

Trabalho Prático 01: Manipulação e Organização de Arquivos de Dados

Vivian Cristina Santana de Souza Lopes

Disciplina: Algoritmos e Estruturas de Dados II

Professor: Rafael Alexandre

1. Introdução

O armazenamento eficiente de dados é um dos pilares fundamentais em sistemas computacionais que lidam com grandes volumes de informação. Em particular, a forma como os registros são organizados e estruturados fisicamente em disco influencia diretamente o desempenho das operações de leitura, escrita e recuperação de dados. Estratégias como definição do tamanho dos registros, particionamento em blocos e métodos de alocação impactam tanto o uso do espaço quanto a velocidade de acesso.

Este trabalho prático tem como objetivo simular o processo de persistência de registros em um arquivo binário (**.DAT**), explorando restrições reais de armazenamento, como o uso de blocos de tamanho fixo e diferentes formas de organização dos registros. A proposta envolve implementar e comparar estratégias de armazenamento utilizando registros de tamanho fixo e registros de tamanho variável, considerando ainda variações como armazenamento contíguo ou espalhado entre blocos.

Ao desenvolver a simulação, busca-se proporcionar uma compreensão prática das implicações que as decisões de projeto como o layout dos registros, o tamanho dos blocos e a técnica de alocação que exercem sobre a eficiência de armazenamento e sobre o uso da memória secundária. Dessa forma, o trabalho reforça conceitos fundamentais conectando teoria e prática por meio da análise e manipulação de arquivos de dados.

2. Objetivos

O trabalho tem como objetivos principais:

1. Implementar a leitura e gravação de registros em arquivos binários (.dat);
2. Simular o armazenamento em blocos com controle de bytes utilizados;
3. Permitir a escolha entre registros de tamanho fixo ou variável;
4. Implementar o espalhamento de registros quando o espaço no bloco for insuficiente;
5. Calcular métricas de desempenho como:
Quantidade total de blocos utilizados.
Percentual médio de ocupação dos blocos.
Número de blocos parcialmente utilizados.
Eficiência total do armazenamento.

3. Metodologia

O programa foi desenvolvido na linguagem C, utilizando estruturas de dados (struct) para representar os registros de alunos e os blocos de armazenamento.

Cada registro é composto pelos seguintes campos:

Campo	Matricula	Tipo
Matricula	Inteiro (9 dígitos)	Identificador unico aluno
Nome	String (até 50 caracteres)	Nome do aluno
Cpf	String (11 caracteres)	CPF do aluno
Curso	String (até 30 caracteres)	Curso em que o aluno está matriculado
Filiação Mãe	String (até 30 caracteres)	Nome completo da mãe
Filiação Pai	String (até 30 caracteres)	Nome completo do pai
Ano de ingresso	Inteiro (4 dígitos)	Ano de entrada na instituição
CA	Float (2 casas decimais)	Coeficiente Académico

A estrutura em C foi definida:

```
typedef struct {

    int matricula;

    char nome[50];

    char cpf[12];

    char curso[30];

    char mae[30];

    char pai[30];

    int ano_ingresso;

    float coeficiente;
```

} Aluno;

Os registros são gerados automaticamente por meio da função gerarAluno(), que atribui valores fictícios a cada campo.

Durante a execução, o usuário informa:

1. A quantidade total de registros.
2. O tamanho máximo do bloco (em bytes).
3. O tipo de registro (fixo ou variável).
4. A permissão para espalhamento (quando aplicável por meio do registro variado).

Os dados são armazenados em blocos simulados, gravados sequencialmente em arquivo binário (alunos.dat).

Após a escrita, o programa realiza a leitura do arquivo e exibe os dados armazenados, bem como as estatísticas de ocupação dos blocos.

5. Implementação

O programa foi implementado em um único arquivo em C, contendo as seguintes funções principais:

1. gerarAluno() : gera automaticamente os registros fictícios;
2. tamanhoRegistro() : calcula o tamanho de cada registro;
3. armazenar() : simula a gravação nos blocos e calcula as estatísticas;
4. lerArquivo() : realiza a leitura e exibição dos dados armazenados no arquivo binário.

A simulação apresenta no terminal todas as informações exigidas pelo roteiro do trabalho:

1. Número total de blocos utilizados;
2. Percentual médio de ocupação dos blocos.
3. Blocos parcialmente utilizados;
4. Eficiência total do armazenamento;
5. Mapa textual de ocupação, representado por barras “#”.

