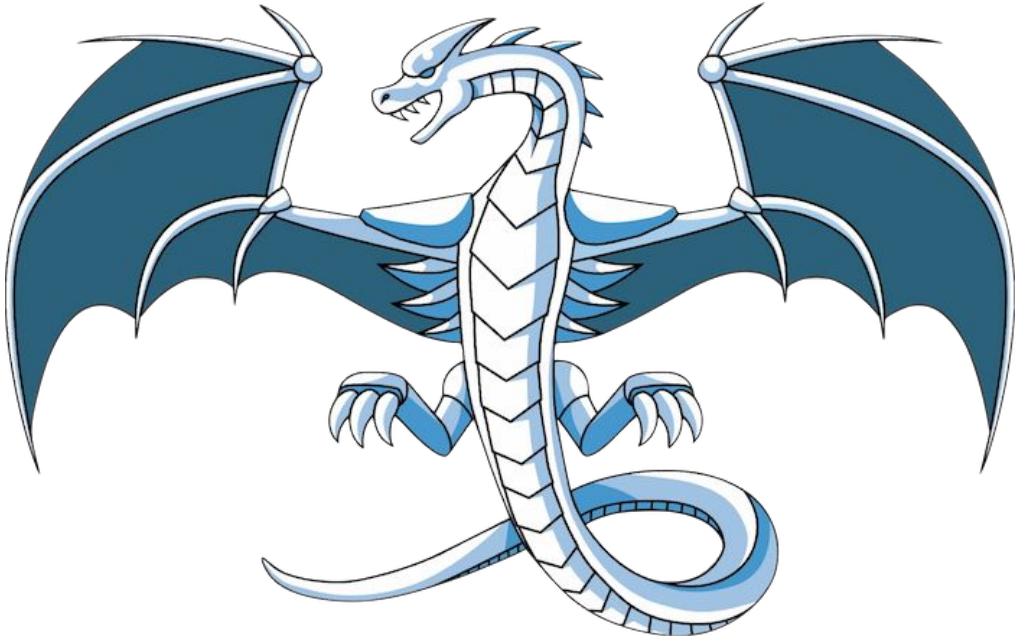
why and how to use clang compiler with Yocto Project

KHEM RAJ

Yocto Project Summit, 2021.11

Agenda

- Introduction to Clang
- Clang integration
- meta-clang
- Use cases
- Future
- QA



Clang /ˈklæŋ/

- **Compiler Frontend, uses LLVM backend**
- **Supported Languages**
  - C/C++, Objective-C/C++, OpenMP, OpenCL, RenderScript, CUDA, Fortran
- **Additional compiler analysis tools**
  - Static Analyzer
  - Code analysis tools
- **Uses Apache-2.0 Licence**

Clang /ˈklæŋ/

- **Written in C++**
- **Supported Platform Architectures**
  - Aarch64, ARM, IA-32, x86_64, RISCV32/RISCV64 MIPS/MIPS64, PPC64
- **Code**
  - https://github.com/llvm/llvm-project
- **Homepage**
  - https://clang.llvm.org/
  -

Clang Goals

- Better Diagnostics
- IDE integration
- Licence compatible with Commercial products
- Developer Friendly
- Fast and low memory use
- GCC compatible
- Modular

Try below samples with Clang & GCC - What do you see?

$ gcc -fsyntax-only <*input*>
$ clang -fsyntax-only <*input*>

```
template<class T>
class a {};
struct b {}
a<int> c;
```

attention_to_details.cpp

```
typedef struct point {
     int x;
     int y;
} point;

struct point origin = { x: 0.0,
y: 0.0 };
```

fixit.c

```
#include <stdio.h>
int foo() {
     printf("%.*d");
     return 0;
}
```

format.c

Clang support in Yocto Project

- **Support via Independent metadata layer**
  - meta-clang - https://github.com/kraj/meta-clang

Using meta-clang

- Adding layer

```
$ git clone https://git.yoctoproject.org/git/poky
$ cd poky
$ git clone git://github.com/kraj/meta-clang.git

$ . ./oe-init-build-env

$ bitbake-layers add-layer ../meta-clang
```

Using meta-clang

- **Default compiler remains gcc**
- **Chosing clang as compiler**
  - Single component - Can be done at recipe level
  - If set in local.conf, it will switch system defaults to clang
- **Some packages can not use clang - see conf/nonclangable.conf**

```
TOOLCHAIN = “clang”
```

Using meta-clang

- **Default runtime remains GNU GCC runtime**
- **Chosing LLVM runtime**
  - LIBCPLUSPLUS - Standard C++ runtime
  - COMPILER_RT - Compiler C runtime
  - UNWINDLIB - System unwinder

```
RUNTIME = “llvm”
```

Using meta-clang

- Default C++ Standard Library Switch

```
LIBCPLUSPLUS = "-stdlib=libc++"
```

- Per package selection (recipe level)

```
LIBCPLUSPLUS:toolchain-clang:pn-<recipe> = "-stdlibc=libc++"
```

