

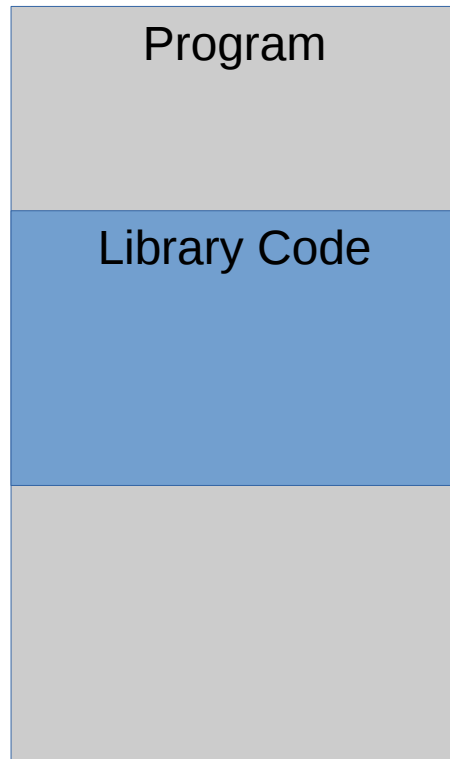
Dynamic Linking

Daniel Townley

Dynamic Linking

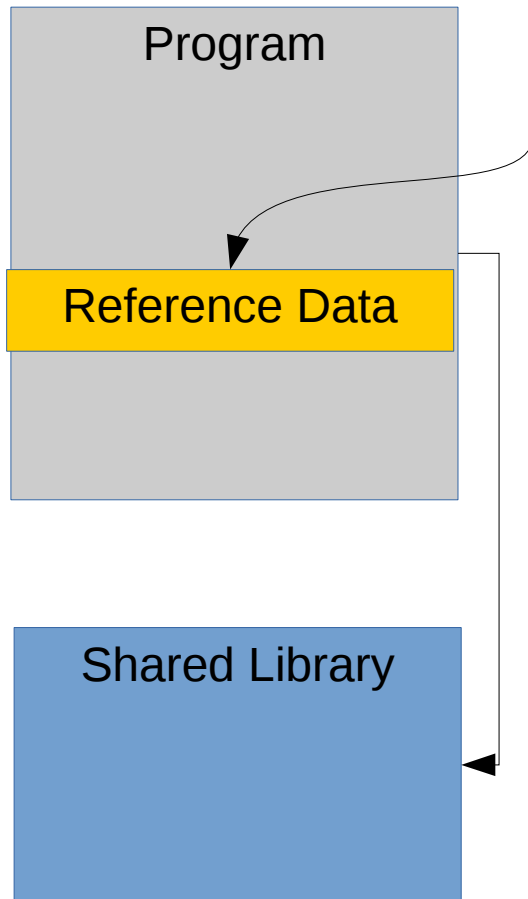
- Linking performed at load or runtime
- Involves the OS
- Libraries shared by multiple programs
 - DLL (Windows)
 - Shared Objects (Linux)
- Confers multiple advantages
 - Memory management
 - Development and maintenance

Static Linking



- Performed before loading
- Library code copied into each program
- All instructions must be present at load time

