

C language in our world

6.5. 2019 FI MUNI

Brno

@jurajmichalek

<https://georgik.rocks>

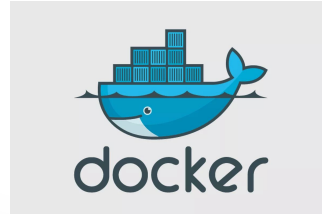
<https://www.ysofters.com>

Grab the source code

<https://github.com/ysoftdevs/cpp-examples>



Who am I?



YSofers Blog: <https://www.ysofters.com>



Blog: <https://georgik.rocks>

C language today

Docker

Console: Minunit, Check, Curl

UI: Allegro5, SDL2, GTK3, QT

Server: CUPS

Gradle

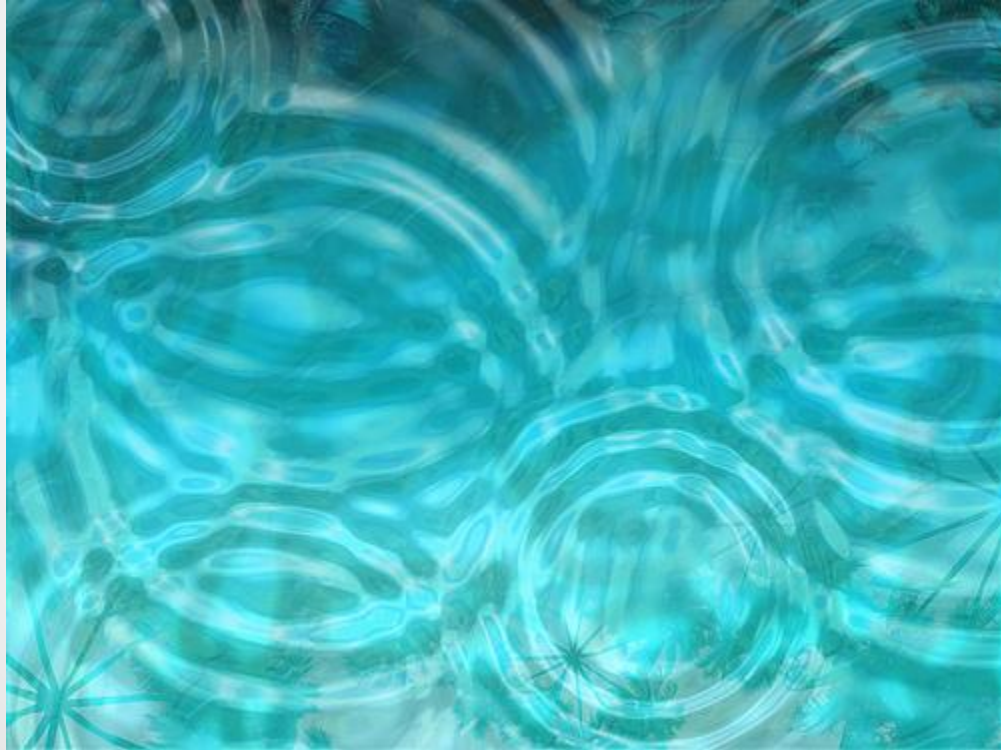
IDEs, NuGet, IoT

Jenkins, Bamboo, TeamCity

Node, Go language

2xFI MUNI theses

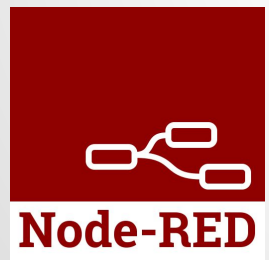
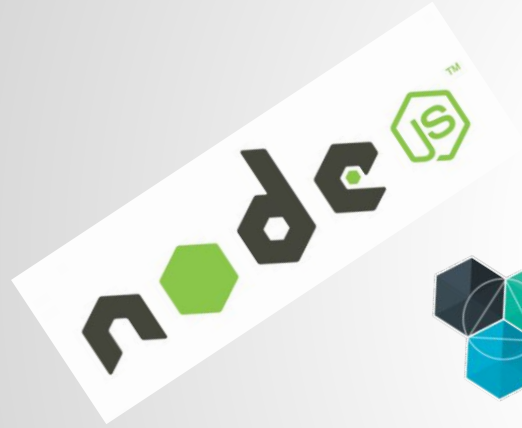
Technologies influencing each other



Programming languages we know
strongly influence the way we think
about programming.

- JS Conf 2014 - Jenna Zeigen

Breeze of fresh ideas starts blowing from other technologies...





