

# **Vorlesungsskript zu** **„Vertiefung Programmieren“** **GNU Toolchain**



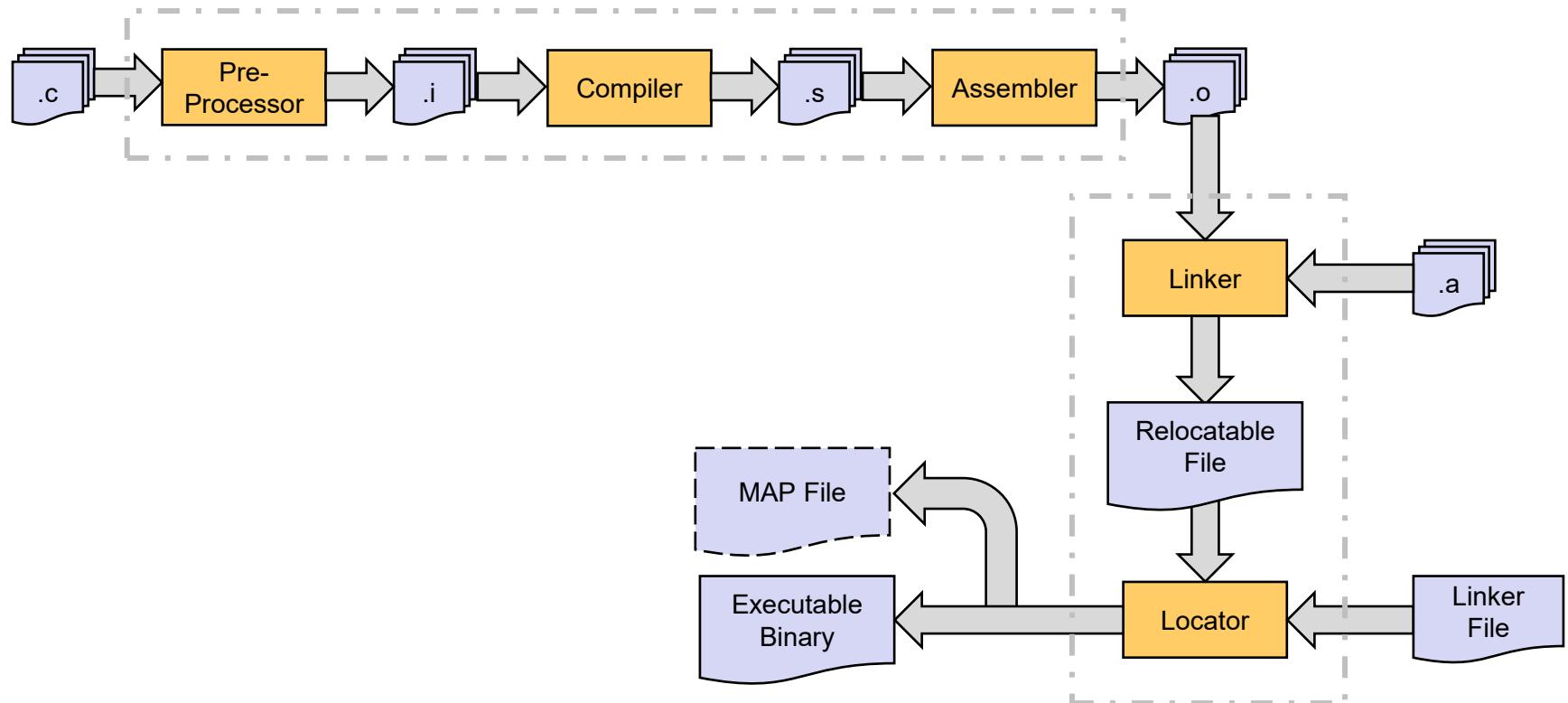
**Dozent:** Dipl.-Inf. (FH) Andreas Schmidt

