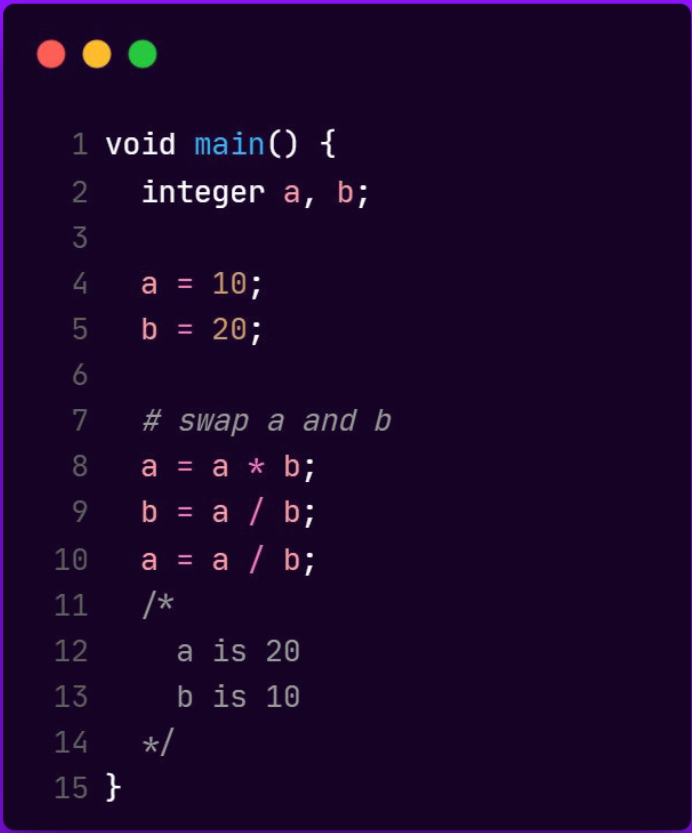


Linguagem K

Lógica da Computação - Cicero Tiago

1 Motivação

A motivação da linguagem criada pode ser definida como a necessidade didática de tornar mais explícitas algumas features da linguagem C, além de uma flexibilização da sintaxe e de operações entre tipos diferentes (strings e inteiros).



```
1 void main() {  
2     integer a, b;  
3  
4     a = 10;  
5     b = 20;  
6  
7     # swap a and b  
8     a = a * b;  
9     b = a / b;  
10    a = a / b;  
11    /*  
12        a is 20  
13        b is 10  
14    */  
15 }
```

2 Características e Curiosidades

```
1 string concat(void a, void b) {
2     return(a $ b);
3 }
4
5 integer main() {
6     integer a, b;
7
8     a = 10; # assignment
9     b = 20;
10
11     stdout(concat("a: ", a)); # a: 10
12     stdout(concat("b: ", b)); # b: 20
13
14     return(0);
15 }
```

A linguagem K possui as operações fundamentais da linguagem C:

- Operações Aritméticas
- Operações Lógicas
- Input/Output
- Comparações
- Loops
- Condicionais
- Funções

Além disso, possui também o tipo built-in **void** e permite operações adicionais como **concatenar** strings e números.

2.1 Operações Lógicas

Suporta operações lógicas fundamentais

- *and*
- *or*
- *not*

```
1 integer log_and(integer a, integer b) {
2   return (a and b);
3 }
4
5 integer log_or(integer a, integer b) {
6   return (a or b);
7 }
8
9 integer log_not(integer a) {
10  return (not a);
11 }
12
13 void main() {
14   integer a, b;
15
16   a = 1;
17   b = 0;
18
19   stdout("Logical Operations");
20
21   stdout("a and b: " $log_and(a, b)); # a and b: 0
22   stdout("a or b: " $log_or(a, b)); # a or b: 1
23   stdout("not a: " $log_not(a)); # not a: 0
24   stdout("not b: " $(not b)); # not b: 1
25 }
```