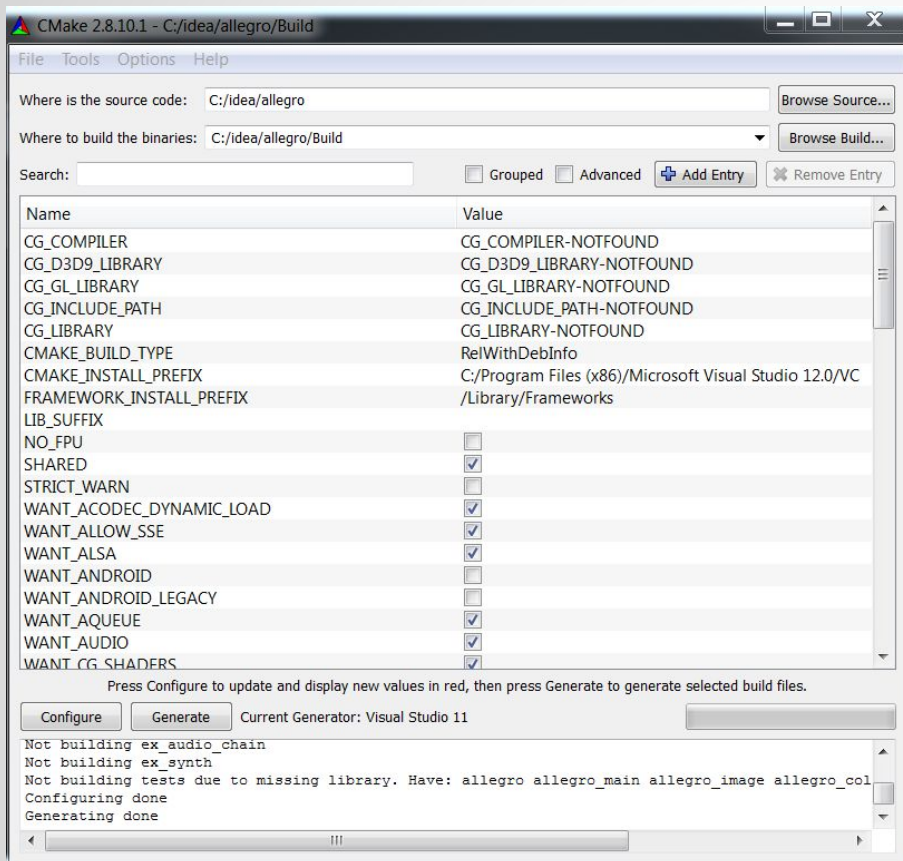
Containers

- Linux based, Windows based
- hub.docker.com



Allegro





Allegro 5

Win, Lin, Mac

iOS, Android

<http://alleg.sourceforge.net/a5docs/refman/>

Initialization

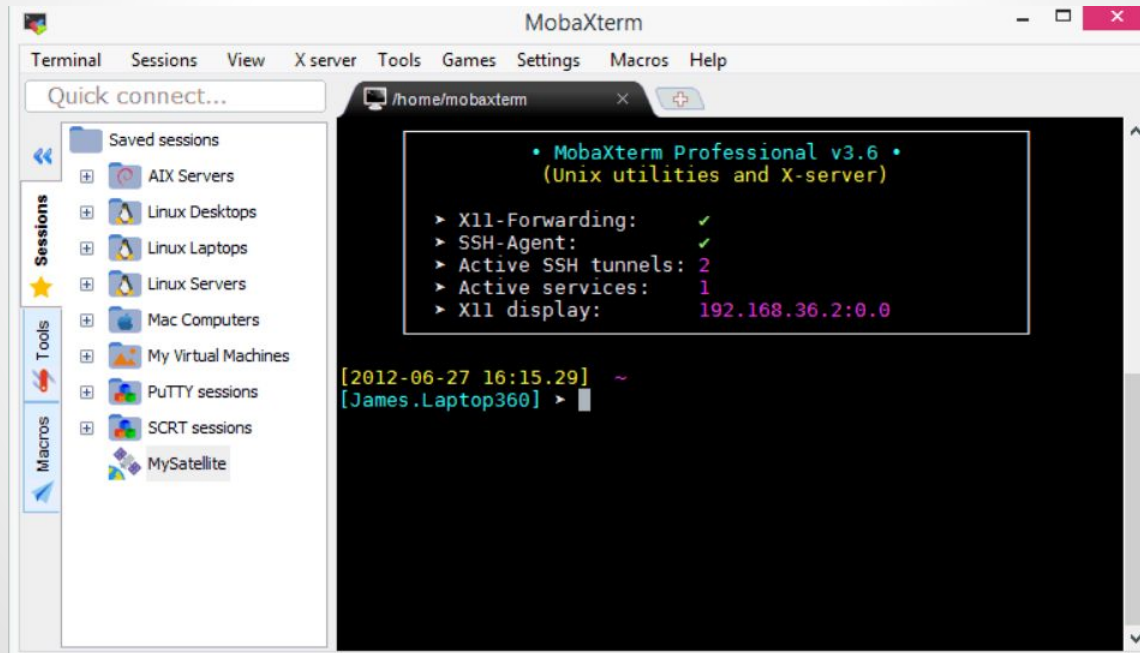
```
al_init();
```

Graphic environment

```
al_create_display(int w, int h)
```

MobaXTerm

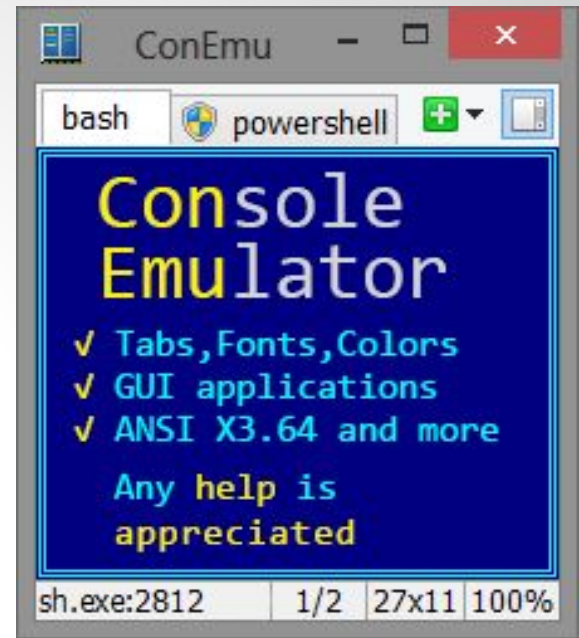
<https://mobaxterm.mobatek.net/>



Conemu Maximus 5

Powerful terminal for Windows

use with PowerShell, Python, Ruby...



<https://code.google.com/p/conemu-maximus5/>



CMake 3.0.2

By: scaftw

CMake is a family of tools designed to build, test and package software. CMake is used to control the software compilation process using simple platform and compiler independent configuration files. CMake generates native makefiles and workspaces that can be used in the compiler environment of your choice.

3,884 downloads | Tags [make](#) [build](#) [test](#) [package](#)

```
C:\> choco install cmake
```

Yum/Apt-like installation of Win packages

<https://chocolatey.org>

SDL

Simple Directmedia Layer



Made with SDL



Made with SDL

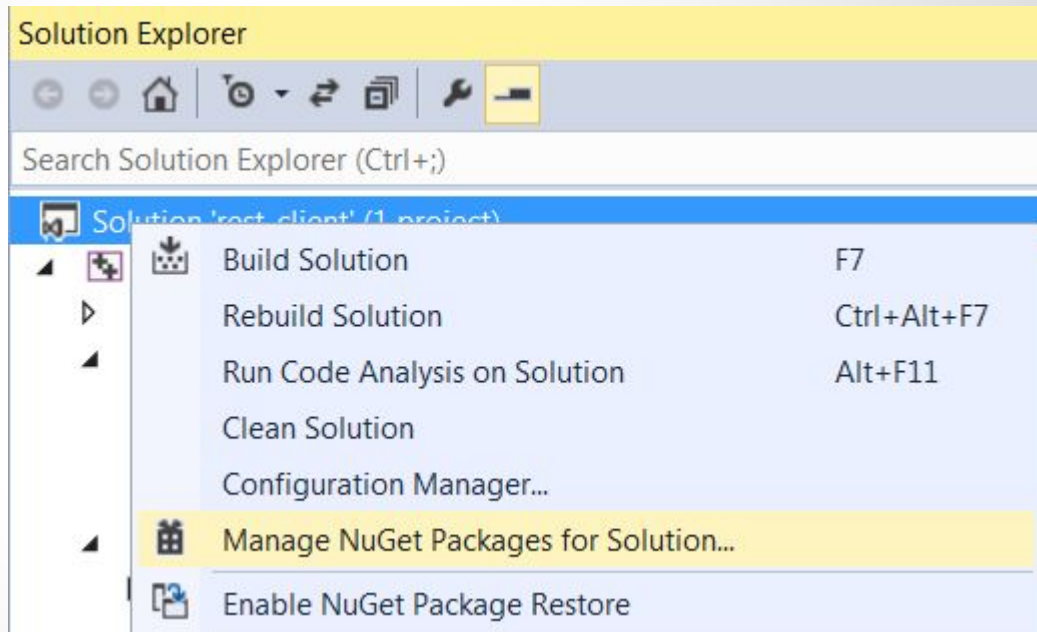


NuGet - <http://www.nuget.org>





NuGet.Tools.2013



▶ Installed packages

Stable Only

Sort by: Relevance

◀ Online

All

nuget.org

Microsoft and .NET

Search Results

▶ Updates

Each package is licensed to you by its owner. Microsoft is not responsible for, nor does it grant any licenses to, third-party packages.

