Cross-platform C++ development is challenging

Let tools help!

Marc Goodner

What are challenges with Cross Platform C++?

- Libraries
- Compilers
- Build
- Testing
- Debugging
- Editing
- Host

Libraries and Compilers

- System provided package mangers are not ideal for libraries
  - Libraries that come from the system package manager are for the OS as built
  - Some operating systems don’t have a package manager at all
- Manually manage libraries from source
  - Pro: Complete control
  - Con: Complete control
- C++ has no package manager, much is happening in the ecosystem
- Compilers are more consistent with supporting the standard today

Build

- CMake
  - Becoming the defacto build system for C++
  - Ideally no platform specific info
  - Lots of projects predate improvements in tooling
    - Compilers specified, or assumed
    - Platforms specified throughout
    - Special logic to find packages interferes with new solutions
- MSBuild
  - Windows only for C++… kinda
  - Many people generate sln files from CMake, you don’t need to anymore
- Make and the rest…

Pong

GCC, CMake, vcpkg, Windows and Linux, Visual Studio

Pong: What you saw

- Using vcpkg on Windows and Linux to acquire the SFML library
- CMake with no platform specific switches, build on Linux
- VS Code editing, build, debug C++ on Linux
- Visual Studio using CMake to target both Windows and Linux
- Visual Studio debugging on Windows and Linux
- Visual Studio platform specific IntelliSense

Testing and Debugging

- Do you test on systems other than your host?
- What happens when a test fails on a target that isn’t your host?
  - Windows devs unfamiliar with gdb
  - Linux devs unfamiliar with Windows tools like VS and windbg

Attach to process

Windows devs don’t have to learn gdb to use it

Attach to process: what you saw

- Open folder of just source code and configuring IntelliSense in Visual Studio
- Using SSH in Attach to Process to see processes on remote Linux machine and attach
- Hitting a breakpoint in local source on Windows from an app launched and already running on Linux

Does your host help or hurt?

- Common abilities
  - VMs run other OS
  - Cross compiling
- What is your shell?
  - Linux has bash, powershell and many package managers
  - Windows has bash, powershell, cmd
    - Windows Subsystem for Linux has the package manager of the distro you install
    - Windows doesn’t have an official package manager, use chocolatey
  - Mac has bash
    - MacOS doesn’t have an official package manager, use homebrew
- Containers
  - Linux and Mac can run Linux containers
  - Windows can run Linux and Windows containers

Now what?

- Libraries
  - Evaluate how you manage and pick a tool right for your needs
- Compilers
  - Use the standards
- Build
  - Remove unneeded platform specific logic
- Testing and Debugging
  - Understand where your gaps in your inner loop
  - Know how to test on all platforms at runtime and not just your host
- Editing
  - Is your editor helping you catch issues before you commit?
- Host
  - Understand all your host offers you and what it’s limits are

More info

@visualc

@robotdad

https://blogs.msdn.microsoft.com/vcblog/