos não afetam codificador e semi adaptativo
* Tipos de compressão de dados:

Sem perdas:

* + Permite a recuperação exata dos dados originais após o processo
  + Remoção das redundâncias
  + Aplicadas a dados, textos, programas

Com perdas:

* Transformada Discreta do Cosseno
* Processamento de imagens

Compressão – Símbolos:

* Redução da Quantidade de Simbolos 🡪 Um símbolo passa e representar um conjunto de outros símbolos

Codificação RLE:

* Sem perdas
* Sequência longa de valores iguais é substituído por sua contagem
* Usa em dados com muitas repetições

Métodos Estatísticos:

* Utilizam códigos de comprimento variável
* Huffman

Métodos de Dicionário:

* Os símbolos serão substituídos por códigos a partir de um dicionário
* LZ77

31/03/2023 – RLE e Métodos estatísticos

* RLE
  + Nome: Run Lenth Enconding
  + Adequado para certas sequencias binarias
  + Simples rápido pra compressão e descompressão
  + Tipos de imagens :
* Interface gráfica do usuário, Texto, Aplicativo, Email

  Descrição gerada automaticamenteMétodos Estatísticos:
  + Utilizam códigos de comprimento variáveis
  + **Entropia** : Menor qntd de bits por símbolos para guardar o conteúdo de informação da fontes , ou seja, limite para compressão
  + Tabela

    Descrição gerada automaticamenteTexto

    Descrição gerada automaticamenteEX:
  + **Tamanho médio por simbolo**:
  + Texto

    Descrição gerada automaticamenteTexto

    Descrição gerada automaticamenteShannon-Fano – Algoritimo – não gera arvore ideal igual huffman:
  + EX: Diagrama

    Descrição gerada automaticamente
  + Texto

    Descrição gerada automaticamenteHuffman:

14/04/2023

LZ78:

* Tabela

  Descrição gerada automaticamenteMétodo de dicionário
* O que fazer quando dicionário ficar cheio?
  + Congelar dicionário:

19/04/2023 – Casamento de padrões

* Objetivo: Suponha que queremos encontrar uma frase em um texto , porém usamos para várias coisas : DNA , VOZ , PLAGIO...
* Arvore TRIE – AEDSII
  + Pesquisa digital na trie – até que o boolean esteja ativo, ou seja , acabou a palavra
* Arvore PATRICIA – AEDSII
  + Otimizar buscas em arquivos em bibliotecas
  + Diferença da trie: Usamos palavras ou invés de letra a letra , patrícia eh mais eficiente e gasta menos nós
  + A patrícia não temos a necessidade de manter informações que não contem nada
* Diferenças:
  + TRIE são mais simples e faciesi de implementar , mas alto consumo de memoria
  + PATRICIA mais eficientes em termos de espaço

Podemos criar uma arvore digital, porém nem sempre essa estratégia vai funcionar e ser a mais eficiente.

* Algoritmos – Casamento exato de padrões:
  + Força Bruta – tentativa um por um
  + KMP – autômato finitio não determinístico – onde iremos voltar caso yenha falha , ou seja , não volta do 0
  + Aho Corasick
  + Boyer Moore – Ler de tras para frente fazendo saltos
* Algoritmos – Casamento aproximado de padrões:
  + Distância de Levenshtein

Professor ficou de decidir se irá ou não aprofundar/cobrar PATRICIA e TRIE

**Força Bruta**

* Casamento/comparação bit a bit até encontrar falha ou padrão, usa shift para mudar o inicio
* Se tiver falha: 🡪 depende da implementação
  + Volta inicio
  + Continue
  + Faz nada
* Se achar temos que escolher se continua ou para
* Melhor caso 🡪 padrão achado em primeiro, ou seja, paga o tamanho do padrão
* Pior caso 🡪 Não achou ou achou em último

26/04/2023

KMP

* Funcionamento:
  + Construa a tabela de prefixo qe guarda informação sobre padrão
  + Varre a string esq pra dir comparando com padrão
  + Se falha , use tabela para determinar o próximo índice do padrão a ser comparado
  + Repita os passos
* Texto, Carta

  Descrição gerada automaticamenteConstrução da tabela:
  + De forma iterativa e usa o próprio padrão para determinar o valor de casa posição
  + Para cada posição i da tabela o valor é determinado pelo tamanho do maior prefixo da substring que termina na posição i
  + Em outras palavras, a tabela de prefixos guarda informações sobre o padrão que podem ser usadas para evitar comparações redundantes durante a busca.
  + Padrão sera marcada

28/04/2023

* Diagrama

  Descrição gerada automaticamenteTabela

  Descrição gerada automaticamenteComparação entre algoritmos:
* Boyer- Moore:
  + Comparações de caracteres são feitas da direita p esquerda 🡪 busca de um sufixo bom
  + Interface gráfica do usuário, Texto, Aplicativo, Word

    Descrição gerada automaticamenteCaracterísticas:

26/05/2023 – Introdução a criptografia

* Conversão de texto para código
* Confidencialidade, Integridade de dados , autenticação e Não repúdido
* Diagrama

  Descrição gerada automaticamenteCifra de César:
* Cifra de Vigenére:
* Diagrama

  Descrição gerada automaticamenteTexto

  Descrição gerada automaticamenteCifra de transposição:
* Tabela

  Descrição gerada automaticamenteCifra das colunas:
* Tipos de cripotografia:
  + Simetrica 🡪 mesma chave ciptografia e decifragem
    - Fluxo 🡪 bit a bit
    - Bloco 🡪 feita em blocos
  + Assimetrica 🡪 chaves diferentes
* One time Pad:
  + A chave é uma string de bits aleatórios do msmo tamanho a ser criptografado
  + É inquebrável se for aleatório e chave mantida em segredo
  + Normalmente usa-se combinações de cifras