**Simple DirectMedia Layer (SDL)**

Simple DirectMedia Layer is a cross-platform multimedia library designed to provide low l...

Install

**Simple DirectMedia Layer**

This is the Simple DirectMedia Layer, a generic API that provides low level access to audio, keyboard, mouse, and dis...

**Simple DirectMedia Layer Redist**

Redistributable components for for package 'sdl2'

**SDL_image**

SDL_image loads images as SDL surfaces.

**DD4T Support for DVM4T**

A DD4T based implementation of the DVM4T framework. Includes a number of basic Attributes for common field types...

**DVM4T Framework**

Domain View Models For Tridion - a .NET framework for creating strongly typed domain view models based on conte...

**DD4T Providers for Tridion 2011sp1**

Providers for SDL Tridion 2011 SP1

1 2 ▶

sdl

Created by: Sam Lantinga, SDL contributors

Id: SDL**Version:** 1.2.15.15**Last Published:** 5.7.2013**Downloads:** 1398**License**[View License](#)

LGPL-2.1

[Project Information](#)[Report Abuse](#)**Description:**

Simple DirectMedia Layer is a cross-platform multimedia library designed to provide low level access to audio, keyboard, mouse, joystick, 3D hardware via OpenGL, and 2D video framebuffer. Homepage: <http://www.libsdl.org/>

Tags: sdl native CoApp nativepackage**Dependencies:**

SDL.redist (≥ 1.2.15.15)

Each item above may have sub-dependencies subject to additional license agreements.

Settings

Close

Multiplatform

SDL officially supports

Windows, Mac OS X, Linux, iOS, and Android.

Support for other platforms may be found in the source code.

SDL versions

1.2 stable - rock solid

2.x development - new features

SDL_init(flags)

SDL_INIT_TIMER - The timer subsystem

SDL_INIT_AUDIO - The audio subsystem

SDL_INIT_VIDEO - The video subsystem

SDL_INIT_CDROM - The cdrom subsystem

SDL_INIT_JOYSTICK - The joystick subsystem

SDL_INIT EVERYTHING - All of the above

SDL_INIT_NOPARACHUTE - Prevents SDL from catching fatal signals

SDL_INIT_EVENTTHREAD - Runs the event manager in a separate thread

Quit application

`SDL_quit()`

Window

```
SDL_CreateWindow("Hello World!", 100, 100,  
640, 480, SDL_WINDOW_SHOWN);
```

Load bitmap

```
SDL_Surface *bmp = nullptr;  
bmp = SDL_LoadBMP("smajlik.bmp");
```

Visual data

SDL_Renderer

SDL_Texture

Keyboard

`SDL_PollEvent(SDL_Event *event)`

`event.key.keysym.sym`

Timer

```
SDL_TimerID SDL_AddTimer(  
    Uint32          interval,  
    SDL_TimerCallback callback,  
    void*           param)
```


Mouse

```
SDL_GetMouseState(*x, *y);
```

Text

Not implemented



Extensions

extension for many languages:

C++, Java, PHP, Python, Ruby

PyGame

Power of C and Power of Python

<http://www.pygame.org>



Kivy.org



iOS

Android

Windows Desktop

Windows Phone

Raspberry Pi

Cross-platform development of smartphone application with the Kivy framework

- Master thesis - Mgr. Ondřej Chrastina: http://is.muni.cz/th/430596/fi_m/



Gradle Native Builds C/C++, Objective-C

<http://gradle.org/getting-started-native/>



Build tool

Extensible by plugins

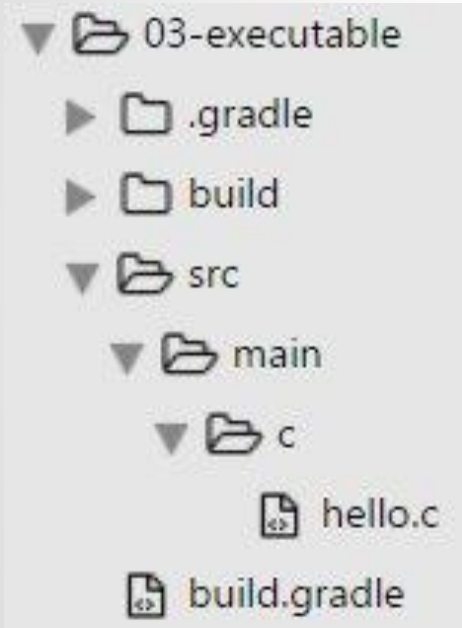
Power of Domain Specific Language

<http://plugins.gradle.org>



Search Gradle Plugins

Project structure



Convention over configuration

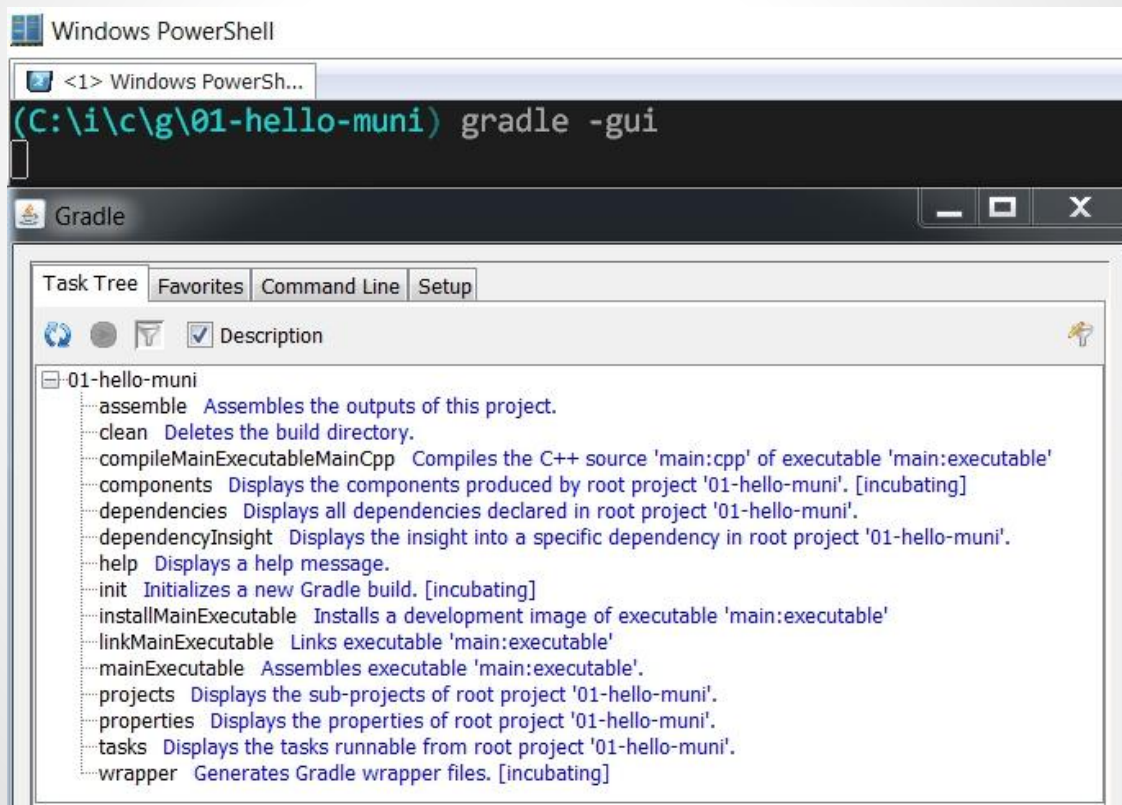
Decrease number of decisions that developers need to make

http://en.wikipedia.org/wiki/Convention_over_configuration

C plugin

```
build.gradle
1  apply plugin: 'c'
2
3  model {
4      components {
5          main(NativeExecutableSpec) {
6
7          }
8      }
9  }
```

