九　尾　狐　の　語

K Language

Polymorphism in a C-like programming language

Diogo César – 11/0027931

A linguagem

A linguagem

Tipos de dados	void	char	int	float			
Palavras chave	if	else	do	while	return	asm	
Operadores aritméticos	+	–	*	/	%	++	––
Operadores relacionais	<	>	<=	>=	==	!=	
Operadores lógicos	&&	||	!				
Operador de atribuição	=						
Delimitadores de literais	'	"					
Vetores*	[	]					
Escopo e inicializadores	{	}					
Fim de comando	;						
Funções	(	)					
Separadores de listas	,						
Comentários	/*	*/	//				

Sintaxe

- Baseada na gramática da linguagem C apresentada por Harbison III e Steele (2002)
- Padrão ISO C (até C98)
- S.P. Harbison III and G.L. Steele. C, A Reference Manual. Fifth Edition. Prentice-Hall, 2002.

Sintaxe: escopo global

- Declaração (sem inicialização) de variáveis
- Definição de funções
- Qualquer ordem (mas não para usar)

Sintaxe: escopo de funções

- Declaração / inicialização de variáveis (1 por comando)
- If/if…else
- While/do…while
- Escopos aninhados
- Return
- Expressões (inclui atribuição)
- Inline “assembler”

Sintaxe: controle de fluxo

- if (condition) { statements } *else { statements }*
- while(condition) { statements }
- do { statements } while (condition);

Análise semântica:
verificação de escopo

- Do mais interno para o mais externo
- Variáveis e funções devem ser declaradas antes do uso
- Variáveis no escopo mais externo da função não podem ter o mesmo nome que a função

Análise semântica:
verificação de tipos

- Verificação de tipos
  - Se de tipos compatíveis: *char* → *int* → *float*
    - Se constantes: análise estática
    - Se variáveis: geração de código (com *casting* implícito)
  - Se de tipos incompatíveis:
    - Erro
    - Atribuição e chamada de funções

Strings

- Inicialização
  - char[] str = “string literal”;
- Vai para a seção .table
- Somente para output
  - Implementação de operações com vetores foi adiada

Polimorfismo

- Funções são discriminadas pelo nome e argumentos
- Tipo dos argumentos é inferido

- int add(int x, int y) { return x+y; }
- int add(int x, int y, int z) { return x+y+z; }
- float add(float x, float y) { return x+y; }

Polimorfismo: implementação

- Gerenciado pela tabela de símbolos
- Declaração
  - 1: insere o nome da função na tabela
  - 2: insere uma string que representa os tipos dos argumentos na nova tabela (ex.: “iif” para int, int, float)
- Chamada
  - 1: Procura pelo nome
  - 2: Procura pelos tipos dos argumentos (sem casting)

Polimorfismo: biblioteca de I/O

- Biblioteca de funções polimórficas “ligada” com todos os programas gerados (#reset)

- int read(int arg) { … }
- char read(char arg) { … }
- float read(float arg) { … }

- int a = read(a);

Polimorfismo: biblioteca de I/O

- void write(int arg) { … }
- void write(char arg) { … }
- void write(float arg) { … }
- void write(char arg[]) { … }

- char str[] = “awooo”;
- write(str);

Inline asm

- O literal é copiado diretamente no código gerado

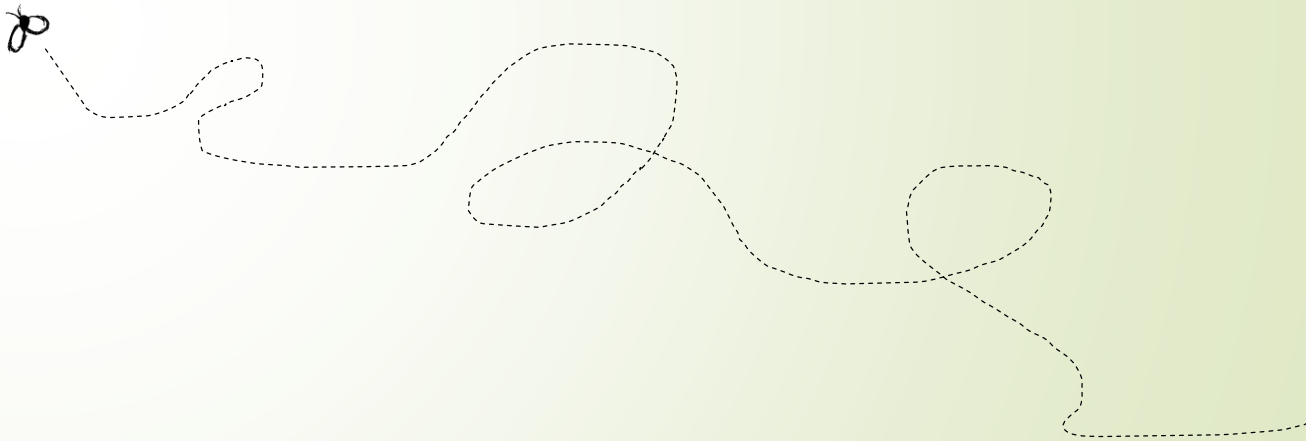
```
int read (int arg) {
        int val;
        asm("     scani  $0");
        return val;
}
```

- Código:

```
read_i:
        scani  $0
        return $0
```

Exemplos

Bugs



Bugs

!!!

~~Bugs~~

- "return" dentro de "if"

```
if (…) { return 1; }
else { return 2; }
```

- Workaround:

```
int ret;
if (…) { ret = 1; }
else { ret = 2; }
return ret;
```

~~Bugs~~Features

- Chamada no meio da expressão

int x = expr + func2(…);

- Workaround:

Int x = func2(…) + expr;

Features

- Type casting em comparações

```
int x = 1;
float y = 2.0;
x < y;
```

- Workaround:

```
int x = 1;
float y = 2.0;
float xf = x;
x < y;
```

Features

- Integração com o SO no caso de alguns erros sintáticos específicos

Features

- Integração com o SO no caso de alguns erros sintáticos específicos
- Redireciona mensagens do SO para o usuário (segmentation fault)

Features

- Integração com o SO no caso de alguns erros sintáticos específicos
- Redireciona mensagens do SO para o usuário (segmentation fault)
- Workaround:

Não programar errado

O que ficou para a próxima versão?

- Vetores de inteiros e ponto flutuante
  - Acesso a elementos por índice (operadores [])
- Atribuição no escopo global
- Remover bugs antigos
- ~~Inventar bugs novos~~

Obrigado!