



PlaneLang – Apresentação APS



PlaneLang – Linguagem de Comandos de Drone

Autor

Helio Henrique

Disciplina

APS – Linguagens e Paradigmas

A Inspiração por Trás do PlaneLang

A motivação para o desenvolvimento do PlaneLang surgiu da necessidade de uma ferramenta simplificada, porém robusta, para a interação com drones.

Controle Simplificado

Criação de uma linguagem acessível para comandos de drone, focada em aspectos essenciais como altitude, velocidade, direção e posição.

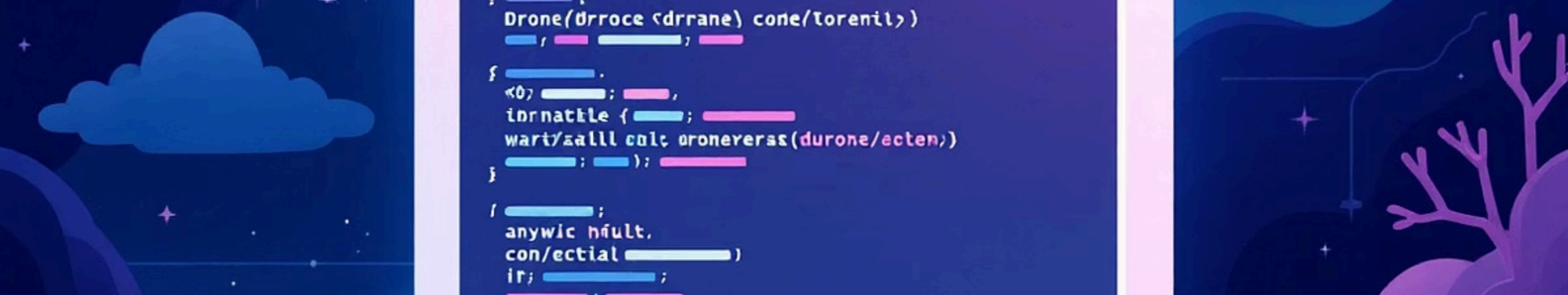
Simulação Aprimorada

Facilitar a simulação de voo e testes de algoritmos sem a dependência de hardware físico, tornando o desenvolvimento mais seguro e eficiente.

Aprendizado Profundo

Aplicação prática de conceitos de compiladores, incluindo parsing, análise léxica e a construção de uma Máquina Virtual (VM) customizada.





Características da Linguagem PlaneLang

PlaneLang foi projetada para ser intuitiva e poderosa, oferecendo um controle preciso sobre o drone através de uma sintaxe clara e concisa.

1

Sintaxe de Comandos

Comandos básicos como `takeoff`, `land`, `set`, `goto`, `wait` e `print` para ações diretas e claras.

2

Controle de Fluxo

Suporte a estruturas condicionais `if/else` e loops `while` para lógicas de voo complexas e automatizadas.

3

Parâmetros Dinâmicos

Manipulação de `speed`, `altitude` e `heading` para ajustes finos e adaptativos durante o voo.

4

Comentários e Debug

Suporte a comentários `//` e o comando `print` para depuração e feedback do estado do drone.

Mecanismos Internos: A VM e o Compilador

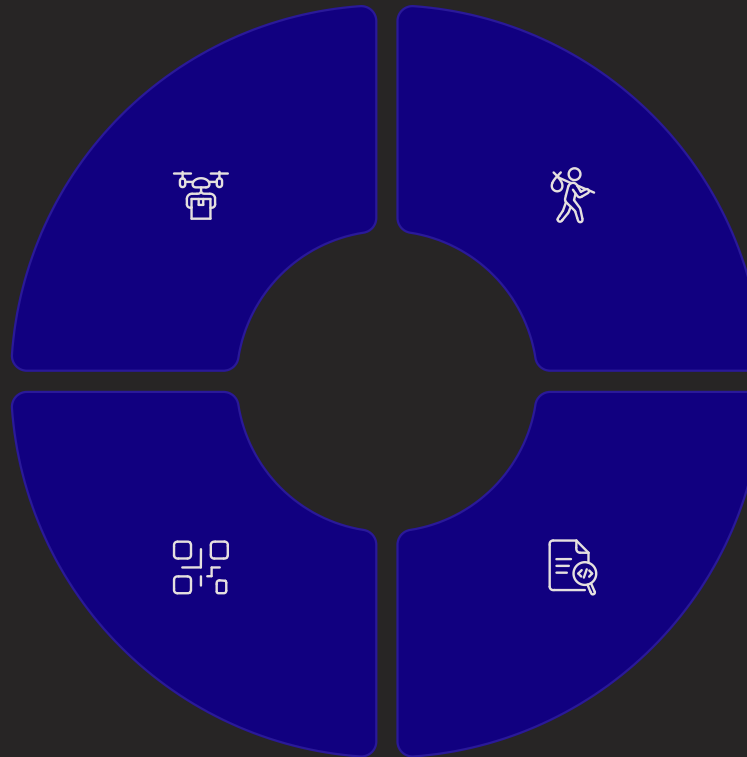
O coração do PlaneLang é sua Máquina Virtual (VM) dedicada, que permite a simulação e execução dos comandos em tempo real, integrando-se perfeitamente com o compilador.