Gradle command line & GUI



gradle components

```
(C:\i\c\g\03-executable) gradle components
:components

-----

Root project
-----

Native executable 'main'
-----

Source sets
    C source 'main:c'
        src\main\c

Binaries
    Executable 'main:executable'
        build using task: :mainExecutable
        install using task: :installMainExecutable
        platform: windows_x86
        build type: debug
        flavor: default
        tool chain: Tool chain 'visualCpp' (Visual Studio)
        executable file: build\binaries\mainExecutable\main.exe
```

Gradle build Linux package

Netflix Nebula OS Package plugin:

<http://plugins.gradle.org/plugin/nebula.os-package>



```
1  plugins {
2      id "nebula.os-package" version "2.0.3"
3  }
4
5  apply plugin: 'c'
6
7  ▼ model {
8      ▼ components {
9          hello(NativeExecutableSpec) {
10
11          }
12      }
13  }
14
15  ▼ ospackage {
16      packageName = "hello"
17      version = "1.0"
18      release = 1
19      os = LINUX
20      packageDescription = "Linux Gradle hello package"
21      summary = "contains binary with hello world example"
22
23      from("build/binaries/helloExecutable") {
24          into "/usr/bin/"
25      }
26  }
27
28  buildDeb {
29      requires("libc6")
30  }
31
32  buildRpm {
33      requires("libc6")
34  }
```

Build package

```
(georgik@pidi:pts/5)——(...-plugin/04-hello-linux-package)─  
(17:33:%)— gradle hE bd —(Sat,Dec06)─  
:compileHelloExecutableHelloCpp  
:linkHelloExecutable  
:helloExecutable  
:buildDeb  
  
BUILD SUCCESSFUL  
  
Total time: 7.12 secs  
(georgik@pidi:pts/5)——(...-plugin/04-hello-linux-package)─  
(17:33:%)— dpkg -c build/distributions/hello_1.0-1_all.deb  
drwxr-xr-x georgik/0          0 2014-12-06 17:33 ./usr/  
drwxr-xr-x georgik/0          0 2014-12-06 17:33 ./usr/bin/  
-rwxr-xr-x georgik/0        6367 2014-12-06 17:33 ./usr/bin/hello
```

Note: Gradle supports abbreviation. You can write hE instead of helloExecutable

SDL2 and Android



Android Studio + NDK + Gradle

<https://github.com/georgik/sdl2-android-example>

<https://georgik.rocks/tag/sdl2/>

GTK

<https://www.gtk.org/>



YSoft + CUPS + QT

Thesis:

