

Cross-Compilation

CIS 198 Lecture 18

Rustup (beta)

- Successor to Multirust.
 - Currently in beta (try it out! - but not until after your projects.)
 - Written in Rust, not shell - works natively on Windows!
- Still allows you to easily maintain multiple toolchains.
 - e.g. `stable`, `beta`, `nightly-2016-04-15`, `1.8.0`
- Supports cross-compilation nicely!
 - Can download cross-compilation targets & `std` for other targets.
 - `rustup target list` - OS X, iOS, Windows, Android, etc.
 - `rustup target add x86_64-pc-windows-gnu`
- Can still `override` the toolchain for a particular project:
 - `rustup override add nightly` - overrides current directory
 - `rustup override list` - lists overrides
- Run commands with a particular toolchain:
 - `rustup run nightly COMMAND`

Linux to Windows

- Minimal demo
 - Requires MinGW-w64.
- Add the 64-bit Windows target:
 - `rustup target add x86_64-pc-windows-gnu`
- Configure Cargo to target Windows and link using the MinGW-w64 linker.

- `.cargo/config`:

```
[target.x86_64-pc-windows-gnu]
linker = "/usr/bin/x86_64-w64-mingw32-gcc"
ar = "/usr/bin/x86_64-w64-mingw32-ar"

[build]
target = "x86_64-pc-windows-gnu"
```

- In the future, this may be handled by:
 - `rustup override add stable-x86_64-pc-windows-gnu`
 - (Currently doesn't quite work for me.)

Linux to Windows

- `cargo build`! Build appears at:
 - `target/x86_64-pc-windows-gnu/debug/xcompile-win.exe`
 - PE32+ executable (console) x86-64, for MS Windows
- Executable can be run on Windows (or under Wine).
 - Demo

Linux to Android

- [Minimal demo](#)
- Rustup has a target for this: `arm-linux-androideabi`
 - But this will only get us a bare binary - not an APK.
 - `ELF 32-bit LSB relocatable, ARM, EABI5 version 1 (SYSV)`
- For APKs, we can use `cargo-apk` ([announcement](#)).
 - Compiles Rust code to a shared object (`.so`) file
 - Uses [android-rs-glue](#).
 - Simple Java template for loading a shared object (`.so`) file and dynamically links it from Java.
 - Builds using Android build tools.
 - Host must be Linux x86_64 (currently).

Linux to Android

- `cargo install cargo-apk`
- `Cargo.toml`:

```
[package.metadata.android]
label = "xcompile-198"

[dependencies]
android_glue = "0.1.3"
```

- Build and install:

```
export ANDROID_SDK_HOME=/opt/android-sdk
export NDK_HOME=/opt/android-ndk
cargo apk install
```

- Demo