Simulação de Drones

O PlaneLang foi especificamente desenvolvido para simular o comportamento de drones, com uma VM que replica o ambiente de voo.

Exemplo de Funções VM

Funções como `vm_execute_takeoff()`, `vm_execute_goto()` e `vm_execute_print()` exemplificam a granularidade do controle da VM.



VM Própria

A Máquina Virtual mantém e atualiza o estado do drone, interpretando e executando comandos em tempo de execução, garantindo precisão na simulação.

Integração Lexer/Parser

Utiliza Flex e Bison para análise léxica e sintática, transformando o código PlaneLang em instruções que a VM pode entender.

Código PlaneLang em Ação (test.pln)

Este exemplo abrangente demonstra a aplicação de todos os comandos, estruturas de controle e sintaxes disponíveis no PlaneLang, proporcionando uma visão clara de sua funcionalidade.

```
// Arquivo de teste para a linguagem PlaneLang
// Testa todos os comandos, estruturas de controle e sintaxes.

// Comandos básicos
takeoff;
land;

// Comando 'set' com diferentes parâmetros e valores
set speed = 100;
set altitude = 5000;
set heading = some_heading_variable; // Usando um ID como valor

// Outros comandos com parâmetros
wait 10 seconds;
goto waypoint_alpha;
print "Iniciando verificação de sistemas.";

// Estrutura condicional IF
if (altitude < 1000) {
    print "Alerta: Baixa altitude!";
    set speed = 50;
}

// Estrutura condicional IF-ELSE
if (speed >= 150) {
    print "Velocidade de cruzeiro atingida.";
} else {
    set speed = 150;
    print "Acelerando para velocidade de cruzeiro.";
}

// Loop WHILE
while (heading != 180) { // Comentário dentro de um loop
    set heading = 180;
    wait 1 seconds;
}

// Testando outros operadores e condições
if (some_variable == 42) {
    goto final_approach;
}

if (another_var <= 10) {
    // Bloco vazio
}

;; // Testando ponto e vírgula extra (declaração vazia)

print "Plano de voo concluído.";
```

Saída da VM: A Simulação em Palavras

A Máquina Virtual traduz o código PlaneLang em uma sequência de eventos e mudanças de estado, que simulam o comportamento do drone. A saída detalhada abaixo ilustra cada passo da execução.



A mensagem "Programa válido (análise + execução OK)" confirma a integridade e o sucesso da simulação.

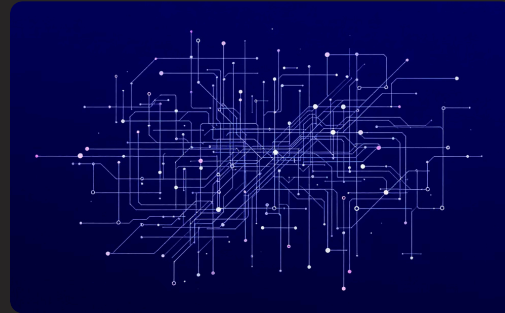
Estrutura do Projeto PlaneLang

O PlaneLang é organizado em um repositório estruturado, com módulos claros para cada fase do processo de compilação e execução.



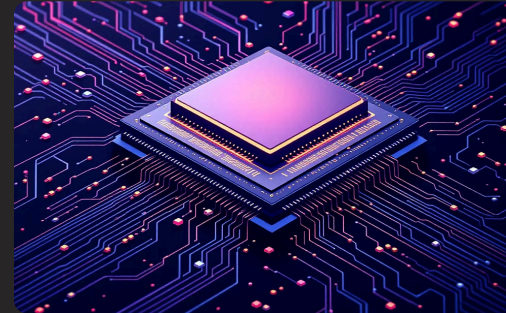
Lexer (lexer.l)

Responsável pela análise léxica, quebra o código-fonte em tokens.



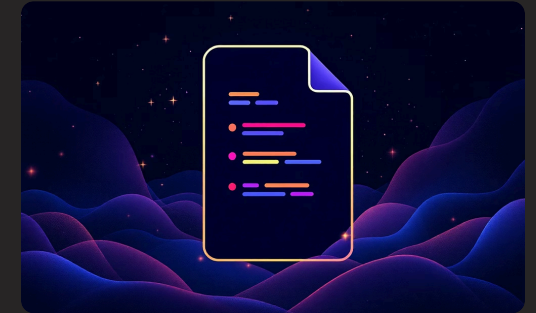
Parser (parser.y)

Realiza a análise sintática, construindo a árvore de sintaxe e integrando-se com a VM.



Máquina Virtual (vm.c/.h)

O coração da simulação, executa os comandos interpretados pelo parser.



Exemplo (test.pln)

Arquivo de script de exemplo para testar as funcionalidades da linguagem.



Makefile

Define as regras para a compilação e construção do executável.

A execução do sistema é simples: `./planelang test.pln`.



Conclusão: O Poder do PlaneLang



Simulação de Voo

PlaneLang oferece uma plataforma eficaz para simular operações de drone de forma controlada e segura.



VM Segura

A VM customizada garante uma execução robusta e rastreável, essencial para testes e desenvolvimento.



Aprendizado Prático

O projeto APS é uma demonstração completa de análise léxica, sintática e interpretação de linguagem.



Fácil Extensão

A arquitetura modular permite fácil adição de novos comandos e lógicas de voo.