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

- Program
- Library Code

Dynamic Linking

- Program
- Reference
- Shared Library

Instructions in many locations

Instructions in one location
Advantage: Less Swapping

Static Linking

- Program
  - Library Code

Dynamic Linking

- Program
  - Reference

Instructions on many pages

Instructions on fewer pages

Shared Library
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
