

ReCode 2013, Montréal

Ы

を装す

期州

Nenad "DocKimbel" Rakocevic

Let's start by taking a bird's eye view



We are still at a very early age of computing...



...and in the future, AI will laugh at our technologies!



There are thousands of programming tools/languages...

A [edit]	
----------	--

- A# .NET
- A# (Axiom)
- A-0 System
- A+
- A++
- ABAP
- ABC
- ABC ALGOL
- ABLE
- ABSET
- ABSYS
- Abundance
- ACC
- Accent
- Ace DASL
- ACT-III
- Action!

- ActionScript
- Ada
- Adenine
- Agda
- Agilent VEE
- Agora
- AIMMS
- Alef
- ALF
- ALGOL 58
- ALGOL 60
- ALGOL 68
- Alice
- Alma-0
- AmbientTalk
- Amiga E
- AMOS

- AMPL
- APL
- AppleScript
- Arc
- ARexx
- Argus
- AspectJ
- Assembly language
- ATS
- Ateji PX
- AutoHotkey
- Autocoder
- · Autolt
- AutoLISP / Visual LISP
- Averest
- AWK
- Axum

...but we are still searching for the right one.



We are often wasting time to workaround dead ends...



...while we should have fun working on computers!



Our computing world is going through deep changes...



"Massively parallel 64 core computer costs \$99" -Parallela

We are moving away from traditional computers.



...yet we are using rusty software solutions.

Position Jun 2013	Position Jun 2012	Delta in Position	Programming Language	Ratings Jun 2013	Delta Jun 2012	Status	Year of Birth
1	1	=	С	17.809%	+0.08%	А	1972
2	2	=	Java	16.656%	+0.39%	A	1995
3	4	t	Objective-C	10.356%	+1.26%	A	1983
4	3	Ļ	C++	8.819%	-0.54%	А	1983
5	7	11	РНР	5.987%	+0.70%	А	1995

Rebol was partially answering those concerns...



...doing wonders, only recently acknowledged,...



...but Red will go far beyond!



What is Red?

- A "full stack" programming language
 - True general-purpose programming solution
 - A stand-alone toolchain
- A tool for empowering users
 - Simple to use
 - Unlimited usage, no arbitrary restrictions
 - Feeling "in control" again



Red is bringing back the fun to programming!

Natural scope of application



A standalone toolchain 1/5



A standalone toolchain 2/5

- Compile script and run it from memory \$ red script.red
- Compile script and output an executable \$ red -o script script.red
- Cross-compile script and output an executable
 \$ red -t Windows -o script script.red
- Compile a Red/System script and output an executable \$ red script.reds
- Compile script as shared library \$ red -dlib script.red
- Launch Red in REPL mode \$ red

Currently: >> do/args %red.r "-dlib script.red"

A standalone toolchain 3/5

• Cross-compilation made right!

```
$ red -t <TargetID> script.red
```

Target ID	Target Description
MSDOS	Windows, x86, console-only applications
Windows	Windows, x86, native applications
Linux	GNU/Linux, x86
Linux-ARM	GNU/Linux, ARMv5
Darwin	Mac OS X Intel, console-only applications
Syllable	Syllable OS, x86
Android	Android, ARMv5
Android-x86	Android, x86

A standalone toolchain 4/5



A standalone toolchain 5/5



Last dependency to remove for full feature access!

Red language 1/4

- Syntax and semantics very close to Rebol
 - Definitional scoping
 - Dynamic binding
- Paradigm-neutral
 - Imperative, OOP, functional, symbolic,...
- Optionally typed arguments and locals
 - From pure interpretation to very specialized compilation
 - Type inference by default, when possible

Red language 2/4

- Unicode support
 - Input sources in UTF-8, external codec for other encodings
 - Auto-adaptative internal representation: UCS-1, UCS-2 or UCS-4
- Concurrency support
 - Task parallelism: light threads over multiple cores with Actors.
 - Data parallelism: SIMD and multi-cores processing using parallel series.
- Meta-DSL abilities
 - DSL-maker dialect (higher-level than PARSE)

Red language 3/4

• Compiles to Red/System code



Red language 4/4

• Compiled, interpreted, JIT-compiled



```
Red [title: "Hello"]
```

```
print "Hello World!"
```

```
inc: func [n [integer!]][n + 1]
```

```
foo: function [a [integer!] /bar return: [string!]][
   z: "zero"
   either integer? a [
       return z
   ][
       append "result=" mold inc a
   ]
]
```

Red internals



Red/System overview

- Purely imperative, C-level language, with a Red syntax
- Statically compiled, x4 slower than C (naïve compilation for now)
- Namespaces support: context, with
- Limited type system:
 - Logic!, byte!, integer!, float!, float32!, struct!, pointer!, c-string!, function!
 - Simple type inference
 - Type casting supported
 - Values type reflexion (variadic RTTI functions)
- Compiler directives: #define, #include, #import, #syscall, #if, #either, #switch,...
- Low-level CPU support (interruptions, I/O, stack, privileged mode)
- Inlined ASM support

Red project metrics

- **BSD** license (BSL for the runtime parts)
- Source code hosted on **Github** since March 2011
 - version 0.3.2, 9 commiters, ~2050 public commits
 - 496 tickets in bugtracker (95% closed)
 - ~18000 unit tests (framework built by Peter WA Wood)
 - 390 KiB of sources for Red/System
 - 8500 LOC for Red/System compiler
 - 2600 LOC for Red/System linker
 - 2800 LOC for Red compiler
 - 800 LOC for Red interpreter
 - 14000 LOC for Red runtime library (~150KiB compiled on x86)

Red remaining tasks for 1.0, until...

- Complete core parts:
 - implement object!, error!, typeset!, binary!, decimal!, date!, time!, ...
 - add proper error handling and arguments type-checking
 - implement I/O infrastructure
 - implement concurrency support
 - define a module system and modular compilation
- Provide a minimal Red IDE
- Documentation:
 - Write Red reference documentation
 - Write some first-steps tutorials
- Prepare a new red-lang.org site for 1.0 launch

... the real Red!

• Real Red will be the 2.0



... the real Red!

- Final compiler architecture
 - Two-stage JIT-compiler, with optional AOT optimizations
 - Documented public API for lexer, parser, emitter and linker
 - Plugin-oriented internal architecture
 - Written purely in Red (self-hosted)
 - Red/System: improved and cleaned-up syntax and semantics
 - Very open to contributors!
- Implications for current codebase in Rebol 2
 - Disposable code with limited lifetime
 - Not documented, monolithic, rigid, not meant for contributions

Project Organisation



- 2 collaborators on Github
- 11 contributors
- 2 mailing-list admins
- 3 Facebook Red page admins
- iiqux (IRCbot for reporting commits)
- You?

Project Funding





Any questions?