

Vorlesungsskript zu „Vertiefung Programmieren“ Bibliotheken

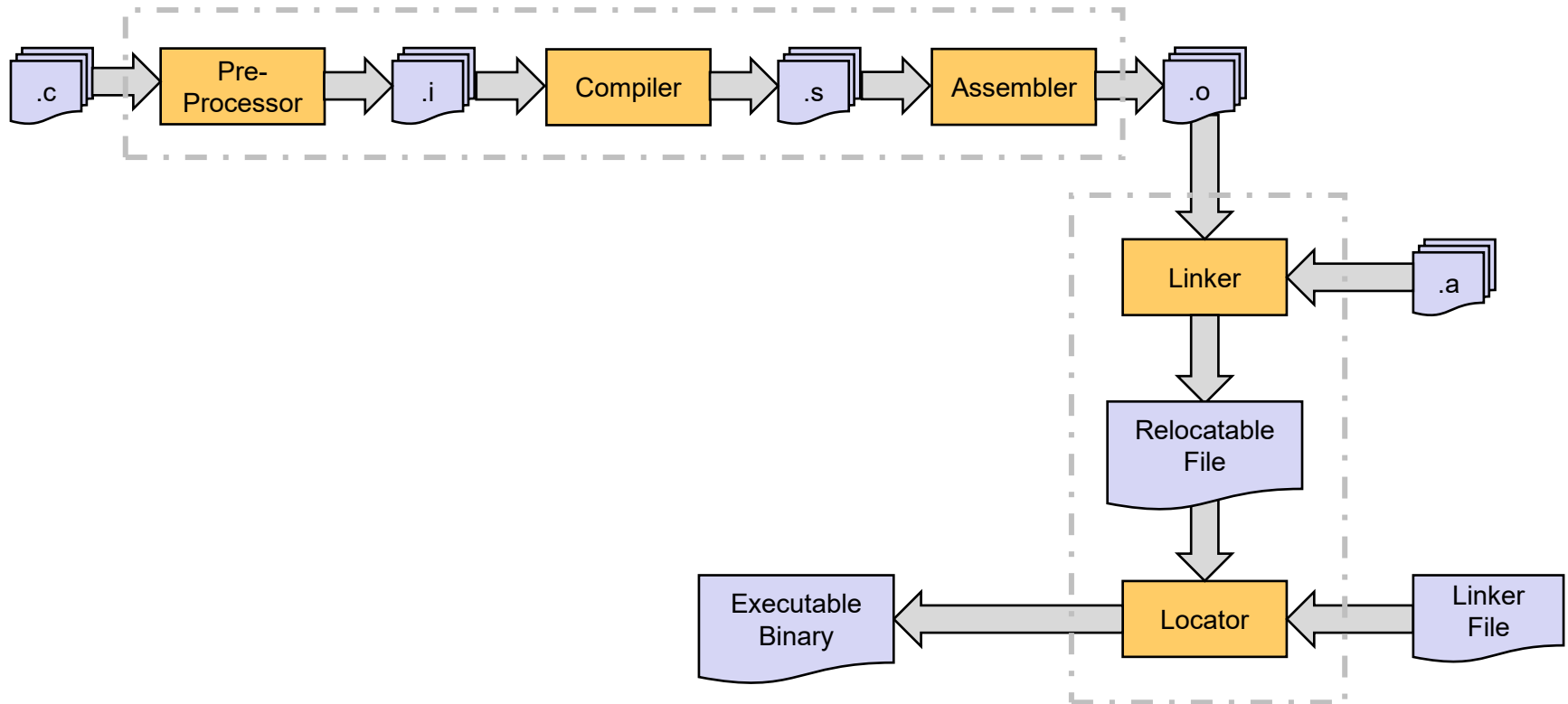


Dozent: Dipl.-Inf. (FH) Andreas Schmidt

Bibliotheken

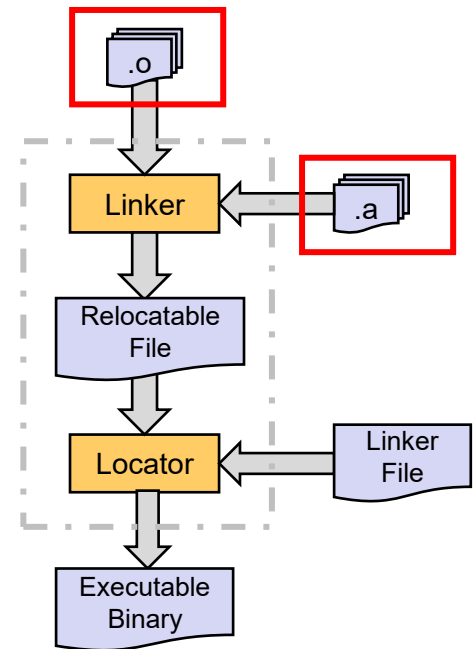
Statische und dynamische Bibliotheken

Grundsätzlicher Aufbau einer „C/C++ Compiler Toolchain“



Linker & Locator

- Linker erstellt aus den Object-Files und statischen Bibliotheken ein sog. Relocatable-File (zumeist nur intern)
- Hierbei führt der Linker folgende Aktionen durch
 - Symbol Resolution
 - Section Merging
- Der Locator führt dann basierend auf dem „Linker File“ ein sog. Section Placement durch
 - Section Placement



Statische Bibliothek

- Eine statische Bibliothek (zumeist .a oder .lib Dateiendung), ist eine Sammlung von Object-Files in einer Art Container.
- Dies ermöglicht dem Linker den Zugriff auf die einzelnen Object Files, ohne das jedes Object File einzeln angegeben werden muss
- Deklarationen (in Form von Header-Dateien .h) müssen weiterhin in Source-Code Form bereitgestellt werden.
 - ➔ eine bereitgestellte statische Bibliothek besteht immer aus Header- (.h) und Library-Dateien (.a/.lib)
- Der Linker nutzt aus den Bibliotheken nur die verwendeten Code-Teile und integriert diese in das „Executable Binary“
 - ➔ kein „toter Code“ durch Bibliotheken, duplizierter Code durch mehrfache Verwendung in unterschiedlichen „Executable Binaries“

Dynamische Bibliothek

- Dynamische Bibliotheken (häufig .so oder .dll Dateiendung) werden zur Laufzeit geladen.
- Deklarationen (in Form von Header-Dateien .h) müssen weiterhin in Source-Code Form bereitgestellt werden.
 - ➔ eine bereitgestellte dynamische Bibliothek besteht immer aus Header- (.h) und Library-Dateien (.so/.dll)
- Der Linker fügt keine Code-Teile aus der Bibliothek in das „Executable Binary“ ein
 - ➔ kein „toter Code“ durch Bibliotheken, keine Duplizierung durch mehrfache Verwendung