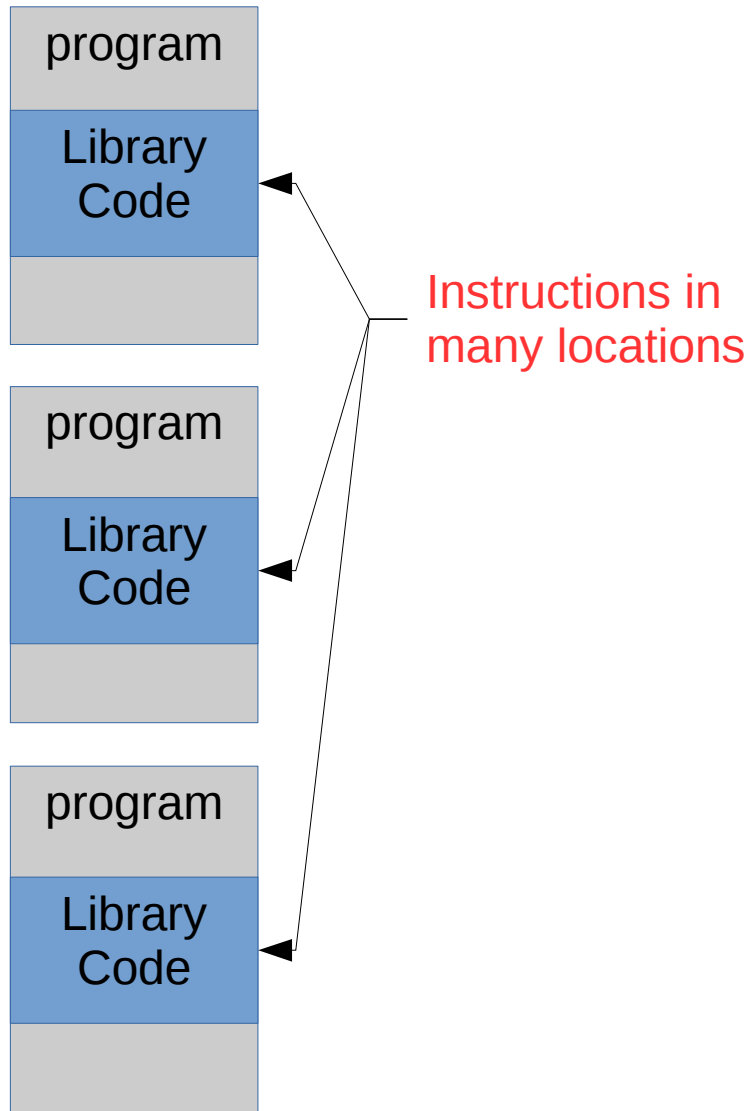
Dynamic Linking



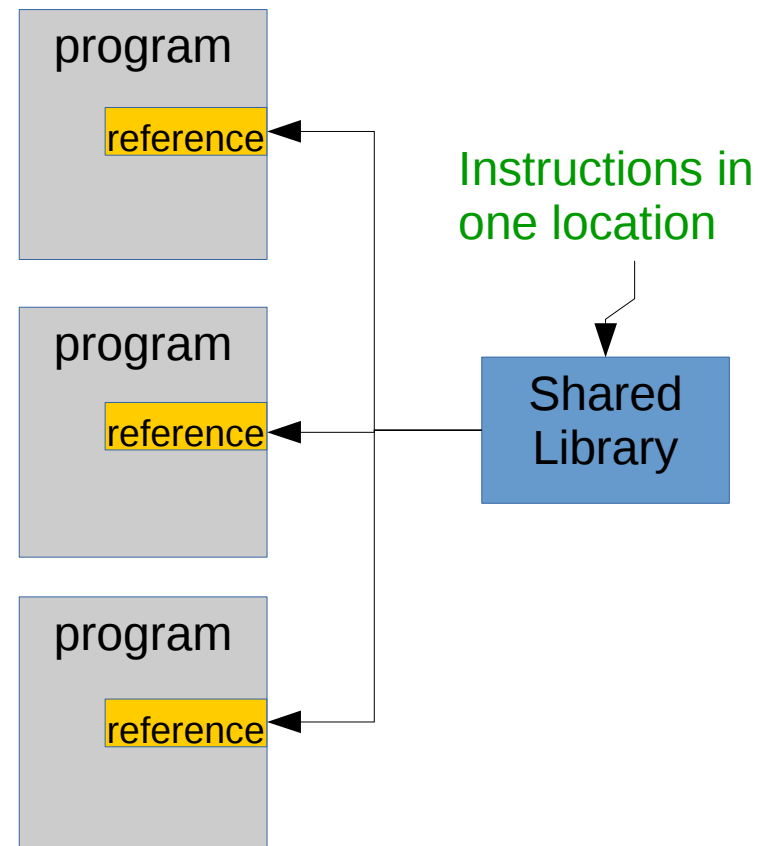
- Reference to shared library added before loading
- Need not contain all instructions at load time
- OS can resolve reference to a shared library at load or run time

