From Source Code to Executable

A Day in the Life of a Build System

@segiddins

You might know me from:

-  CocoaPods
-  Bundler
-  Bazel
-  Twitter

Agenda

What is a build system?

Why do we need build systems?

Describing a build

Executing a build

What comes next?

What is a build system?

Build System

$$\text{buildsystem} := \text{code} \rightarrow \text{executable}$$

did you know, you use build systems every day?

Why do we need build systems?

main.m

```
@import Foundation;

int main(int argc, char** argv) {
 NSLog(@"Ran with %d args", argc);
 return 0;
}
```

Building main.m

```
$ clang \
 -x objective-c \
 -fmodules \
-o executable \
main.m
```

Wait

this was supposed to be the **simple** example?

And even then, it's not so simple

```
$ clang -x objective-c -fmodules -o executable main.m -v

"/Applications/Xcode-14.0.0.app/Contents/Developer/Toolchains/XcodeDefault.xctoolchain/usr/bin/clang" -cc1 -triple arm64-apple-macosx12.0.0 -Wundef-prefix=TARGET_OS_ -Wdeprecated-objc-isa-usage -Werror=deprecated-objc-isa-usage -Werror
 ↳ =implicit-function-declaration -emit-obj -mrelax-all --mrelax-relocations -disable-free -clear-ast-before-backend -disable-llvm-verifier -discard-value-names -main-file-name main.m -mrelocation-model pic -pic-level 2 -mframe-pointer=non-leaf
 ↳ -fno-strict-return -fno-rounding-math -funwind-tables=2 -fobjc-msgsend-selector-stubs -target-sdk-version=12.3 -fvisibility-inlines-hidden-static-local-var -target-cpu apple-m1 -target-feature +v8.5a -target-feature +fp-armv8 -target-feature +neon -target-feature +crc
 ↳ -target-feature +crypto -target-feature +dotprod -target-feature +fp16fml -target-feature +ras -target-feature +lse -target-feature +rdm -target-feature +rcpc -target-feature +zcm -target-feature +zcz -target-feature +fullfp16 -target-feature +sm4 -target-feature +sha3
 ↳ -target-feature +sha2 -target-feature +aes -target-abi darwinpcs -fallow-half-arguments-and-returns -debugger-tuning=lldb -target-linker-version 819.6 -v -resource-dir
 ↳ /Applications/Xcode-14.0.0.app/Contents/Developer/Toolchains/XcodeDefault.xctoolchain/usr/lib/clang/14.0.0 -isysroot /Applications/Xcode-14.0.0.app/Contents/Developer/Platforms/MacOSX.platform/Developer/SDKs/MacOSX.sdk -I/usr/local/include
 ↳ -internal-isystem /Applications/Xcode-14.0.0.app/Contents/Developer/Platforms/MacOSX.platform/Developer/SDKs/MacOSX.sdk/usr/local/include -internal-isystem
 ↳ /Applications/Xcode-14.0.0.app/Contents/Developer/Toolchains/XcodeDefault.xctoolchain/usr/lib/clang/14.0.0/include -internal-externc-isystem
 ↳ /Applications/Xcode-14.0.0.app/Contents/Developer/Platforms/MacOSX.platform/Developer/SDKs/MacOSX.sdk/usr/include -internal-externc-isystem /Applications/Xcode-14.0.0.app/Contents/Developer/Toolchains/XcodeDefault.xctoolchain/usr/include
 ↳ -Wno-reorder-init-list -Wno-implicit-int-float-conversion -Wno-c99-designator -Wno-final-dtor-non-final-class -Wno-extra-semi-stmt -Wno-misleading-indentation -Wno-quoted-include-in-framework-header -Wno-implicit-fallthrough -Wno-enum-enum-conversion -Wno-enum-float-conversion -Wno-elaborated-en
 ↳ =/Users/segiddins/Desktop/BuildSystemExample -ferror-limit 19 -stack-protector 1 -fstack-check -mdarwin-stkchk-strong-link -fblocks -fencode-extended-block-signature -fregister-global-dtors-with-atexit -fgnuc-version=4.2.1
 ↳ -fmodules -fimplicit-module-maps -fmodules-cache-path=/var/folders/p3/950_j_ps7xq47vrzdj1qvg940000gn/C/clang/ModuleCache -fmodules-validate-system-headers -fobjc-runtime=macosx-12.0.0 -fobjc-exceptions -fexceptions -fmax-type-align=16
 ↳ -fcommon -fcolor-diagnostics -clang-vendor-feature=+messageToSelfInClassMethodIdReturnType -clang-vendor-feature=+disableInferNewAvailabilityFromInit -clang-vendor-feature=+disableNonDependentMemberExprInCurrentInstantiation
 ↳ -fno-odr-hash-protocols -clang-vendor-feature=+enableAggressiveVLAFolding -clang-vendor-feature=+revert09abecef7bbf -clang-vendor-feature=+thisNoAlignAttr -clang-vendor-feature=+thisNoNullAttr -mllvm -disable-aligned-alloc-awareness=1
 ↳ -D__GCC_HAVE_DWARF2_CFI_ASM=1 -o /var/folders/p3/950_j_ps7xq47vrzdj1qvg940000gn/T/main-67f3af.o -x objective-c main.m


"/Applications/Xcode-14.0.0.app/Contents/Developer/Toolchains/XcodeDefault.xctoolchain/usr/bin/ld" -demangle -lto_library /Applications/Xcode-14.0.0.app/Contents/Developer/Toolchains/XcodeDefault.xctoolchain/usr/lib/libLTO.dylib -no_deduplicate -dynamic -arch
 ↳ arm64 -platform_version macos 12.0.0 12.3 -syslibroot /Applications/Xcode-14.0.0.app/Contents/Developer/Platforms/MacOSX.platform/Developer/SDKs/MacOSX.sdk -o executable -L
 ↳ /usr/local/lib /var/folders/p3/950_j_ps7xq47vrzdj1qvg940000gn/T/main-67f3af.o -lSystem /Applications/Xcode-14.0.0.app/Contents/Developer/Toolchains/XcodeDefault.xctoolchain/usr/lib/clang/14.0.0/lib/darwin/libclang_rt.osx.a
```