2.2 Operações Aritméticas

```
1 void main() {
2     integer a, b;
3     a = 30;
4     b = 10;
5
6     stdout("Arithmetical Operations");
7
8     stdout("a: " $a);
9     stdout("b: " $b);
10
11    stdout("");
12
13    stdout("(a + b): " $(a + b)); # (a + b): 40
14    stdout("(a - b): " $(a - b)); # (a - b): 20
15    stdout("(a / b): " $(a / b)); # (a / b): 3
16    stdout("(a * b): " $(a * b)); # (a * b): 300
17 }
```

Operações aritméticas fundamentais:

- Soma
- Subtração
- Divisão
- Multiplicação

2.3 Input/Output

Input e output podem ser feitos por meio das funções **built-in** de entrada e saída de dados. Por enquanto, a única entrada permitida é de número inteiros, mas **strings poderiam ser aceitas** em uma versão subsequente da linguagem. As funções built-in são:

- *stdin*
- *stdout*

```
1 void main() {  
2     integer in_value;  
3  
4     stdout("Test stdout"); # Test stdout  
5  
6     in_value = stdin();  
7  
8     stdout("stdin: " $in_value); # stdin: ...  
9 }
```

2.4 Comparações

```
1 void main() {
2   integer a, b, c;
3
4   b = 10;
5   c = 10;
6
7   stdout("b: "$b);
8   stdout("c: "$c);
9
10  stdout("");
11
12  stdout("b > c: " $(b > c)); # b > c: 0
13  stdout("b ≥ c: " $(b ≥ c)); # b ≥ c: 1
14  stdout("b < c: " $(b < c)); # b < c: 0
15  stdout("b ≤ c: " $(b ≤ c)); # b ≤ c: 1
16  stdout("b = c: " $(b = c)); # b = c: 1
17  stdout("b ≠ c: " $(b ≠ c)); # b ≠ c: 0
18 }
```

Comparações fundamentais:

- Maior;
- Maior ou igual;
- Menor;
- Menor ou igual;
- Igual;
- Diferente;

2.5 Loops

Aceita o loop while.

O exemplo ao lado gera os seguintes valores no console:

```
a ≤ b: a: 1 b: 5
a ≤ b: a: 2 b: 5
a ≤ b: a: 3 b: 5
a ≤ b: a: 4 b: 5
a ≤ b: a: 5 b: 5
a > b: a: 6 b: 5
```

```
1 void main() {
2     integer a, b;
3
4     a = 1;
5     b = 5;
6
7     while (a ≤ b) {
8         stdout("a ≤ b: " $"a: " $a $" b: " $b);
9         a = a + 1;
10    }
11
12    stdout("a > b: " $"a: " $a $" b: " $b);
13 }
```


2.6 Condicionais

Aceita a execução condicional conforme o exemplo ao lado.

A depender do valor *inputado* para a variável **c**, a saída em **stdout** vai ser alguma combinação das linhas 13, 15, 18 ou 20.

```
1 void main() {  
2     integer b, c;  
3  
4     b = 20;  
5     c = stdin();  
6  
7     stdout("Test Comparisons");  
8     stdout("b: " $b);  
9     stdout("c: " $c);  
10    stdout("");  
11  
12    if (b == c) {  
13        stdout("b == c");  
14    } else {  
15        stdout("b != c");  
16  
17        if (b > c) {  
18            stdout("b > c");  
19        } else {  
20            stdout("b < c");  
21        }  
22    }  
23 }
```

2.7 Funções

```
1 integer max(integer a, integer b) {
2   if (a > b) {
3     return(a);
4   }
5   return(b);
6 }
7
8 integer main() {
9   integer a, b;
10  a = 10;
11  b = 20;
12
13  stdout("a: " $a); # a: 10
14  stdout("b: " $b); # b: 20
15
16  stdout("max(a, b): " $max(a, b)); # max(a, b): 2
17
18  return(0);
19 }
```

A linguagem K permite a declaração e a chamada de funções.