



**INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA**
Acre

VINÍCIUS BARROS DE MELO

GIOVANNA SOUZA CORREIA

ÁRVORE GENÉRICA

Teoria

Rio Branco - AC

2025

VINÍCIUS BARROS DE MELO
GIOVANNA SOUZA CORREIA

ÁRVORE GENÉRICA

Teoria

Trabalho sobre árvore
genérica,
apresentado à
matéria de Estrutura
de Dados, como
requisito para
obtenção de nota de
atividade.

Orientador: Prof.
Breno

Rio Branco - AC

2025

Árvore Genérica em C

Introdução

Uma árvore genérica é uma estrutura de dados em que cada nó pode ter um número variável de filhos, ao contrário das árvores binárias, onde cada nó pode ter, no máximo, dois filhos. Essa flexibilidade torna as árvores genéricas úteis para representar hierarquias complexas em sistemas, como diretórios de arquivos, organizações empresariais e outras estruturas que envolvem relações entre elementos de forma mais dinâmica.

Em uma árvore genérica, os nós são organizados de maneira hierárquica, com a possibilidade de cada nó ter múltiplos filhos, formando uma árvore com ramificações de profundidade variável. Essa estrutura pode ser representada utilizando ponteiros, onde cada nó tem um ponteiro para seus filhos e para seus irmãos, permitindo que cada nó tenha qualquer número de filhos.

Estrutura de Dados

Cada nó de uma árvore genérica geralmente armazena dois componentes principais: os dados que ele contém e referências (ponteiros) para seus filhos e irmãos. A referência para os filhos é utilizada para indicar o primeiro filho do nó, enquanto a referência para os irmãos é usada para apontar para o próximo nó que compartilha o mesmo pai.

Componentes da Estrutura

Valor: O nó pode armazenar dados. No caso de uma árvore genérica, esse valor pode ser qualquer tipo de dado, como inteiros, strings, objetos, etc.

Ponteiro para o filho: Cada nó tem um ponteiro que aponta para o primeiro filho. Se o nó não tiver filhos, esse ponteiro é NULL.

Ponteiro para o irmão: Cada nó também possui um ponteiro para o próximo irmão. Se não houver irmãos, o ponteiro será NULL.

Operações Comuns em Árvores Genéricas

Criação de Nó

A criação de um nó em uma árvore genérica envolve a alocação de memória dinâmica para armazenar o valor e os ponteiros para filhos e irmãos. Um nó recém-criado geralmente começa com seus ponteiros configurados como NULL, pois ele ainda não tem filhos ou irmãos.

Inserção de Filho

A inserção de um filho é feita alterando o ponteiro filho do nó pai. Se o nó pai já tiver filhos, o novo filho será inserido como o primeiro filho, ou como o último irmão, se o nó já tiver outros filhos.

Percorrendo a Árvore (Travessia)

A travessia (ou percurso) de uma árvore genérica pode ser feita de diferentes maneiras. Uma das formas mais comuns é a travessia pré-ordem, onde o nó é processado antes de seus filhos e irmãos. A travessia de cada nó é feita de maneira recursiva, visitando primeiro o nó atual, depois seus filhos, e por fim seus irmãos.

Destruição de Nó

A destruição de um nó envolve liberar a memória alocada para ele e seus filhos, utilizando um processo recursivo. Ao destruir a árvore, todos os nós e suas respectivas alocações de memória são liberados para evitar vazamentos de memória.

Vantagens da Árvore Genérica

Flexibilidade: Uma árvore genérica permite que um nó tenha qualquer número de filhos, o que a torna adequada para representar dados em estruturas onde a quantidade de filhos pode variar bastante entre os nós.

Hierarquias Complexas: Árvores genéricas são perfeitas para representar estruturas hierárquicas complexas, como diretórios de arquivos ou qualquer organização onde os elementos têm uma relação de "pai para filho", mas o número de filhos pode ser variável.

Simplicidade na Implementação: Embora a árvore genérica seja mais flexível que outras estruturas de árvore (como a árvore binária), sua implementação é

simples e pode ser adaptada para diferentes tipos de dados e necessidades.

Conclusão

As árvores genéricas são uma estrutura de dados poderosa e versátil, especialmente quando se precisa lidar com dados hierárquicos e flexíveis. Elas são comumente usadas em sistemas de arquivos, em modelos organizacionais ou em qualquer situação que envolva um número variável de filhos para cada elemento. Sua implementação é relativamente simples, e sua flexibilidade permite representar relações complexas de forma eficiente e eficaz.