

Associative Arrays

Giovanni Milan Câmara Pinto, João Henrique Alves Silva, e Samuel Alves Gomes
Palaoro

Universidade Vila Velha

2026-03-27

Resumo

A coluna apresenta os arrays associativos como uma estrutura de dados poderosa que permite a indexação por chaves, ao invés de índices numéricos. Essa abordagem simplifica problemas como contagem de palavras, remoção de duplicatas, agregação de dados e verificação de pertencimento.

Por meio da linguagem Awk, Bentley demonstra que muitos problemas que exigiriam estruturas complexas em linguagens tradicionais podem ser resolvidos com poucas linhas de código. Além disso, a coluna apresenta aplicações mais avançadas, como a implementação de máquinas de estados finitos e a utilização de ordenação topológica, um algoritmo utilizado para organizar vértices de um grafo direcionado de forma linear.

Dessa forma, a coluna reforça a ideia de que a escolha adequada da estrutura de dados pode simplificar significativamente o desenvolvimento e melhorar a clareza do código.

Sumário

1	Introdução	2
2	Revisão bibliográfica	2
3	Metodologia	2
3.1	Definição do problema	2
3.2	Análise das abordagens tradicionais	2
3.3	Estrutura adequada	3
3.4	Implementação das soluções	3
3.5	Justificativa da eficiência	3
4	Conclusão	3

1 Introdução

A coluna inicia com uma introdução à linguagem AWK, que se destaca como uma ferramenta poderosa para processamento de texto, busca de padrões e geração de relatórios em sistemas **UNIX/Linux**. Um de seus principais diferenciais é o suporte a arrays associativos, que, ao contrário dos vetores tradicionais, permitem o uso de chaves arbitrárias, como strings, possibilitando soluções mais diretas e expressivas para diversos problemas computacionais.

Na coluna do livro *Programming Pearls*, de Jon Bentley, são apresentados exemplos do uso dessa estrutura de dados na linguagem AWK, demonstrando como ela pode simplificar tarefas que, em linguagens tradicionais, seriam mais complexas. Entre as aplicações destacam-se a contagem de palavras, a remoção de duplicatas, a agregação de dados e a verificação de pertencimento.

Além disso, a coluna explora aplicações mais avançadas, como a implementação de máquinas de estados finitos e algoritmos de ordenação topológica, evidenciando que a escolha adequada da estrutura de dados impacta diretamente na clareza e eficiência do código. Dessa forma, o estudo da linguagem AWK contribui para compreender como abstrações simples podem resolver problemas complexos de maneira elegante.

2 Revisão bibliográfica

Os arrays associativos são uma estrutura de dados amplamente utilizada em diversas linguagens modernas, sendo conhecidos por permitir o mapeamento entre chaves e valores. Na linguagem AWK, essa é a estrutura central, pois todos os arrays são associativos por padrão, dispensando declarações explícitas e inicializações manuais.

De acordo com Bentley, essa característica é ideal para resolver problemas clássicos. Um exemplo é a contagem de ocorrências de palavras em um texto, na qual cada palavra pode ser utilizada diretamente como índice do array. Da mesma forma, é possível verificar e remover duplicatas por meio da checagem da existência prévia de uma chave.

Além disso, a coluna demonstra o uso de arrays associativos na implementação de máquinas de estados finitos. Nesse contexto, estados e transições são representados por estruturas que mapeiam entradas e saídas, permitindo simular o comportamento computacional de forma simples e flexível.

Outro tema relevante abordado é a ordenação topológica de grafos direcionados. O algoritmo utiliza arrays associativos para armazenar o número de predecessores de cada nó e suas listas de sucessores, permitindo a construção de uma solução eficiente baseada em filas. Essa abordagem reduz a complexidade de implementação e mantém um desempenho adequado.

De maneira geral, a literatura destaca que o uso de estruturas de dados adequadas, como arrays associativos, pode reduzir a complexidade do código, facilitando o desenvolvimento e a manutenção, especialmente quando se utiliza uma representação apropriada para cada problema.

3 Metodologia

3.1 Definição do problema

A metodologia apresentada na coluna consiste em explorar como arrays associativos podem ser utilizados para resolver problemas clássicos da computação, com foco na simplificação da implementação sem comprometer a corretude dos algoritmos.

3.2 Análise das abordagens tradicionais

Problemas como contagem de palavras, remoção de duplicatas e processamento de grafos são frequentemente resolvidos, em linguagens tradicionais, com estruturas mais complexas, como listas encadeadas e tabelas hash. Embora eficientes, essas abordagens tendem a aumentar a complexidade do código.

3.3 Estrutura adequada

A coluna identifica que os arrays associativos da linguagem AWK são suficientes para representar diferentes estruturas abstratas, como tabelas de símbolos, filas e grafos, permitindo soluções mais diretas e com menor complexidade de implementação.

3.4 Implementação das soluções

As soluções são implementadas utilizando a sintaxe simples da linguagem AWK, destacando-se:

- Contagem de palavras, utilizando palavras como índices;
- Remoção de duplicatas por verificação de chaves;
- Simulação de máquinas de estados finitos, representando estados e transições por meio de arrays associativos;
- Ordenação topológica, utilizando contadores de predecessores e filas para processar grafos direcionados.

Nos casos de máquinas de estados finitos, os arrays permitem mapear diretamente entradas, estados e saídas, simplificando a simulação do sistema. Já na ordenação topológica, são utilizados para armazenar relações entre nós e controlar o processamento por meio de uma fila, garantindo uma solução eficiente.

3.5 Justificativa da eficiência

A eficiência das soluções está na redução da complexidade estrutural do código. O uso de arrays associativos elimina a necessidade de estruturas auxiliares mais elaboradas, reduzindo o número de linhas e a possibilidade de erros. Embora nem sempre seja a abordagem mais eficiente em termos de execução, sua simplicidade a torna adequada para prototipação e desenvolvimento rápido.

4 Conclusão

A análise da coluna evidencia que a escolha adequada da estrutura de dados pode simplificar significativamente a resolução de problemas computacionais. Nesse contexto, a linguagem AWK se destaca por oferecer suporte nativo a arrays associativos, permitindo a construção de soluções elegantes com código reduzido e de fácil compreensão.

Os exemplos apresentados, como a contagem de palavras, a simulação de máquinas de estados finitos e a ordenação topológica de grafos, demonstram como essa estrutura pode ser aplicada em diferentes níveis de complexidade. Em todos os casos, observa-se que a utilização de arrays associativos permite representar dados e relações de forma direta, reduzindo a necessidade de estruturas auxiliares mais complexas.

Entretanto, a principal contribuição da coluna está nos princípios apresentados ao final: a importância de escolher representações adequadas, priorizar a simplicidade e desenvolver soluções claras antes de buscar otimizações mais complexas. A linguagem AWK exemplifica bem essa abordagem, ao permitir que problemas sofisticados sejam resolvidos com programas curtos e expressivos.

Dessa forma, conclui-se que a eficiência de um programa não está apenas em seu desempenho, mas também na clareza, concisão e facilidade de desenvolvimento. A combinação entre boas abstrações, estruturas de dados adequadas e ferramentas apropriadas, como AWK, é fundamental para a construção de sistemas eficientes e compreensíveis.