K2 Language Documentation

Table of Contents

Introduction

K2 is an ultra-fast programming language designed to execute operations in the range of 70 nanoseconds to 9 milliseconds. It features a simple syntax for basic arithmetic operations and variable management.

Key Features

- **Ultra-Fast Execution**: Most operations complete in under 1000 nanoseconds
- **Simple Syntax**: Easy to learn and use
- **Performance Monitoring**: Built-in execution time display
- **File-Based Execution**: Run programs from files
- **Direct Expression Mode**: Execute single expressions from the command line

Installation

From Source

```
# Download the package
git clone https://github.com/k2-language/k2.git
cd k2

# Compile
g++ -std=c++17 -O3 k2.cpp -o k2

# Install globally (optional)
sudo cp k2 /usr/local/bin/
```

Using the Installation Script

```
# Download the installation script
wget https://k2-language.org/downloads/install.sh

# Make it executable
chmod +x install.sh

# Run the installer
sudo ./install.sh
```

Using the Debian Package

```
# Download the Debian package
wget https://k2-language.org/downloads/k2-language_1.0.0-1_amd64.deb

# Install the package
sudo dpkg -i k2-language_1.0.0-1_amd64.deb
```

Language Syntax

K2 has a clean, minimal syntax designed for readability and performance.

Comments

Comments start with `#` and continue to the end of the line:

```
# This is a comment
```

Statements

Each statement is written on a separate line:

```
x = 10
y = 20
sum = x + y
print sum
```

Variables

Variables in K2 store integer values. Variable names can contain letters, numbers, and underscores, but must start with a letter.

Variable Assignment

```
x = 10
y = 20
result = x + y
```

Using Variables

Variables can be used in expressions and printed:

```
x = 5
y = x * 2
print y  # Outputs: 10
```

Operations

K2 supports the following arithmetic operations:

Addition

```
sum = a + b
```

Subtraction

```
difference = a - b
```

Multiplication

```
product = a * b
```

Division

```
quotient = a / b
```

Complex Expressions

```
result = a + b * c
```

Note: K2 follows standard operator precedence (multiplication and division before addition and subtraction).

Commands

K2 has several built-in commands:

Print

The `print` command outputs a value:

```
print 42
print x
print x + y
```

Show Execution Time

The `show_exec_time` command controls whether execution times are displayed:

```
# Enable execution time display
show_exec_time on

# Disable execution time display
show_exec_time off
```

Examples

Basic Example

```
# Basic K2 program
x = 42
print x
```

Output:

```
42
Execution time: 1230 nanoseconds
```

Arithmetic Example

```
# Arithmetic operations
a = 10
b = 5

sum = a + b
diff = a - b
product = a * b
quotient = a / b

print sum
print diff
print product
print quotient
```

Output:

```
15
Execution time: 1150 nanoseconds
5
Execution time: 1120 nanoseconds
50
Execution time: 1130 nanoseconds
2
Execution time: 1140 nanoseconds
```

Fibonacci Example

```
# Fibonacci sequence
a = 0
b = 1

print a
print b

c = a + b
print c
a = b
b = c

c = a + b
print c
a = b
b = c

c = a + b
print c
```

Output:

```
0
1
1
2
3
```

Timing Example

```
# Execution time control
x = 100
y = 200

# Show execution time
print x + y

# Disable execution time display
show_exec_time off
print x * y

# Enable execution time display
show_exec_time on
print x - y
```

Output:

```
300
Execution time: 1250 nanoseconds
20000
-100
Execution time: 1180 nanoseconds
```

Performance

K2 is designed for speed. Here are typical execution times:

Operation Type	Execution Time Range
Variable assignment	100-400 nanoseconds
Simple arithmetic	70-500 nanoseconds
Print operation	1000-2000 nanoseconds
Complex expressions	300-5000 nanoseconds

Advanced Usage

Command Line Options

K2 supports several command line options:

```
# Run a K2 program file
k2 program.k2

# Execute a single expression
k2 -e "print 100 + 200"
```

Performance Optimization

To get the best performance from K2:

1. Keep expressions simple
2. Reuse variables when possible
3. Minimize the number of print statements
4. Use `show_exec_time off` when not benchmarking

Troubleshooting

Common Errors

Undefined Variable

```
Error: undefined variable 'x'
```

This error occurs when you try to use a variable that hasn't been defined. Make sure to assign a value to the variable before using it.

Unknown Operator

```
Error: unknown operator '&'
```

K2 only supports the basic arithmetic operators (+, -, *, /). Make sure you're using a supported operator.

Division by Zero

```
Error: division by zero
```

This error occurs when you try to divide by zero. Check your divisor to ensure it's not zero.

Getting Help

If you encounter issues with K2, you can:

1. Check the documentation at https://k2-language.org/documentation
2. Report bugs on GitHub at https://github.com/k2-language/k2/issues
3. Join the K2 community on Discord at https://discord.gg/k2-language

License