5. Resultados e Análises

Teste 1 – Registros Fixos

```

C:\Users\vivian\Downloads\A x + -
TP1 AEDS II - Armazenamento em Blocos
Quantidade de registros: 10
Tamanho maximo do bloco (em bytes): 512
Tipo de registro (1 - Fixo | 2 - Variavel): 1

---- Armazenamento ----

===== ARMAZENAMENTO =====
Total de registros: 10
Blocos utilizados: 4
Blocos parcialmente usados: 4
Tamanho do bloco: 512 bytes
Total de bytes: 1536
Eficiencia total de armazenamento: 74.22%
Ocupacao media dos blocos: 74.22%

===== MAPA DE OCUPACAO DOS BLOCOS =====
Bloco 1: 100.00% cheio | #####
Bloco 2: 100.00% cheio | #####
Bloco 3: 100.00% cheio | #####
Bloco 4: 74.22% cheio | #####
Armacenamento concluido!

```

Resultados:

Total de blocos: 4

Eficiência total: 74,22%

Ocupação média dos blocos: 74,22%

O armazenamento distribuiu os registros em quatro blocos, sendo os três primeiros totalmente ocupados e o último parcialmente utilizado.

Isso demonstra que o uso de registros de tamanho fixo pode gerar fragmentação interna, reduzindo a eficiência de uso do espaço.

Teste 2 – Registros Variáveis (Sem Espalhamento)

```

C:\Users\vivian\Downloads\A x + -
---- TP1 AEDS II - Armazenamento em Blocos ----
Quantidade de registros: 10
Tamanho maximo do bloco (em bytes): 512
Tipo de registro (1 - Fixo | 2 - Variavel): 2
Permitir espalhamento (1 - Sim | 0 - Nao): 0

---- Armazenamento ----

===== ARMAZENAMENTO =====
Total de registros: 10
Blocos utilizados: 1
Blocos parcialmente usados: 1
Tamanho do bloco: 512 bytes
Total de bytes: 473
Eficiencia total de armazenamento: 92.38%
Ocupacao media dos blocos: 92.38%

===== MAPA DE OCUPACAO DOS BLOCOS =====
Bloco 1: 92.38% cheio | #####
Armacenamento concluido!
Process returned 0 (0x0) execution time : 9.249 s

```

Resultados:

Total de blocos: 1

Eficiência total: 92,38%

Ocupação média dos blocos: 92,38%

Cada registro possui tamanho variável, mas o armazenamento só grava registros inteiros dentro de um bloco.

Como resultado, obteve-se alta eficiência com mínima fragmentação interna e apenas um bloco parcialmente ocupado.

Teste 3 – Registros Variáveis (Com Espalhamento)

```
C:\Users\vivian\Downloads>A ... + ...  
---- TPI AEDS II - Armazenamento em Blocos ----  
Quantidade de registros: 20  
Tamanho maximo do bloco (em bytes): 512  
Tipo de registro (1 - Fixo | 2 - Variavel): 2  
Permitir espalhamento (1 - Sim | 0 - Nao): 1  
  
--- Armazenamento ---  
Registro 1 espalhado: 39 bytes no bloco 1 e 11 bytes no bloco 2  
  
===== ARMAZENAMENTO =====  
Total de registros: 20  
Blocos utilizados: 2  
Blocos parcialmente usados: 1  
Tamanho do bloco: 512 bytes  
Total de bytes: 973  
Eficiencia total de armazenamento: 95.02%  
Ocupacao media dos blocos: 95.02%  
  
===== MAPA DE OCUPACAO DOS BLOCOS =====  
Bloco 1: 100.00% cheio | #####  
Bloco 2: 95.02% cheio | #####  
  
===== LEITURA DO ARQUIVO BINARIO =====  
Matricula: 100000001 | Nome: Aluno_1 | CPF: 00000000001 | Curso: Curso_1 | Mae: Mae_1 | Pai: Pai_1 | Ano: 2021 | Coef: 6.10  
Matricula: 100000002 | Nome: Aluno_2 | CPF: 00000000002 | Curso: Curso_2 | Mae: Mae_2 | Pai: Pai_2 | Ano: 2022 | Coef: 8.70  
Matricula: 100000003 | Nome: Aluno_3 | CPF: 00000000003 | Curso: Curso_3 | Mae: Mae_3 | Pai: Pai_3 | Ano: 2023 | Coef: 7.40  
Matricula: 100000004 | Nome: Aluno_4 | CPF: 00000000004 | Curso: Curso_4 | Mae: Mae_4 | Pai: Pai_4 | Ano: 2024 | Coef: 8.00  
Matricula: 100000005 | Nome: Aluno_5 | CPF: 00000000005 | Curso: Curso_0 | Mae: Mae_5 | Pai: Pai_5 | Ano: 2020 | Coef: 6.90  
Matricula: 100000006 | Nome: Aluno_6 | CPF: 00000000006 | Curso: Curso_1 | Mae: Mae_6 | Pai: Pai_6 | Ano: 2021 | Coef: 6.40  
Matricula: 100000007 | Nome: Aluno_7 | CPF: 00000000007 | Curso: Curso_2 | Mae: Mae_7 | Pai: Pai_7 | Ano: 2022 | Coef: 9.80  
Matricula: 100000008 | Nome: Aluno_8 | CPF: 00000000008 | Curso: Curso_3 | Mae: Mae_8 | Pai: Pai_8 | Ano: 2023 | Coef: 9.80  
Matricula: 100000009 | Nome: Aluno_9 | CPF: 00000000009 | Curso: Curso_4 | Mae: Mae_9 | Pai: Pai_9 | Ano: 2024 | Coef: 6.20  
Matricula: 100000010 | Nome: Aluno_10 | CPF: 00000000010 | Curso: Curso_0 | Mae: Mae_10 | Pai: Pai_10 | Ano: 2020 | Coef: 8.40  
Matricula: 100000011 | Nome: Aluno_11 | CPF: 00000000011 | Curso: Curso_1 | Mae: Mae_11 | Pai: Pai_11 | Ano: 2021 | Coef: 8.50  
Matricula: 100000012 | Nome: Aluno_12 | CPF: 00000000012 | Curso: Curso_2 | Mae: Mae_12 | Pai: Pai_12 | Ano: 2022 | Coef: 8.50
```

