

Resumo Conceitual: A Hash Table no Minishell

1. O que é uma Hash Table?

Uma Hash Table (ou Tabela de Dispersão) é uma estrutura de dados que associa **chaves** a **valores** (pares chave-valor). Ela utiliza uma **função de hash** para calcular um índice numérico a partir da chave. Esse índice corresponde a uma posição (ou "bucket") em um array, onde o valor é armazenado. O objetivo é permitir acesso, inserção e remoção de dados de forma extremamente rápida, idealmente em tempo constante ($O(1)$).

2. Qual a sua importância geral?

Sua principal importância é a **eficiência**. Enquanto procurar um item em uma lista ou array pode exigir a verificação de todos os elementos (complexidade $O(n)$), uma Hash Table permite encontrar um item quase que instantaneamente, independentemente do número de elementos armazenados. Isso a torna ideal para qualquer cenário que exija buscas rápidas, como bancos de dados, caches e implementações de dicionários.

3. Onde ela será usada no projeto?

No Minishell, a Hash Table será a estrutura central para **gerenciar as variáveis de ambiente**. As variáveis de ambiente do sistema (como `PATH`, `HOME`, `USER`, etc.) são naturalmente pares chave-valor. A tabela será utilizada diretamente pelas seguintes funcionalidades:

- * `export VAR=valor`: Inserir ou atualizar um par na tabela.
- * `unset VAR`: Remover um par da tabela.
- * `env`: Listar todos os pares armazenados na tabela.
- * **Expansão de variáveis** (`$VAR`): Buscar o valor associado a uma chave na tabela.

4. Qual a sua importância *para o projeto*?

A importância para o Minishell é garantir que o shell seja **rápido e responsivo**. Um shell precisa consultar variáveis de ambiente constantemente, principalmente a variável `PATH` para localizar comandos executáveis.

Se as variáveis fossem armazenadas em uma lista, cada vez que um comando fosse executado, o programa teria que percorrer essa lista para encontrar o `PATH`. Com uma Hash Table, essa busca é quase imediata ($O(1)$). Isso garante que o desempenho do

shell não se degrade à medida que mais variáveis são adicionadas ao ambiente, fornecendo uma base escalável e eficiente para uma das funcionalidades mais críticas de um shell.