C language in our world

16.5. 2016 FI MUNI
Brno
@jurajmichalek
https://www.ysofters.com
Grab the source code

https://github.com/ysoftdevs/cpp-examples
Who am I?

Blog: http://georgik.sinusgear.com
C language today
NuGet
Gradle & C language
Jenkins
IDEs
Go language
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...
NuGet - http://www.nuget.org
Simple DirectMedia Layer (SDL)
Simple DirectMedia Layer is a cross-platform multimedia library designed to provide low l...

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 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

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

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:
SDLredist (≥ 1.2.15.15)
Each item above may have sub-dependencies subject to additional license agreements.
Made with SDL
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);
Not implemented
Extensions

extension for many languages:

C++, Java, PHP, Python, Ruby
PyGame

Power of C and Power of Python

http://www.pygame.org
Cross-platform development of smartphone application with the Kivy framework
### CMake Configuration

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CG_COMPILER</td>
<td>CG_COMPILER-NOTFOUND</td>
</tr>
<tr>
<td>CG_D3D9_LIBRARY</td>
<td>CG_D3D9_LIBRARY-NOTFOUND</td>
</tr>
<tr>
<td>CG_GL_LIBRARY</td>
<td>CG_GL_LIBRARY-NOTFOUND</td>
</tr>
<tr>
<td>CG_INCLUDE_PATH</td>
<td>CG_INCLUDE_PATH-NOTFOUND</td>
</tr>
<tr>
<td>CG_LIBRARY</td>
<td>CG_LIBRARY-NOTFOUND</td>
</tr>
<tr>
<td>CMAKE_BUILD_TYPE</td>
<td>RelWithDebInfo</td>
</tr>
<tr>
<td>CMAKE_INSTALL_PREFIX</td>
<td>C:/Program Files (x86)/Microsoft Visual Studio 12.0/VC</td>
</tr>
<tr>
<td>FRAMEWORK_INSTALL_PREFIX</td>
<td>/Library/Frameworks</td>
</tr>
<tr>
<td>LIB_SUFFIX</td>
<td></td>
</tr>
<tr>
<td>NO_FPU</td>
<td></td>
</tr>
<tr>
<td>SHARED</td>
<td>X</td>
</tr>
<tr>
<td>STRICT_WARN</td>
<td></td>
</tr>
<tr>
<td>WANT_ACODEC_DYNAMIC_LOAD</td>
<td>X</td>
</tr>
<tr>
<td>WANT_ALLOW_SSE</td>
<td>X</td>
</tr>
<tr>
<td>WANT_ALSA</td>
<td>X</td>
</tr>
<tr>
<td>WANT_ANDROID</td>
<td></td>
</tr>
<tr>
<td>WANT_ANDROID_LEGACY</td>
<td></td>
</tr>
<tr>
<td>WANT_AQUEUE</td>
<td></td>
</tr>
<tr>
<td>WANT_AUDIO</td>
<td></td>
</tr>
<tr>
<td>WANT_CG_FRAMEWORKERS</td>
<td></td>
</tr>
</tbody>
</table>

Press Configure to update and display new values in red, then press Generate to generate selected build files.
Allegro 5.1

Win, Lin, Mac

iOS, Android

http://alleg.sourceforge.net/a5docs/refman/
Initialization

al_init();
Graphic environment

al_create_display(int w, int h)
L10N - verify your translations

http://www.microsoft.com/Language
ConEmu Maximus 5

Powerful terminal for Windows use with PowerShell, Python, Ruby…

https://code.google.com/p/conemu-maximus5/
Yum/Apt-like installation of Win packages

https://chocolatey.org
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
Project structure

Convention over configuration
Decrease number of decisions that developers need to make

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

```groovy
apply plugin: 'c'

model {
    components {
        main(NativeExecutableSpec) {

        }
    }
}
```
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
plugins {
    id "nebula.os-package" version "2.0.3"
}

apply plugin: 'c'

model {
    components {
        hello(NativeExecutableSpec) {

        }
    }
}

ospackage {
    packageName = "hello"
    version = "1.0"
    release = 1
    os = LINUX
    packageDescription = "Linux Gradle hello package"
    summary = "contains binary with hello world example"
    from("build/binaries/helloExecutable") {
        into "/usr/bin/"
    }
}

buildDeb []
    requires("libc6")
}

buildRpm {
    requires("libc6")
}
Build package

Note: Gradle supports abbreviation. You can write hE instead of helloExecutable
Swiss knife tool for web [https://curl.haxx.se/](https://curl.haxx.se/)

Generate source code:
curl http://www.ysoft.com -o index.html --libcurl download.c
Continuous integration
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
IDE & Text editors
```c
void base64_encode(const uint8_t *data, size_t len, char *dst)
{
    size_t src_idx = 0;
    size_t dst_idx = 0;
    for (; (src_idx + 2) < len; src_idx += 3, dst_idx += 4)
    {
        uint8_t s0 = data[src_idx];
        uint8_t s1 = data[src_idx + 1];
        uint8_t s2 = data[src_idx + 2];

        dst[dst_idx + 0] = charset[(s0 & 0xfc) >> 2];
        dst[dst_idx + 1] = charset[((s0 & 0x3f) << 4) | ((s1 & 0xfc) >> 4)];
        dst[dst_idx + 2] = charset[((s1 & 0xf0) << 2) | (s2 & 0xc0)];
        dst[dst_idx + 3] = charset[(s2 & 0x3f)];
    }

    if (src_idx < len)
    {
        uint8_t s0 = data[src_idx];
        uint8_t s1 = (src_idx + 1 < len) ? data[src_idx + 1] : 0;

        dst[dst_idx++] = charset[(s0 & 0xfc) >> 2];
        dst[dst_idx++] = charset[((s0 & 0x3f) << 4) | ((s1 & 0xf0) >> 4)];
        if (src_idx + 1 < len)
            dst[dst_idx++] = charset[(s1 & 0x0f) << 2];
    }
}
```
Visual Studio Code

Code editing. Redefined.
- [https://code.visualstudio.com/](https://code.visualstudio.com/)
Toolchain detection
Edit project
Leverage Refactor
Use Debugger
Fine tune

Disabled by default for all JetBrains tools :-(
Qt Creator

```cpp
#include <QApplication>
#include "mainwindow.h"

int main(int argc, char *argv[])
{
    Q_INIT_RESOURCE(application);
    QApplication app(argc, argv);
    app.setOrganizationName("QtProject");
    app.setApplicationName("Application Example");
    MainWindow mainWin;
    mainWin.show();
    return app.exec();
}
```
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 Krennmaier
http://synflood.at/tmp/golang-slides/mrmcd2012.html#1

Steve Francia
Thanks to artists

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

YSofters

Twitter: @ysoftdevs
GitHub: github.com/ysoftdevs
Blog: www.ysofters.com
Technology Hour: www.meetup.com/ysoft-th
Thesis: Andryi.Stetsko@ysoft.com