C Archives and Linking

Things I wish I had known ahead of time.

Archives

Generating Archives

The archive tool `ar` can be used to pack object files or shared object files into a singular archive for linking purposes.

```
# Compiling some source, such as blib
# Note, the -c flag compiles, but does not perform any form of linking.
x86_64-w64-mingw32-gcc -nostdlib -shared -fFPIC -fshort-enums -nostdlib -c $(SRC)

# Compile into an archive file
x86_64-w64-mingw32 -rcs ./export/lib/libblib.a $(wildcard ./bin/*.o)

# At this point you can link against the archive. Just ensure that any dependencies also linked are included in order for the linker. Since linking hasn't been performed in blib, this needs to be done after the fact.
# It is ideal to copy this to somewhere, like /usr/local/lib with sudo.
```

When GCC/MINGW CC searches for a linked library it will search the -L path for the proper extension for the system (.dll/.so), then lib archives such as (.dll.a, .a, lib<lib>.dll/so/a).

> Linking is evaluated from left to right, including source code.
> If the source code of the binary 'a.c' requires linking to a library such as blib, and blib requires libraries such as kernel32.dll, the order in the compiler and linker should be:
> `$(CC) a.c -L/usr/local/lib -lblib -lkernel32 -luser32`
> This allows blib a.c to link to blib to link to kernel32/user32.