Resultados:

Total de blocos: 2

Eficiência total: 95,02%

Ocupação média dos blocos: 95,02%

Parte de um registro que não coube no final de um bloco foi espalhada para o bloco seguinte.

Essa técnica aumentou a eficiência do armazenamento, reduzindo o desperdício de espaço e aproveitando melhor os blocos.

Teste 4 – Blocos Menores

```

===== ARMAZENAMENTO =====
Total de registros: 5
Blocos utilizados: 5
Blocos parcialmente usados: 5
Tamanho do bloco: 200 bytes
Total de bytes: 820
Eficiencia total de armazenamento: 82.00%
Ocupação media dos blocos: 82.00%

===== MAPA DE OCUPACAO DOS BLOCOS =====
Bloco 1: 100.00% cheio | #####
Bloco 2: 100.00% cheio | #####
Bloco 3: 100.00% cheio | #####
Bloco 4: 100.00% cheio | #####
Bloco 5: 82.00% cheio | #####
#### LEITURA DO ARQUIVO BINARIO #####
Matricula: 100000001 | Nome: Aluno_1 | CPF: 0000000001 | Curso: Curso_1 | Mae: Mae_1 | Pai: Pai_1 | Ano: 2021 | Coef: 6.10
Matricula: 100000002 | Nome: Aluno_2 | CPF: 0000000002 | Curso: Curso_2 | Mae: Mae_2 | Pai: Pai_2 | Ano: 2022 | Coef: 8.70
Matricula: 100000003 | Nome: Aluno_3 | CPF: 0000000003 | Curso: Curso_3 | Mae: Mae_3 | Pai: Pai_3 | Ano: 2023 | Coef: 7.40
Matricula: 100000004 | Nome: Aluno_4 | CPF: 0000000004 | Curso: Curso_4 | Mae: Mae_4 | Pai: Pai_4 | Ano: 2024 | Coef: 8.00
Matricula: 100000005 | Nome: Aluno_5 | CPF: 0000000005 | Curso: Curso_0 | Mae: Mae_5 | Pai: Pai_5 | Ano: 2020 | Coef: 6.90

```

Resultados:

Total de blocos: 5

Eficiência total: 82,00%

Ocupação média dos blocos: 82,00%

Com blocos de tamanho reduzido, observou-se um aumento no número total de blocos necessários para armazenar a mesma quantidade de registros.

Apesar da boa taxa de ocupação, blocos pequenos geram mais acessos e menor eficiência global, mostrando a importância do balanceamento no tamanho dos blocos.

6. Conclusão

Com base nos testes realizados, foi possível constatar que tanto o tamanho do bloco quanto o modo de armazenamento fixo, variável, com ou sem espalhamento influenciam diretamente a eficiência de utilização do espaço em disco.

Os resultados demonstraram que os registros de tamanho variável com espalhamento apresentaram o melhor aproveitamento dos blocos, reduzindo a fragmentação interna e otimizando o uso do espaço de armazenamento. Em contrapartida, os registros de tamanho fixo ou sem espalhamento resultaram em blocos parcialmente preenchidos, comprometendo a eficiência total do sistema.

Conclui-se que o uso de espalhamento controlado é uma estratégia eficaz para melhorar o aproveitamento dos blocos, especialmente em ambientes com registros de tamanhos variados. Além disso, o tamanho do bloco deve ser proporcional ao tamanho médio dos registros, de modo a equilibrar desempenho e eficiência. Assim, estratégias de armazenamento flexíveis revelam-se fundamentais para reduzir o desperdício de espaço e aumentar o desempenho global do sistema de armazenamento.