

Análise da linguagem Rust

Integrantes:

- Oseias Romeiro 211036123
- Weliton Cesar 200044532
- Marcus Paulo 200041096



Introdução

- Rust é uma linguagem moderna que combina segurança de memória, abstrações com pouco custo, simultaneidade e segurança de memória.
- Nesta apresentação, vamos analisar os recursos e as vantagens do Rust, compará-lo com seu principal concorrente C e fornece exemplos de códigos escritos

Projeto

- Desenvolvimento de um jogo de xadrez utilizando a linguagem Rust
- Comprovando o alto desempenho da linguagem programando um algoritmo de busca em largura, o qual irá buscar a melhor jogada em um determinado tempo, quanto mais rápida for a implementação mais precisa será a jogada, assim podendo ter uma ideia sobre a capacidade da linguagem.

História

- Rust foi criado por Graydon Hoare em 2010, originalmente como um projeto pessoal. Foi lançado publicamente em 2015 e, desde então, tem visto um rápido crescimento e adoção. Sendo adotada por diversas empresas, como no GitHub, Cloudflare e até mesmo no Android (Google).

Características

- Rust é uma linguagem tipada estaticamente, o que significa que um tipo é verificado em tempo de compilação. Isso ajuda a garantir que o código esteja correto e simplifica o processo de depuração.
- Além disso, o Rust possui vários recursos que o tornam uma opção atraente para os desenvolvedores, incluindo:
 - Segurança de memória
 - Abstrações com pouco custo
 - Desempenho
 - Paralelismo

Segurança de memória

- Rust evita erros de memória, como estouros de buffer e ponteiros pendentes, que são fontes comuns de vulnerabilidades de segurança.
- Verificação de código em tempo de compilação para garantir que todos os acessos à memória sejam válidos. Como resultado, o Rust pode impedir o acesso à memória inválida e evitar corridas de dados, o que significa que os programas Rust têm muito menos probabilidade de ter erros de memória do que os programas C.

```
int main() {  
    int* ptr = malloc(sizeof(int));  
    free(ptr);  
    *ptr = 20;  
    return 0;  
}
```

```
fn main() {  
    let mut x = Box::new(10);  
    *x = 20; // lança um erro  
}
```

Abstração

- Rust não incorre na sobrecarga associada a outras linguagens para abstrações como classes, genéricos e funções virtuais, tornando-o mais eficiente.

```
enum Peca {  
    Cavalo(Cor),  
    Peao(Cor),  
    Dama(Cor),  
    Rei(Cor),  
    Torre(Cor),  
    Bispo(Cor),  
    Vazio,  
}
```

```
match self {  
    Peca::Torre(Cor::Preto) => "\u{2656}",  
    Peca::Cavalo(Cor::Preto) => "\u{2658}",  
    Peca::Bispo(Cor::Preto) => "\u{2657}",  
    Peca::Dama(Cor::Preto) => "\u{2655}",  
    Peca::Rei(Cor::Preto) => "\u{2654}",  
    Peca::Peao(Cor::Preto) => "\u{2659}",  
    Peca::Torre(Cor::Branco) => "\u{265C}",  
    Peca::Cavalo(Cor::Branco) => "\u{265E}",  
    Peca::Bispo(Cor::Branco) => "\u{265D}",  
    Peca::Dama(Cor::Branco) => "\u{265B}",  
    Peca::Rei(Cor::Branco) => "\u{265A}",  
    Peca::Peao(Cor::Branco) => "\u{265F}",  
    Peca::Vazio => " ",  
}
```

Desempenho

- Rust é projetado para ser rápido e é frequentemente comparado a C e C++ em termos de desempenho.
- Em termos de desempenho, Rust e C são bastante semelhantes. O Rust geralmente supera o C quando se trata de velocidade de execução, mas o C ainda é geralmente mais rápido para aplicativos de thread único. Além disso, as abstrações de baixo custo do Rust, tornam mais fácil para os desenvolvedores escrever código de alto desempenho.

Usabilidade

- Rust é uma linguagem versátil e pode ser usada para uma variedade de aplicações, que podem incluir:
- Desenvolvimento Web: Rust pode ser usado para criar servidores Web e serviços de back-end
- Sistemas embarcados: Rust pode ser usado para criar firmware e sistemas embarcados, como sistemas robóticos e veículos autônomos.
- Desenvolvimento de jogos: Rust pode ser usado para criar mecanismos de jogos e lógica de jogos

Participação

- Oseias Romeiro
 - desenvolvimento do jogo
 - criação de apresentação
 - análise da linguagem
- Weliton Cesar
 - Desenvolvimento do jogo
- Marcus Paulo
 - Apresentação inicial do projeto

Conclusão

- Rust é uma linguagem de programação cada vez mais popular que é usada para uma variedade de aplicações. É uma linguagem segura, rápida e confiável, além de ter vários recursos que ajudam os desenvolvedores.
- Apesar de não termos conseguido concluir o projeto, foi de grande proveito o estudo da linguagem e sua aplicação e análise feita com base no conhecimento adquirido durante a disciplina.
- Projeto (GitHub): <https://github.com/oseias-romeiro/xadrez-LP/>