Dávid Kaya - Linux client for YSoft SafeQ

https://is.muni.cz/th/409878/fi_b/

CUPS

<https://github.com/apple/cups>

Server, backend, filter

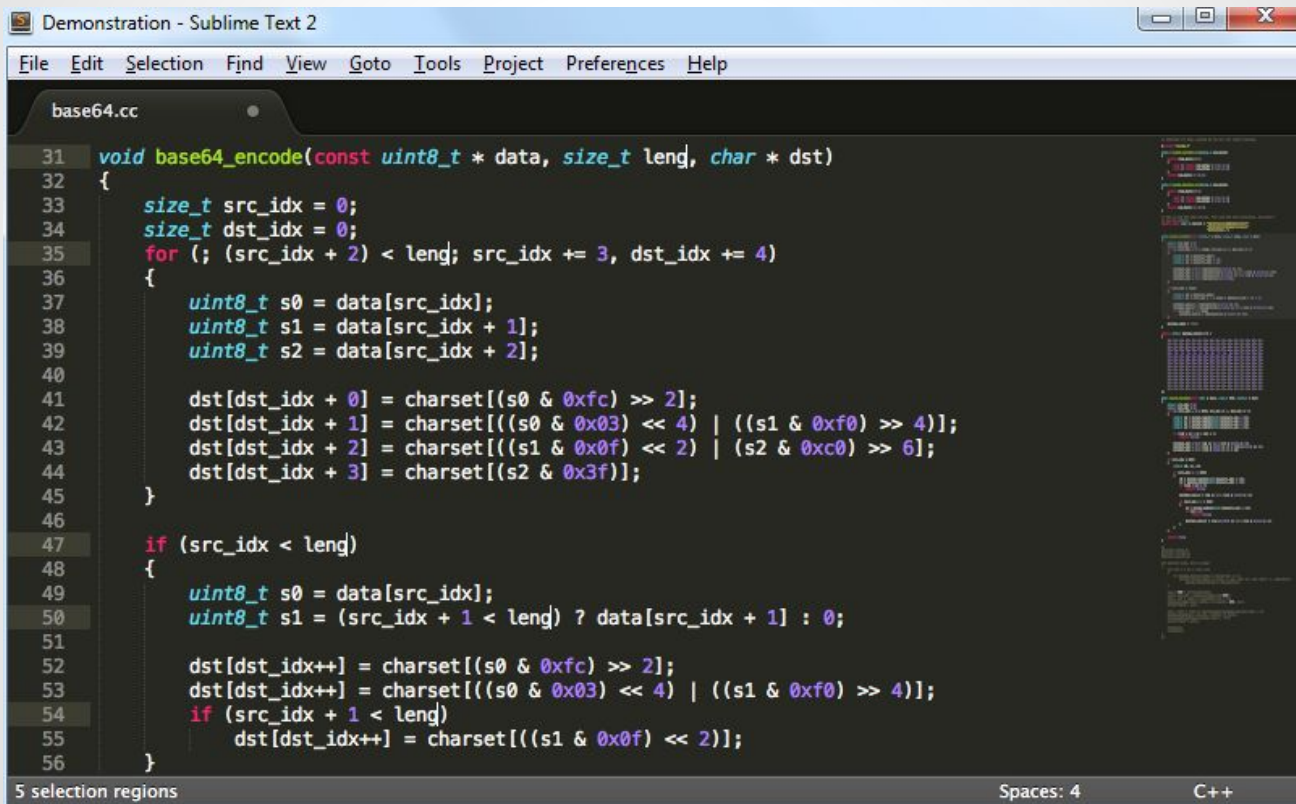
IDE & Text editors



A hackable text editor
for the 21st Century

<https://atom.io/>

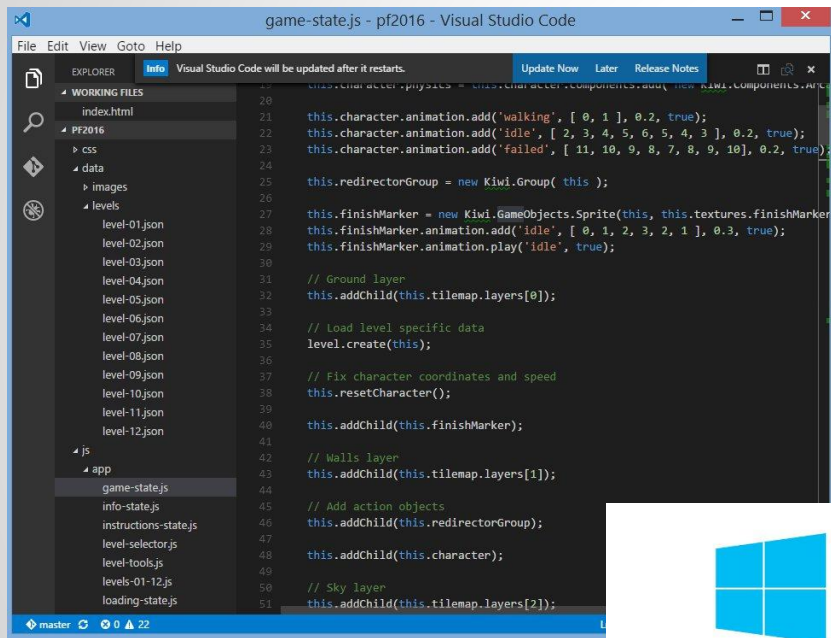
Sublime Text



```
31 void base64_encode(const uint8_t * data, size_t leng, char * dst)
32 {
33     size_t src_idx = 0;
34     size_t dst_idx = 0;
35     for (; (src_idx + 2) < leng; src_idx += 3, dst_idx += 4)
36     {
37         uint8_t s0 = data[src_idx];
38         uint8_t s1 = data[src_idx + 1];
39         uint8_t s2 = data[src_idx + 2];
40
41         dst[dst_idx + 0] = charset[(s0 & 0xfc) >> 2];
42         dst[dst_idx + 1] = charset[((s0 & 0x03) << 4) | ((s1 & 0xf0) >> 4)];
43         dst[dst_idx + 2] = charset[((s1 & 0x0f) << 2) | (s2 & 0xc0) >> 6];
44         dst[dst_idx + 3] = charset[(s2 & 0x3f)];
45     }
46
47     if (src_idx < leng)
48     {
49         uint8_t s0 = data[src_idx];
50         uint8_t s1 = (src_idx + 1 < leng) ? data[src_idx + 1] : 0;
51
52         dst[dst_idx++] = charset[(s0 & 0xfc) >> 2];
53         dst[dst_idx++] = charset[((s0 & 0x03) << 4) | ((s1 & 0xf0) >> 4)];
54         if (src_idx + 1 < leng)
55             dst[dst_idx++] = charset[((s1 & 0x0f) << 2)];
56     }
57 }
```

5 selection regions Spaces: 4 C++

Visual Studio Code



Code editing. Redefined.

