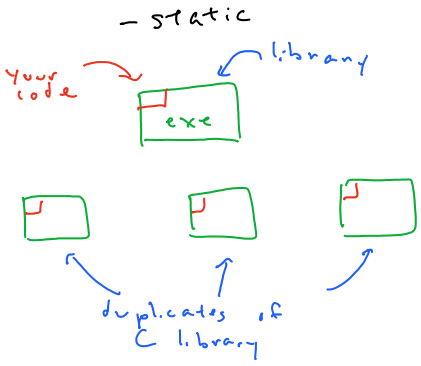
33 C Library

Friday, November 20, 2015 09:58

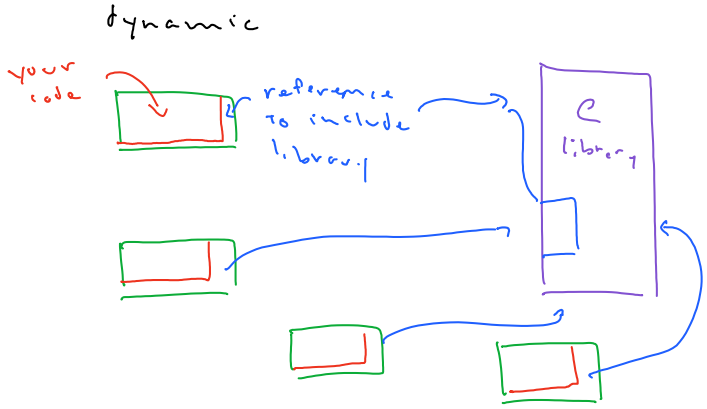


What the C library actually does
Hello.c
Once compiled, 7,000 bytes

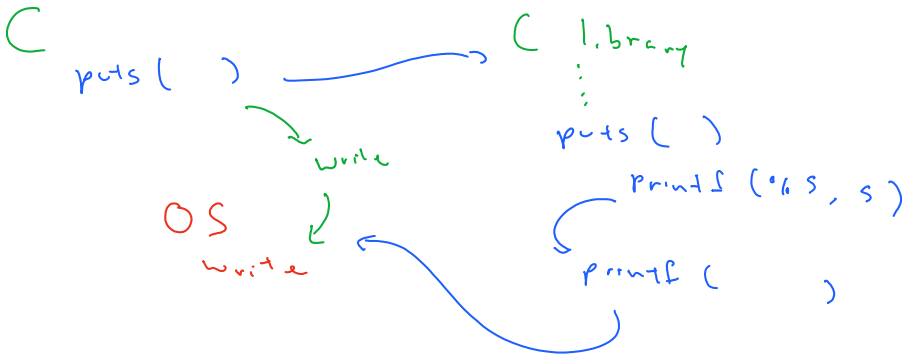
When compiled with -static
Size is now 740,000 bytes

What does - static actually do?

- Without -static, linker uses library on system, but with -static, it pulls the relevant code into the program
- If you always compile with static, you'll have many programs with copies of the C library in it. Wasting disk space.
- Once you bake the library into the program, can't update the library in that program.
- Thus the default mode these days is dynamically, no -dynamic flag, because it's the default
    - The linker creates an exe that's mostly your code but also a little reference that says, in order to run, need a copy of the C library to run.



Now you can upgrade library independent of your programs



Replace
Puts("Hello World!");

with
Write(1, "Hello World!\n", 13);
Works!

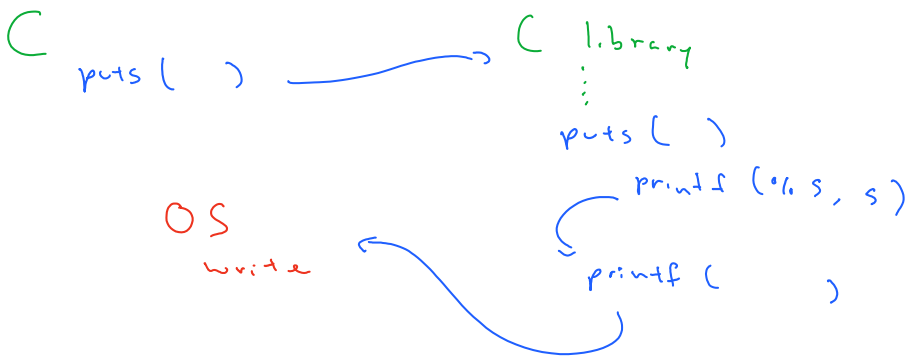
Compiled with -static still 700 k bytes
Without -static, now 7k bytes, not much smaller from our original dynamic..

Strings hello,
See what linker is doing,
Program still has reference for c library, **but why?**

Objdump hello

1. Theres a call. Write is not a system call, still a c call

So "write" has a wrapper in C, so you need the C library



2. Also who calls the main function?

Startup code, comes from C library

Once we get rid of both, its only 700 bytes
But now its tied to x86, because we use x86 assembly