# Compiler Toolchain

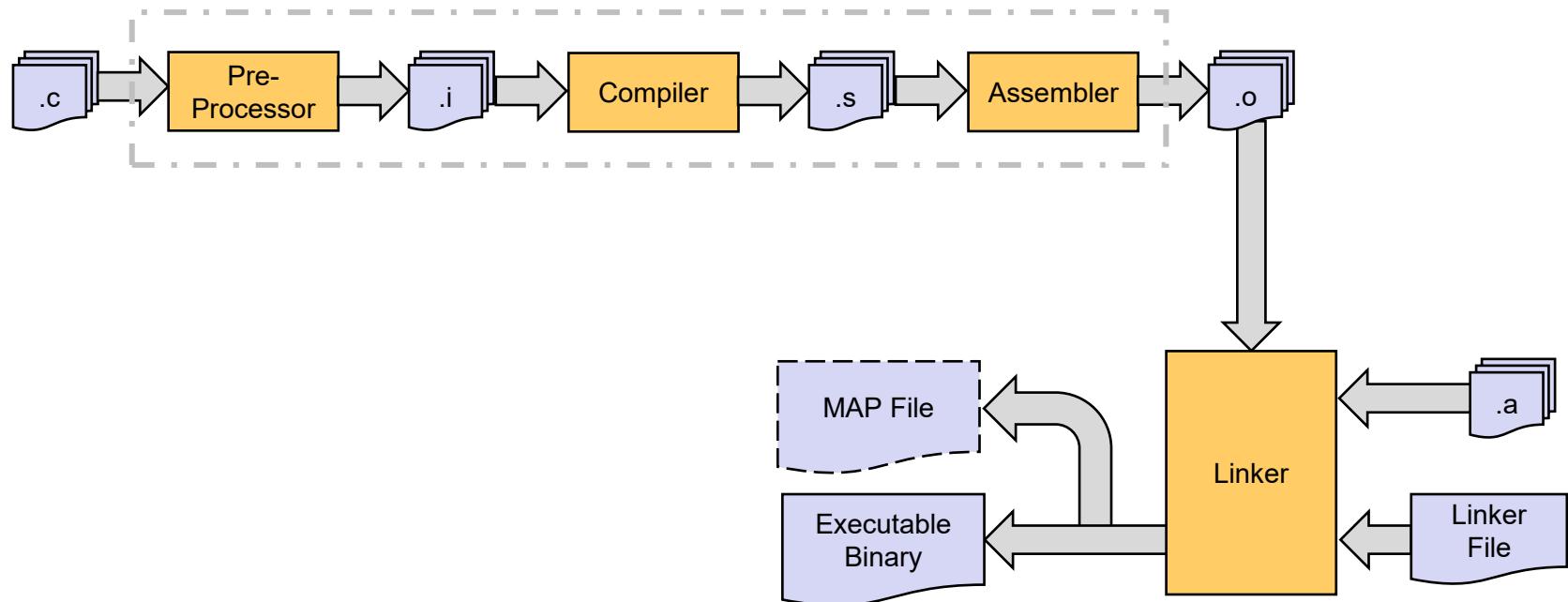
Präprozessor, Compiler, Assembler und Linker

