链滴

一个基于Lua的跨平台自动构建工具：xmake

作者：waruqi
原文链接：https://ld246.com/article/1466933844731
来源网站：链滴

简介

XMake是一个跨平台自动构建工具，支持在各种主流平台上构建项目，类似cmake、automake、pr make，但是更加的方便易用，工程描述语法更简洁直观，支持平台更多，并且集创建、配置、编译打包、安装、卸载、运行于一体。

- 项目主页
- 在线文档
- 在线源码

支持特性

1. 支持windows、mac、linux、ios、android等平台，自动检测不同平台上的编译工具链（也可手配置）

编译windows项目采用原生vs的工具链，不需要使用cygwin、mingw（当然这些也支持）

2. 支持自定义平台编译配置，可以很方便的扩展第三方平台支持

3. 采用lua脚本语法描述项目，描述规则简单高效，逻辑规则可灵活修改，并且不会生成相关平台的程文件，是工程更加简单明了

4. 支持创建模板工程、配置项目、编译项目、运行、打包、安装和卸载等常用功能（后续还会增加：动生成文档、调试等模块）

5. 支持编译c/c++/objc/swift成静态库、动态库、命令行可执行程序

6. 提供丰富的工程描述api，使用简单灵活，例如添加编译文件只需（还支持过滤排除）：

```
add_files("src/*.c", "src/asm/**.S", "src/*.m")
```

7. 支持头文件、接口、链接库依赖、类型的自动检测，并可自动生成配置头文件config.h

8. 支持自定义编译配置开关，例如如果在工程描述文件中增加了enable_xxx的开关，那么配置编译的候就可以手动进行配置来启用它：

```
xmake config --enable_xxx=y
```

9. 提供一键打包功能，不管在哪个平台上进行打包，都只需要执行一条相同的命令，非常的方便

10. 支持全局配置，一些常用的项目配置，例如工具链、规则描述等等，都可以进行全局配置，这样不需要每次编译不同工程，都去配置一遍

11. 除了可以自动检测依赖模块，也支持手动强制配置模块，还有各种编译flags。

12. 支持插件扩展、平台扩展、模板扩展、选项自定义等高级功能

13. 提供一些内置的常用插件（例如：自动生成doxygen文档插件，宏脚本记录和运行插件）

14. 宏记录插件里面提供了一些内置的宏脚本（例如：批量打包一个平台的所有archs等），也可以在令行中手动记录宏并回放执行

15. 提供强大的task任务机制

16. 不依赖makefile和make，实现直接编译，内置自动多任务加速编译, xmake是一个真正的构架工，而不仅仅是一个工程文件生成器

17. 自动检测ccache，进行自动缓存提升构建速度

####简单例子

创建一个c++ console项目：

```
    xmake create -l c++ -t 1 console
 or xmake create --language=c++ --template=1 console
```

工程描述文件：xmake.lua

```
target("console")
    set_kind("binary")
    add_files("src/*.c")
```

配置工程：

这个是可选的步骤，如果只想编译当前主机平台的项目，是可以不用配置的，默认编译release版本。

```
   xmake f -p iphoneos -m debug
or xmake f --plat=macosx --arch=x86_64
or xmake f -p windows
or xmake config --plat=iphoneos --mode=debug
or xmake config --plat=android --arch=armv7-a --ndk=xxxxx
or xmake config -p linux -a i386
or xmake config -p mingw --cross=i386-mingw32- --toolchains=/xxx/bin
or xmake config -p mingw --sdk=/mingwsdk
or xmake config --help
```

编译工程：

```
   xmake
or xmake -r
or xmake --rebuild
```

运行目标：

```
   xmake r console
or xmake run console
```

打包所有：

```
   xmake p
or xmake package
or xmake package console
or xmake package -o /tmp
or xmake package --output=/tmp
```

通过宏脚本打包所有架构:

```
   xmake m package
or xmake m package -p iphoneos
or xmake m package -p macosx -f "-m debug" -o /tmp/
```

```
or xmake m package --help
```

安装目标：

```
   xmake i
or xmake install
or xmake install console
or xmake install -o /tmp
or xmake install --output=/tmp
```

详细使用方式和参数说明，请参考文档

或者运行：

```
   xmake -h
or xmake --help
or xmake config --help
or xmake package --help
or xmake macro --help
...
```

一些使用xmake的项目：

- tbox
- gbox
- libsvx
- 更多项目

####后续工作

1. 实现生成.ipa、.apk、.app、.deb、.rmp的打包插件
2. 实现包的自动依赖管理，如果依赖包不存在，会去自动下载编译安装后，继续进行构建，支持交叉台的包管理
3. 实现转换automake、cmake的工到xmake的描述文件的插件，实现无缝编译
4. 实现vs、xcode等工程文件生成插件

####简单例子

```
-- the debug mode
if is_mode("debug") then

    -- enable the debug symbols
    set_symbols("debug")

    -- disable optimization
    set_optimize("none")
end

-- the release mode
```

```
if is_mode("release") then

    -- set the symbols visibility: hidden
    set_symbols("hidden")

    -- enable fastest optimization
    set_optimize("fastest")

    -- strip all symbols
    set_strip("all")
end

-- add target
target("test")

    -- set kind
    set_kind("static")

    -- add files
    add_files("src/*.c")
```