Advantage: Less Memory Usage

Static Linking

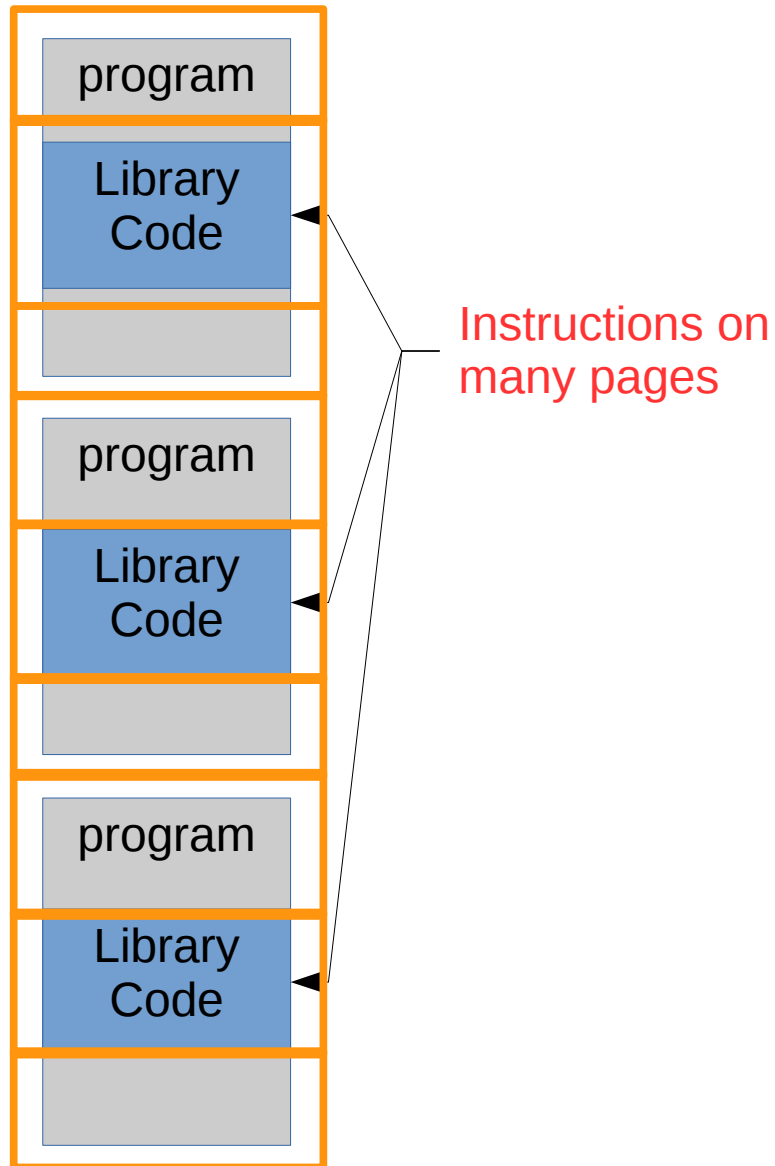


Dynamic Linking

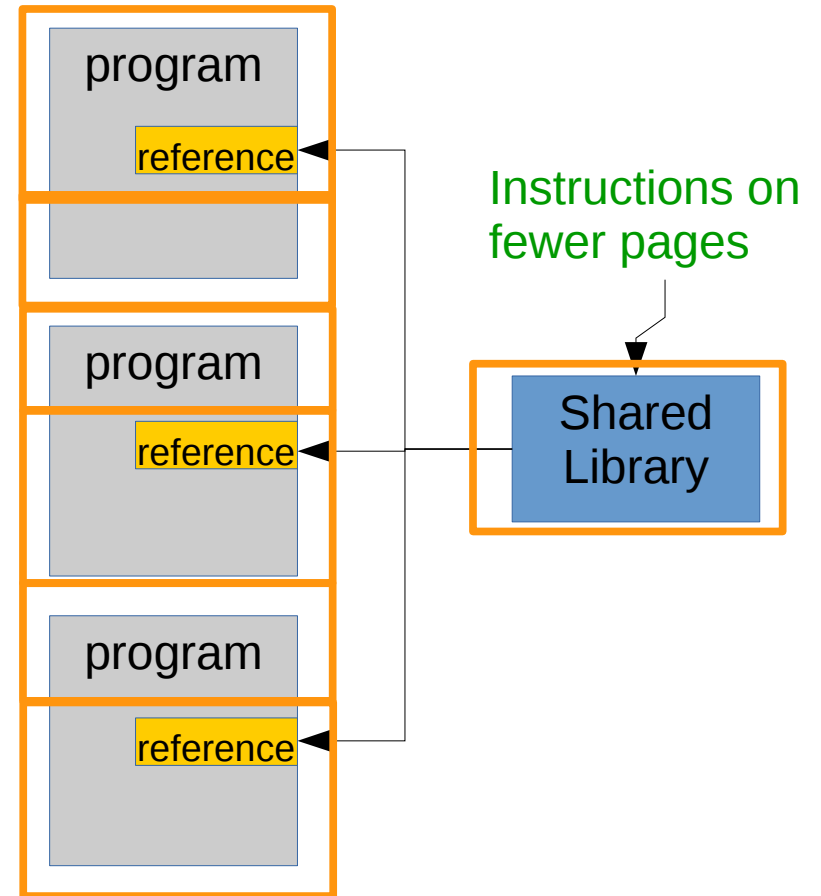


Advantage: Less Swapping

Static Linking



Dynamic Linking



Advantage: Modularity

- Standard functionality
- Can simplify updates
- Some caveats: “DLL Hell”
 - Programs may require different versions of shared libraries
 - Updates could create compatibility issues

Dynamic Linking: Summary

- A valuable OS service to programming languages
- Can improve program performance
- Can simplify programming, if handled correctly

References

Stallings. William. “Memory Management” in *Operating Systems, Internals and Design Principles*, 7th ed., Prentice Hall, 2011.

Bryant, Randal and David O'Hallaron. “ Linking” in *Computer Systems, A Programmers Perspective*, 2nd ed., Pearson, 2010.