Using meta-clang

- Generating SDK

```
CLANGSDK = "1"
```

Why We need Clang

- **Some key packages have switched exclusively to use clang compiler**
  - Chromium, bcc, bpftrace, SPIRV-LLVM-Translator android-tools
- **Some prefer clang as first choice**
  - Firefox, kernel-selftest

Why We need Clang

- **Additional Tools**
  - Static analyser

```
INHERIT += "scan-build"
SCAN_BUILD ?= ""
SCAN_BUILD:pn-openssl = "1"
```

```
$ bitbake -cscanbuild openssl
$ bitake -cscanview openssl
```

Why We need Clang

openssl-3.0.0

User:	kraj@apollo
Working Directory:	/mnt/b/yoe/master/build/tmp/work/cortexa72-yoe-linux/openssl/3.0.0-r0/build
Command Line:	make -j 44
Clang Version:	clang version 13.0.1 (https://github.com/llvm/llvm-project 08e3a5ccd952edee36b3c002e3a29c6b1b5153de)
Date:	Mon Nov 29 06:29:27 2021

Bug Summary

Bug Type	Quantity	Display?
All Bugs	**320**	☑
API		
Argument with 'nonnull' attribute passed null	10	☑
Logic error		
Assigned value is garbage or undefined	3	☑
Dereference of null pointer	20	☑
Result of operation is garbage or undefined	7	☑
Uninitialized argument value	8	☑
Unused code		
Dead assignment	127	☑
Dead increment	5	☑
Dead initialization	5	☑
Dead nested assignment	135	☑

Reports

Bug Group	Bug Type	File	Function/Method	Line	Path Length			
API	Argument with 'nonnull' attribute passed null	crypto/param_build.c	param_bld_convert	333	21	View Report	Report Bug	Open File
API	Argument with 'nonnull' attribute passed null	crypto/param_build.c	param_bld_convert	339	23	View Report	Report Bug	Open File
API	Argument with 'nonnull' attribute passed null	crypto/cms/cms_enc.c	ossl_cms_EncryptedContent_init_bio	165	43	View Report	Report Bug	Open File
API	Argument with 'nonnull' attribute passed null	crypto/param_build.c	param_bld_convert	331	21	View Report	Report Bug	Open File
API	Argument with 'nonnull' attribute passed null	test/params_conversion_test.c	param_conversion_load_stanza	136	17	View Report	Report Bug	Open File
API	Argument with 'nonnull' attribute passed null	ssl/statem/statem_dtls.c	dtls1_buffer_message	1054	30	View Report	Report Bug	Open File
API	Argument with 'nonnull' attribute passed null	include/internal/packet.h	PACKET_contains_zero_byte	460	38	View Report	Report Bug	Open File
API	Argument with 'nonnull' attribute passed null	test/endecode_test.c	check_protected_legacy_PEM	747	18	View Report	Report Bug	Open File
API	Argument with 'nonnull' attribute passed null	test/ssl_old_test.c	verify_alpn	348	57	View Report	Report Bug	Open File
API	Argument with 'nonnull' attribute passed null	crypto/param_build.c	param_bld_convert	341	25	View Report	Report Bug	Open File
Logic error	Assigned value is garbage or undefined	crypto/asn1/a_object.c	a2d_ASN1_OBJECT	166	68	View Report	Report Bug	Open File

Why We need Clang

- **Additional Tools**
  - LLDB - Debugger
    - Provides a target agent lldb-server like gdbserver
    - Powerful CLI, python integration
  - LLD - Linker
    - Very fast linker
    - Helps large binaries link times
  - Clang-format
  - Clangd
  - Clang-tidy

Status of Clang compiled packages in Yocto

- **World builds**
  - qemuarm, qemuarm64, qemux86, qemuriscv64, qemuriscv32
- **Distributions**
  - Yoe Distro uses clang as default compiler
- **Packages needing GCC**
  - Kernel
    - ARM/x86/RISCV ready to use clang
  - Glibc
  - GCC + runtime
  - Bootloaders e.g. syslinux
    - u-boot can use clang
- **Complete list**
  - https://github.com/kraj/meta-clang/blob/master/conf/nonclangable.conf

Limitations

- **Used for target packages only**
  - chromium does use native clang + runtime
- **Relocatable SDKs may not work**
  - Needs special patch to extend ldso paths in binaries
- **Not all target packages will be able to use clang**
  - GCC'ism
  - Dependency on language Undefined behaviors
- **GCC supports many more architectures than Clang**
  - https://gcc.gnu.org/backends.html
  - Csky, ARC — no luck
- **Reproducible builds are not tested hence may not work**

Wish List

- Use Clang to compile Kernel
- Permanent OE core switch to select compiler
  - TOOLCHAIN variable like
- Yocto Autobuilder Testing
- Benchmarks
- MingW SDK port
- LLVM runtime as default
- List - https://github.com/kraj/meta-clang/issues?q=is%3Aissue+is%3Aopen+label%3Aenhancement
- Enable Extra tools e.g. clang-format etc.
- Enable LTO by default

Thanks for your time!

yocto PROJECT | THE LINUX FOUNDATION