- <https://code.visualstudio.com/>



↓ Windows

Windows 7, 8, 10



↓ .deb

Debian, Ubuntu

↓ .rpm

Red Hat, Fedora, CentOS



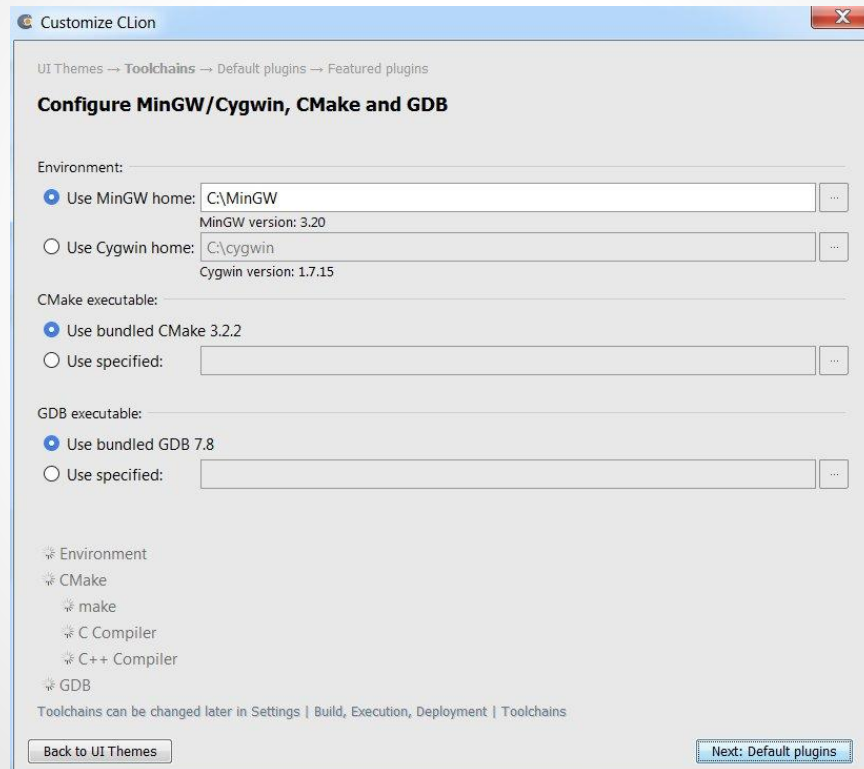
↓ OS X

OS X Yosemite, El Capitan



CLion

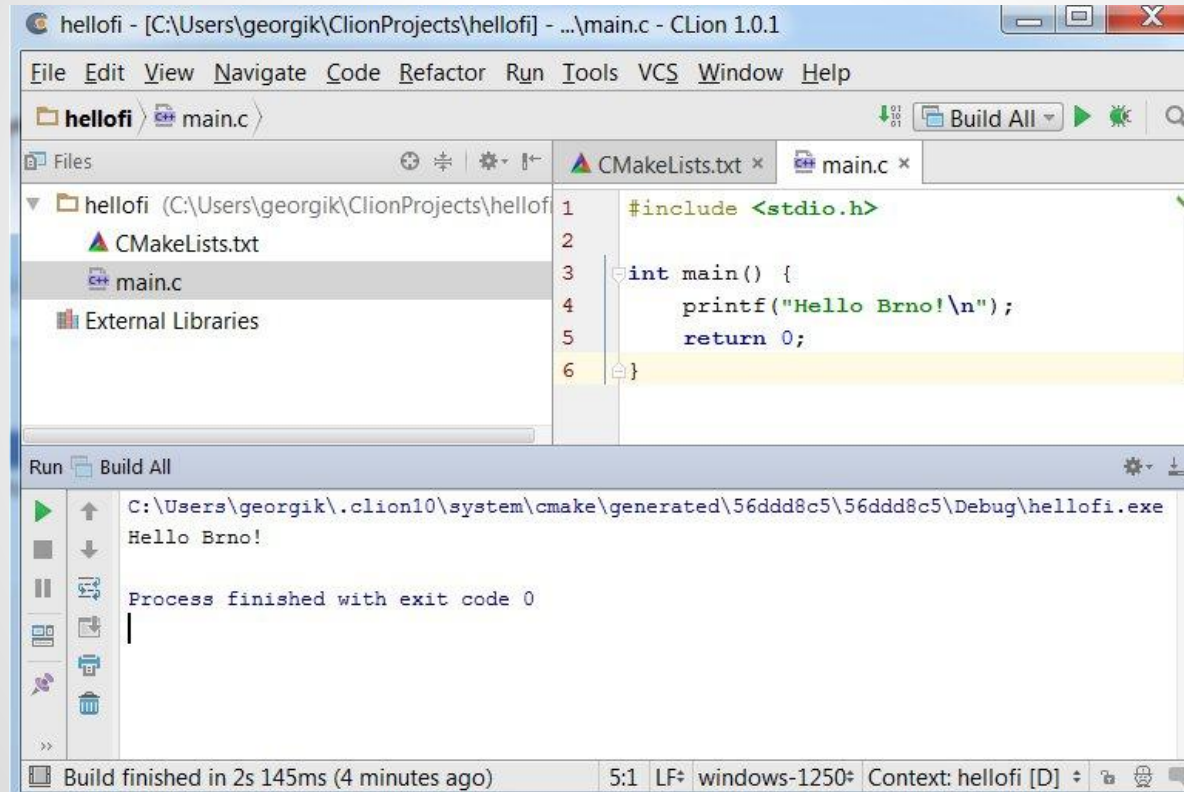
Toolchain detection



Edit project



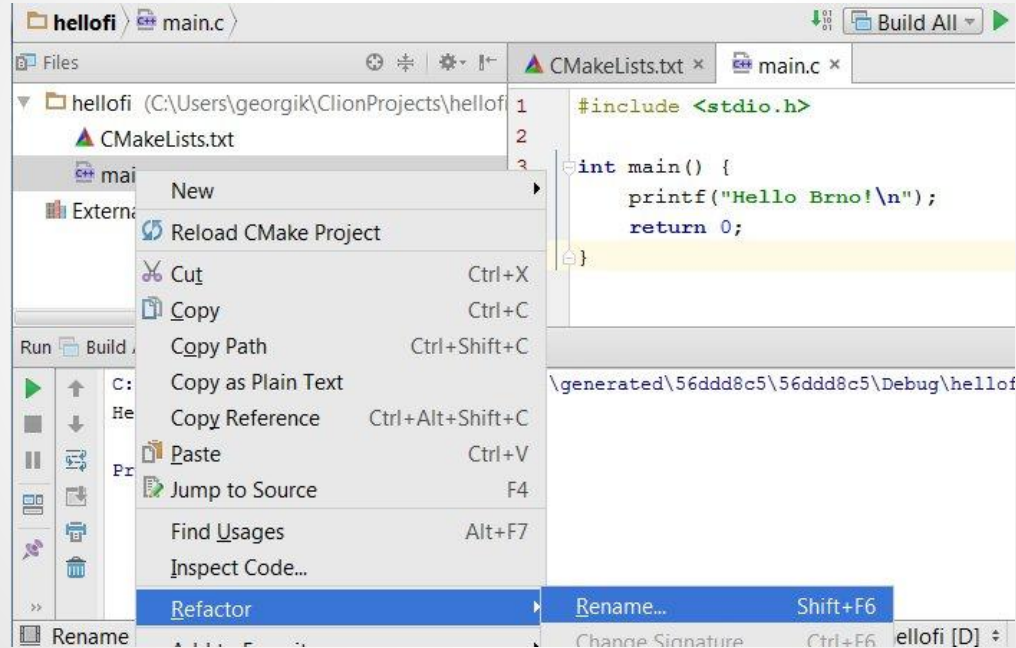
CLion



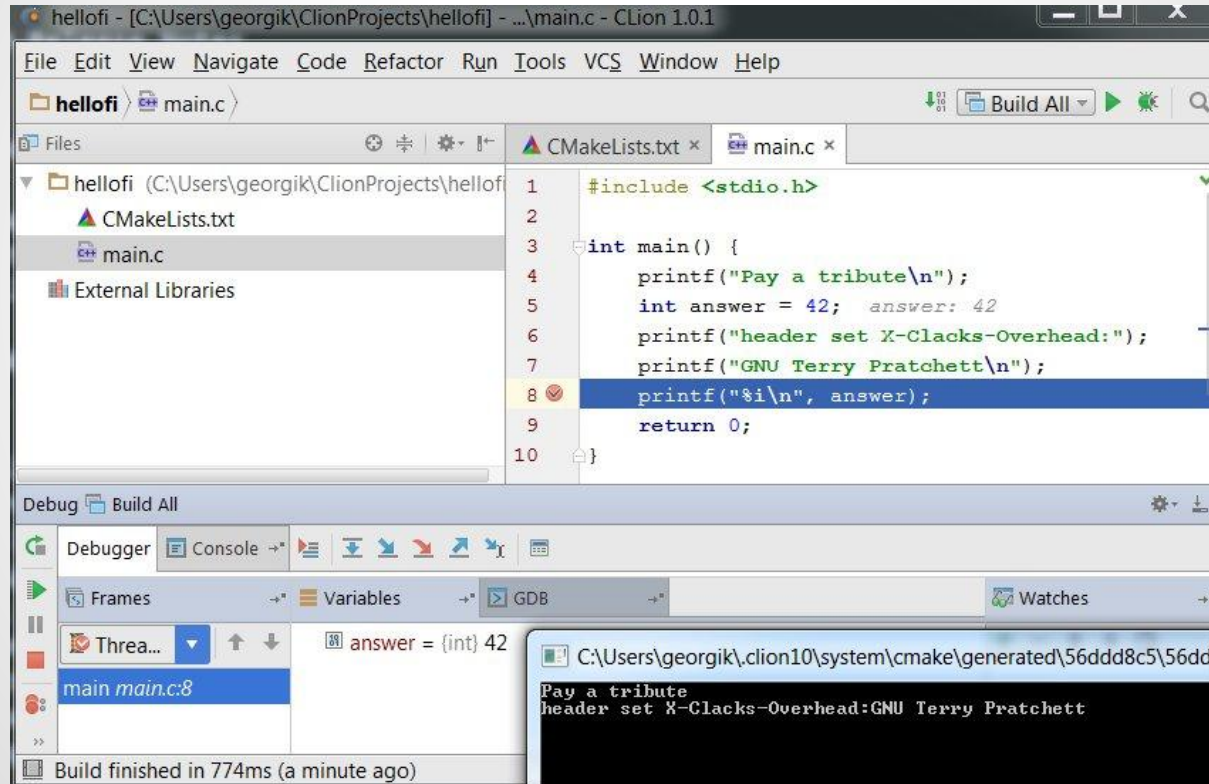
Leverage Refactor



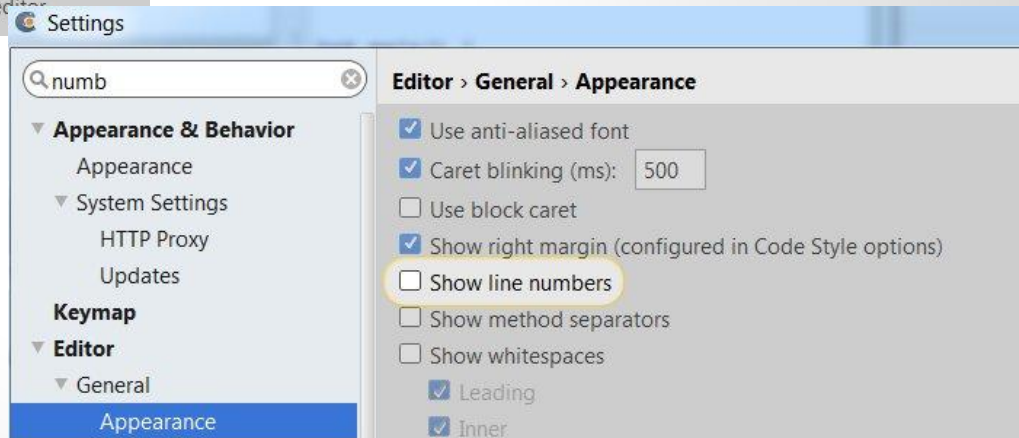
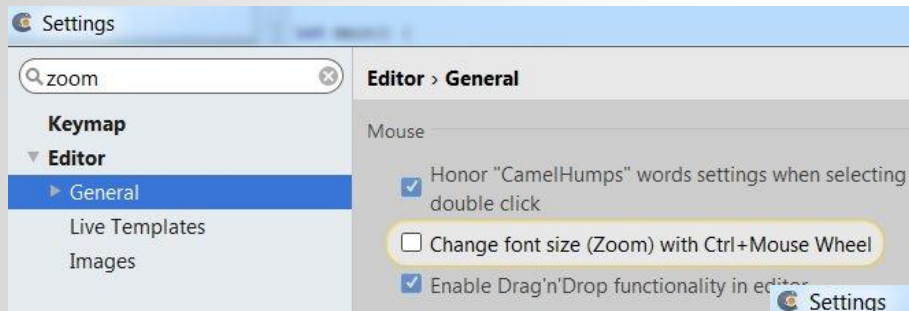
