Libraries

- [Raylib](#)

  - [Definitions when working with Windows.h](#)

  - [CrossCompiling from Linux](#)

  - [LINKING ON BUILD (MUST USE C-STD-LIB)](#)

- [Cross Compiling OPENSSL from WSL/Linux to Windows](#)

- [Open Quantum Safe](#)

  - [Documentation](#)

Raylib

Definitions when working with Windows.h

```
// GODTIER COMMENT DUE TO SDHARED NAMES
// https://github.com/raysan5/raylib/issues/1217

#if defined(_WIN32)
    #define NOGDI          // All GDI defines and routines
    #define NOUSER          // All USER defines and routines
#endif

#include <blibstd.h>
#include <blibhash.h>
#include <blibwin.h>

#if defined(_WIN32)         // raylib uses these names as function parameters
    #undef near
    #undef far
#endif

#include <raylib.h>
```

CrossCompiling from Linux

May need to override AR and CC.

```
sudo make PLATFORM=PLATFORM_DESKTOP PLATFORM_OS=WINDOWS
```

LINKING ON BUILD (MUST USE C-STD-LIB)

You must link to RAYLIB and to GDI32

https://github.com/raysan5/raylib/discussions/2492

-lraylib -lgdi32 -lwinmm

Cross Compiling OPENSSL from WSL/Linux to Windows

Flags for the different binaries need to be overwritten when generating the Makefile with Configure

https://github.com/openssl/openssl?tab=readme-ov-file

https://gist.github.com/udnaan/80c5ad125fc4702309b9

```
./Configure --cross-compile-prefix=x86_64-w64-mingw32- mingw64


make
make install
```

.. . .

So the above was from a guide but at least my BELOW thing compiles...

```
./Configure --static -static --prefix=/opt/Cross-openssl/win64 --opensssldir=/opt/Cross-openssl/win64 mingw64
CC=x86_64-w64-mingw32-gcc RC=x86_64-w64-mingw32-windres AR=x86_64-w64-mingw32-ar
RANLIB=x86_64-w64-mingw32-ranlib
```

The flags -ffunctions-sections and -fdata-sections help reduce static linking size, otherwise your binary will be like 6mb with no content of lib.

`-rwxrwxrwx 1 lepus lepus 6.0M Aug 14 19:34 bunpass.exe`

Consider this below to reduce by ~2mb.

```
./config no-ssl2 no-ssl3 no-comp no-hw no-engine no-async no-err no-ocsp no-psk no-srp no-tests no-idea no-
md2 no-md4 no-mdc2 no-rc2 no-rc4 no-rc5 no-rmd160 no-whirlpool no-dso no-ec no-ecdsa no-ecdh no-sm2 no-
sm3 no-sm4 no-camellia no-seed no-bf no-cast no-des no-dh no-dsa no-ssl no-tls no-aria no-chacha no-poly1305
no-siphash no-siv no-ct no-aria no-ktls no-srtp no-ssl-trace no-ssl3-method no-weak-ssl-ciphers no-ssl3 no-tls1
no-tls1_1 no-tls1_2 no-tls1_3 no-dtls no-dtls1 --static -static --prefix=/opt/Cross-openssl/win64 --
opensssldir=/opt/Cross-openssl/win64 mingw64 CC=x86_64-w64-mingw32-gcc RC=x86_64-w64-mingw32-
windres AR=x86_64-w64-mingw32-ar RANLIB=x86_64-w64-mingw32-ranlib
```

```
make install_dev
```

Ref:

https://github.com/openssl/openssl/issues/9922

Open Quantum Safe

Lean Post-Qunatum lib

Documentation

https://openquantumsafe.org/liboqs/api/

Building was super easy with cross-compilation.

https://github.com/open-quantum-safe/liboqs/wiki/Platform-specific-notes-for-building-liboqs#cross-compiling