GNU Toolchain

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

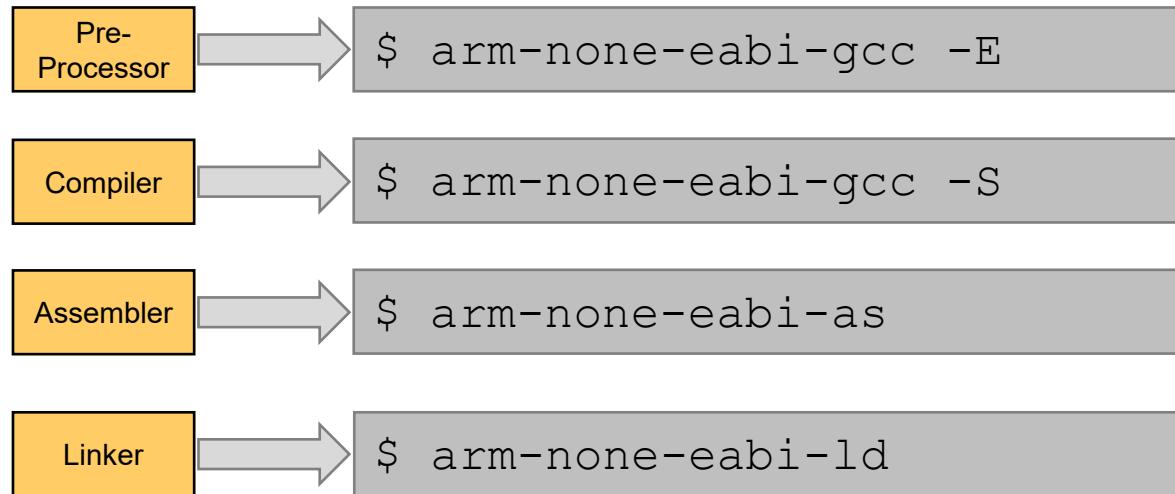


## Grundsätzlicher Aufbau der „GNU C/C++ Compiler Toolchain“



# GNU Compiler Toolchain

## Programme der „GNU C/C++ Compiler Toolchain“



## Pre-Processor der „GNU C/C++ Compiler Toolchain“



Der Pre-Processor ist in der GNU Toolchain kein separates Programm, sondern Teil des C-Compilers und kann mit folgendem Aufruf gestartet werden.

```
$ arm-none-eabi-gcc -E <Input C-File> -o <Output File>
```

-E                    Aktiviert nur den Pre-Processor

<Input C-File>    C-Quelltext der verarbeitet werden soll

<Output File>    Dateiname des „pre-processed“ Quelltext

Optional:

-D<Symbol>      Definiert ein Präprozessor Symbol (vgl. #define <Symbol>)

## Compiler der „GNU C/C++ Compiler Toolchain“

Compiler

→ \$ arm-none-eabi-gcc -S

Der Compiler der GNU Toolchain kann mit folgendem Aufruf gestartet werden. (Hinweis: Üblicherweise führt der GNU Compiler die Schritte einer Toolchain aufeinmal durch, wenn dies nicht explizit durch entsprechende Argumente gesteuert wird)

```
$ arm-none-eabi-gcc -S <Input C-File> -o <Output Asm File>
```

-S Führt Pre-Processor + Compile-Vorgang durch

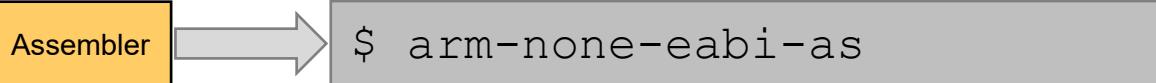
<Input C-File> C-Quelltext der verarbeitet werden soll

<Output File> Dateiname des erzeugten Assembler Files

Optional:

-D<Symbol> Definiert ein Präprozessor Symbol (vgl. #define <Symbol>)

## Assembler der „GNU C/C++ Compiler Toolchain“



Der Assembler der GNU Toolchain kann mit folgendem Aufruf gestartet werden.

```
$ arm-none-eabi-as <Input Asm File> -o <Output Obj File>
```

<Input Asm File> Assembler-Quelltext der übersetzt werden soll

<Output Obj File> Dateiname des erzeugten Object Files

## Linker der „GNU C/C++ Compiler Toolchain“

Linker



\$ arm-none-eabi-ld

Der Linker der GNU Toolchain kann mit folgendem Aufruf gestartet werden.

```
$ arm-none-eabi-ld -T <LinkerFile> -o <Output> <<Obj Files>>
```

<-T <LinkerFile> Dateiname des Linker Files das verwendet werden soll  
<Output> Dateiname des erzeugten ELF Binary

Optional:

-Map <MapFile> Dateiname der zu erzeugenden Map-Datei  
-nostdlib Keine Verwendung der C-Standard Bibliotheken

## Linker der „GNU C/C++ Compiler Toolchain“

Die GNU Toolchain bietet auch eine umfangreiche Dokumentation mit allen notwendigen Infos zu

- Kommandozeilen Optionen
- Optimierungsstufen
- Etc.
- **C/C++-Compiler**
- <https://gcc.gnu.org/onlinedocs/gcc/Invoking-GCC.html>
- **Pre-Processor**
- <https://gcc.gnu.org/onlinedocs/cpp/>
- **Assembler & Linker**
- <https://www.gnu.org/software/binutils/>
  - <https://sourceware.org/binutils/docs/as/index.html>
  - <https://sourceware.org/binutils/docs/ld/index.html>
  -

## Zusätzliche Tools

Tool	Einsatzweck
arm-none-eabi-nm	Listet Symbole in einem Binary auf
arm-none-eabi-objcopy	Daten aus einem Binary extrahieren/modifizieren
arm-none-eabi-objdump	Daten und Meta-Daten aus einem Binary extrahieren
arm-none-eabi-readelf	Daten und Meta-Daten aus einem ELF Binary extrahieren
arm-none-eabi-strip	Entfernt Debug- und/oder Symbol-Informationen aus einem Binary
file	Liefert grundlegende Informationen zu einer Datei sowie deren Typ/Inhalt