CLion



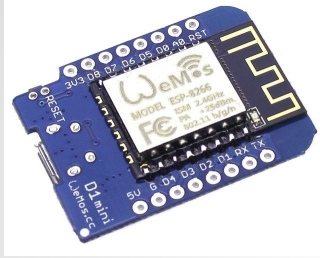
Use Debugger



Fine tune



Disabled by default for all JetBrains tools :-)



C in embedded and IoT world



Arduino + Platform IO



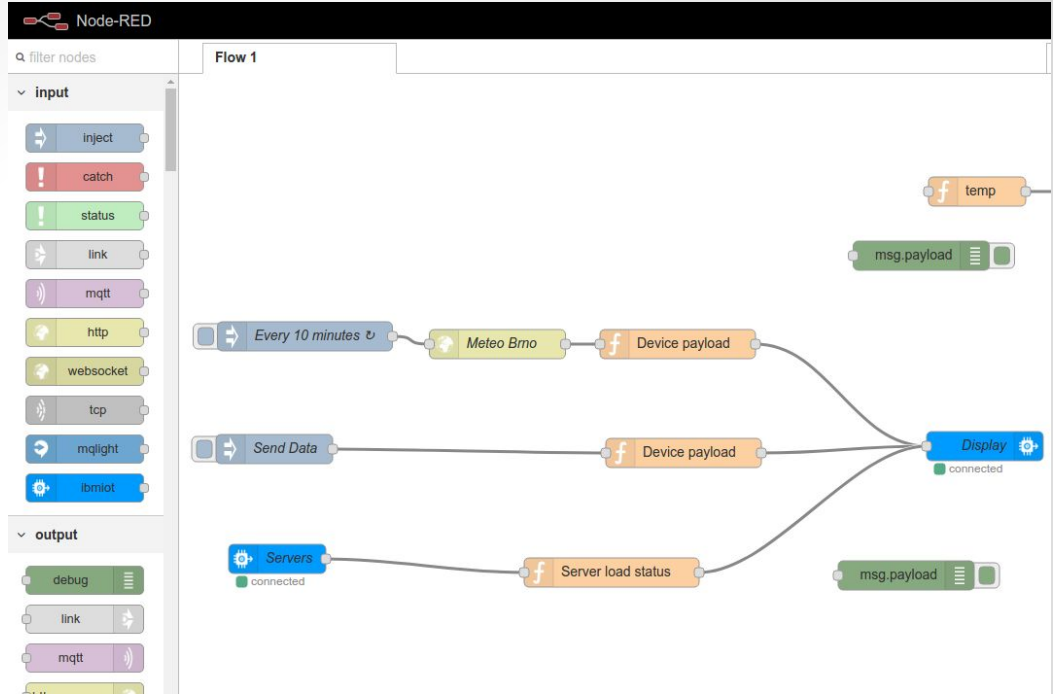
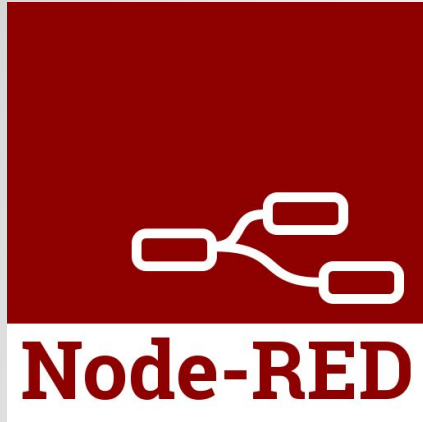
<http://platformio.org/>

LampESP example

<https://github.com/georgik/LampESP>

- OTA
- WifiManager
- Web Server
- MQTT Client (works also with Bluemix)
- TaskScheduler (async style)