Aside

Yes, this means that clang itself is a build system.

Pretty meta, isn't it?

Build systems help manage that overwhelming list of flags that are passed to the build tools,

figuring out which tools turn which inputs into which outputs,

and coordinate the pipeline of build tool invocations so everything is built in the right order,

every time you hit Build.

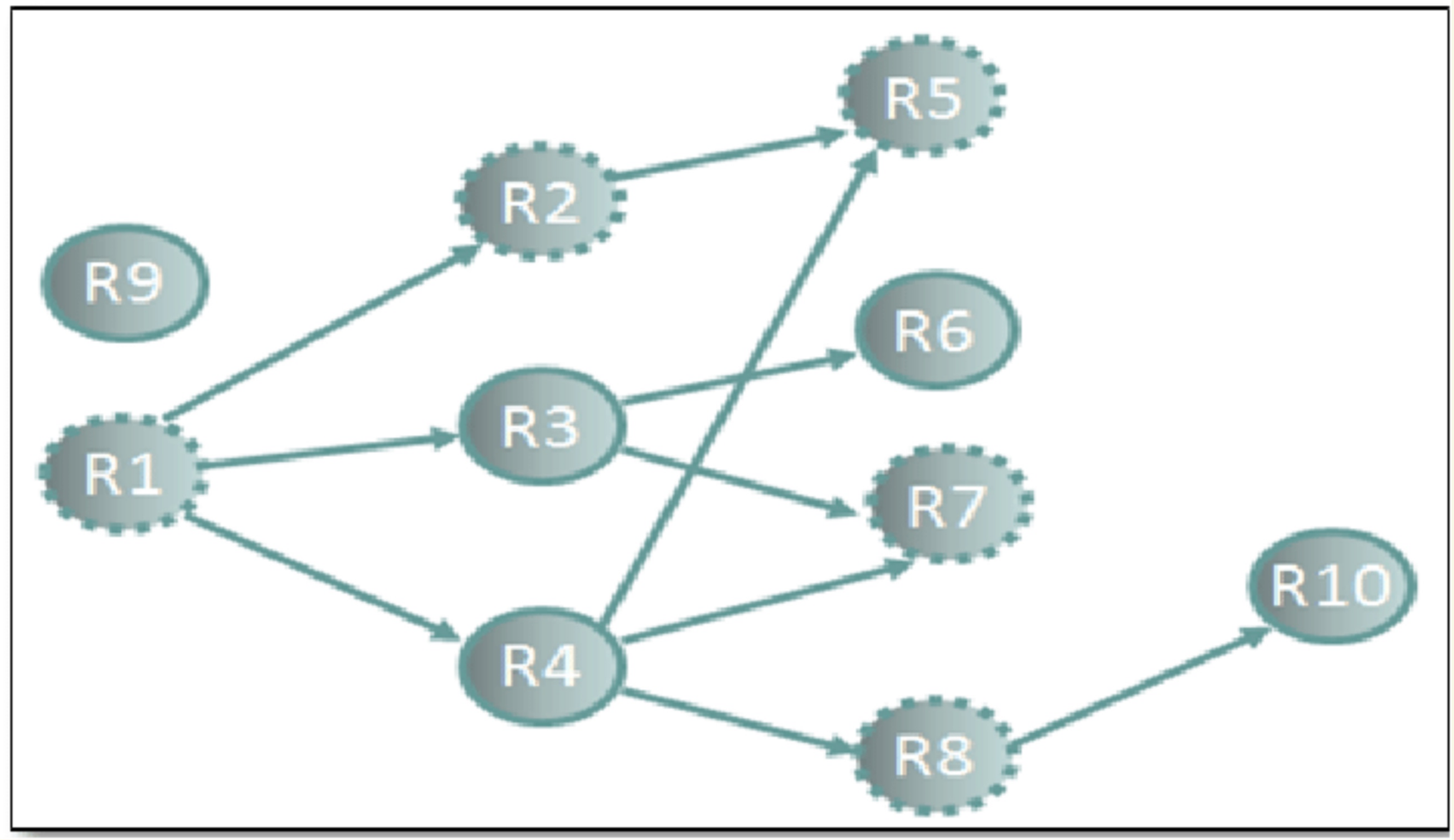
Describing a build

- Project.pbxproj
- Package.swift
- Pod.podspec

What do all these formats have in common?

- Describe one or more targets
- List files compiled into each target
- Dependencies
- What gets produced

Executing a build


R5
R2
R9
R6
R3
R1
R7
R10
R4
R8

Read build description

Find necessary tasks that are unblocked

Build those tasks

GOTO 2 as long as our "goal" product is not yet built & there are no errors

Sounds simple, right?

What comes next