“Which targets does Clang support?”

EuroLLVM 2014: Lightening Talk
Jonathan Roelofs
jonathan@codesourcery.com

mentor.com/embedded
“Which targets does Clang support?”

“Some stuff doesn’t seem to be documented at ALL…. what are the valid inputs to the ‘-arch’ … option? … This really is frustrating.”

– Tim Hill [1]
“Which targets does Clang support?”

“I read the man page … but I haven't been able to find a list of what ‘-march’ options are available…. Could someone point me to a list of supported options?”

– Tim Nackos [2]
“Which targets does Clang support?”

“I think the best way to get the answer is reading the source” – a’Q [3]
“Which targets does Clang support?”

Clearly we need a better answer!
Prior Work

LLVM (http://llvm.org/):
LLVM version 3.5.0svn
DEBUG build with assertions.
Default target: x86_64-apple-darwin13.1.0
Host CPU: corei7

Registered Targets:
  aarch64 - AArch64 (ARM 64-bit little endian target)
  aarch64_be - AArch64 (ARM 64-bit big endian target)
  arm - ARM
  cpp - C++ backend
  hexagon - Hexagon
  mips - Mips
  mips64 - Mips64 [experimental]
  mips64el - Mips64el [experimental]
  mipsel - Mipsel
  msp430 - MSP430 [experimental]
  nvptx - NVIDIA PTX 32-bit
  nvptx64 - NVIDIA PTX 64-bit
  ppc32 - PowerPC 32
  ppc64 - PowerPC 64
  ppc64le - PowerPC 64 LE
  r600 - AMD GPUs HD2XXX-HD6XXX
  sparc - Sparc
  sparcv9 - Sparc V9
  systemz - SystemZ
  thumb - Thumb
  x86 - 32-bit X86: Pentium-Pro and above
  x86-64 - 64-bit X86: EM64T and AMD64
  xcore - XCore
Prior Work

$> \text{clang -target <foo> --print-multi-libs}$

(Based on patches I submitted earlier this spring)
Universal Driver

“Clang is inherently a cross compiler…. However, actually cross compiling in practice involves much more than just generating the right assembly”

– Daniel Dunbar [4]
Proposed Solution

Target Triple: `<arch><sub>-<vendor>-<sys>-<abi>`

--print-supported-archs
--print-supported-vendors
--print-supported-systems
--print-supported-abis
--print-multi-libs
--print-available-targets
Proposed Solution: Examples

$> $ clang --print-supported-archs
x86  
...

$> $ clang -march x86 --print-supported-systems
auroraux  
darwin  
macosx  
...

$> $ clang -march x86 --print-available-systems
linux
Proposed Solution: Examples

```bash
$> clang --print-supported-targets
x86-linux-gnu
ppc-apple-darwin
arm-none-eabi
$> clang --print-available-targets
x86-linux-gnu
$> clang -target ppc-apple-darwin foo.c
```

Sorry, but the toolchain for: ppc-apple-darwin has not been installed.
Conclusion

It should be simple to ask Clang which targets it could support, and of those, which ones it does support.
Thank you!
Backup Slides
Bibliography