More info: <http://georgik.rocks/category/iot/>



<https://nodered.org/>

Monkey C

Garmin Connect IQ

- <https://developer.garmin.com/connect-iq>



lot-inc - podcast



<http://www.iot-inc.com/category/mediatype/podcasts/>

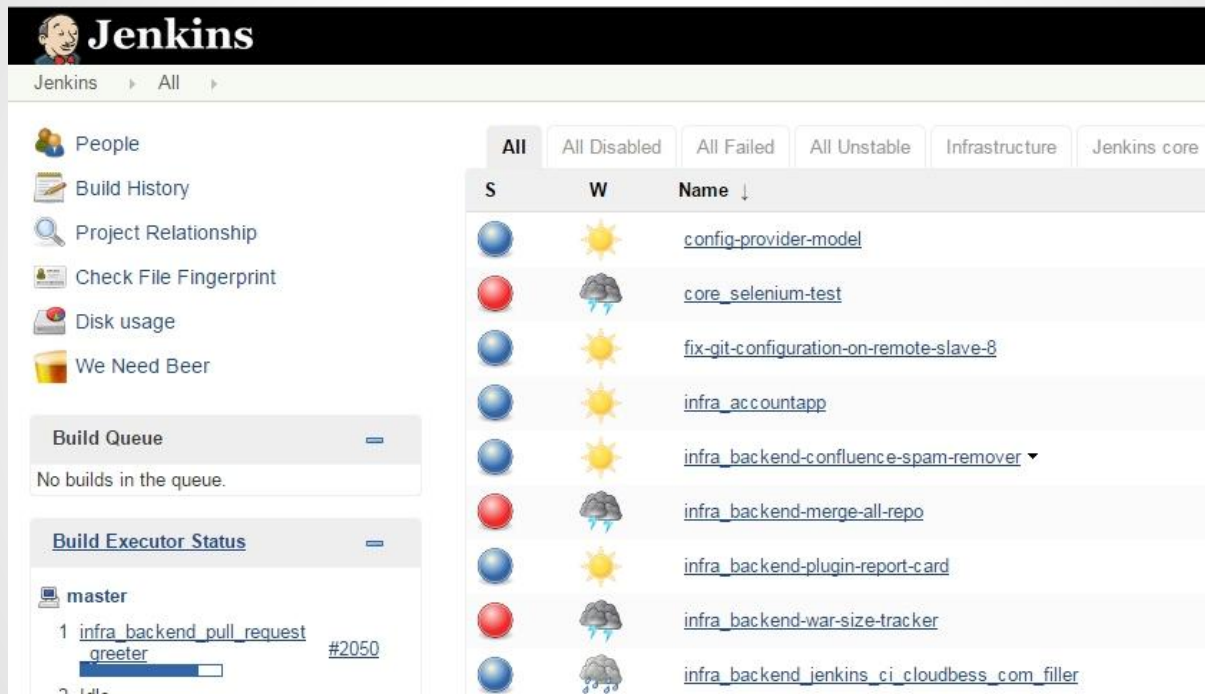
Continuous integration



Jenkins



Jenkins



The screenshot shows the Jenkins web interface. The top header has the Jenkins logo and name. Below it, there's a navigation bar with 'Jenkins' and 'All' tabs. The left sidebar contains several menu items: 'People', 'Build History', 'Project Relationship', 'Check File Fingerprint', 'Disk usage', and 'We Need Beer'. Below these are two expandable sections: 'Build Queue' (showing 'No builds in the queue.') and 'Build Executor Status' (showing a list of executors, including 'master' with a build 'infra_backend_pull_request' in progress). The main content area on the right displays a table of builds, filtered by 'All'. The table has columns for status (S), weather icon (W), and name. The builds listed include 'config-provider-model', 'core_selenium-test', 'fix-git-c-configuration-on-remote-slave-8', 'infra_accountapp', 'infra_backend-confluence-spam-remover', 'infra_backend-merge-all-repo', 'infra_backend-plugin-report-card', 'infra_backend-war-size-tracker', and 'infra_backend_jenkins_ci_cloudbess_com_filler'.

S	W	Name ↓
		config-provider-model
		core_selenium-test
		fix-git-c-configuration-on-remote-slave-8
		infra_accountapp
		infra_backend-confluence-spam-remover
		infra_backend-merge-all-repo
		infra_backend-plugin-report-card
		infra_backend-war-size-tracker
		infra_backend_jenkins_ci_cloudbess_com_filler

Hit for Windows users: Do not install Jenkins into path with special characters and blank space.
E.g: Wrong: C:\Program Files (x86)\Jenkins. Correct: Use C:\projects\jenkins

From desktop to cloud

Software is slow

Software is hard to write

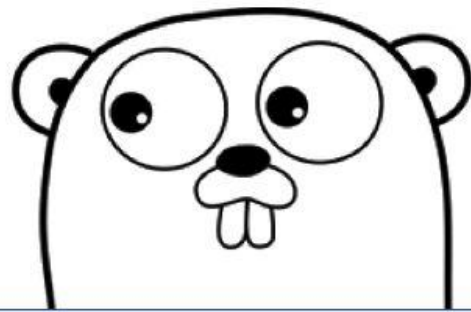
Software is hard to scale

Go

<http://golang.org>

Authors:

- Ken Thompson - known for Unix
- Rob Pike - known for UTF-8
- Robert Griesemer



Main features of language

syntax patterns from dynamic languages

performance of C

blazing fast compilation

output one binary

concurrency

libraries from internet (e.g. Github)

works on: Mac, Linux, Windows and more...

Materials

Andreas Krennmair

<http://synflood.at/tmp/golang-slides/mrmcd2012.html#1>

Steve Francia

<http://spf13.com/presentation/first-go-app/>

Thanks to artists

images used in this presentation were published under creative commons license. Links to originals:

<http://www.flickr.com/photos/fatboyke/3405148748/>

<http://www.flickr.com/photos/teveve/6301993588/>

<http://www.flickr.com/photos/stevewilhelm/6242822362/>

<http://en.wikipedia.org/wiki/Chess>

<http://www.flickr.com/photos/akosma/9486807123/>

<http://www.flickr.com/photos/charlestilford/6362884553/>

<http://www.flickr.com/photos/ciat/6917871707/>

<http://www.flickr.com/photos/anieto2k/4455227465/>

<http://www.geograph.ie/photo/1113036>

[http://commons.wikimedia.org/wiki/File:Dark_Sky_\(3274525313\).jpg](http://commons.wikimedia.org/wiki/File:Dark_Sky_(3274525313).jpg)

<http://www.elfwood.com/~arknott/Red-Dragon.2539297.html>

<http://commons.wikimedia.org/wiki/File:Wolf-River-swamp-North-Mississippi.jpg>

<http://pako0007.deviantart.com/art/Zombie-Imp-2-267822507>

<http://www.flickr.com/photos/bogenfreund/367091428/>

<http://www.flickr.com/photos/infinite-magic/4016608841/>

<http://www.flickr.com/photos/lennysan/4403695597/>

<http://www.flickr.com/photos/avaverino/4870587458/>

L10N - verify your translations



<http://www.microsoft.com/Language>



Swiss knife tool for web <https://curl.haxx.se/>

Generate source code:

```
curl http://www.ysoft.com -o index.html --libcurl download.c
```



**The future is already
here — it's just not
very evenly
distributed.**

YSofters

Twitter: [@ysoftdevs](https://twitter.com/ysoftdevs)

GitHub: github.com/ysoftdevs

Blog: www.ysofters.com

Thesis: Andrij.Stecko@ysoft.com