Vorlesungsskript zu „Vertiefung Programmieren“ C-Runtime/Standard Bibliotheken

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

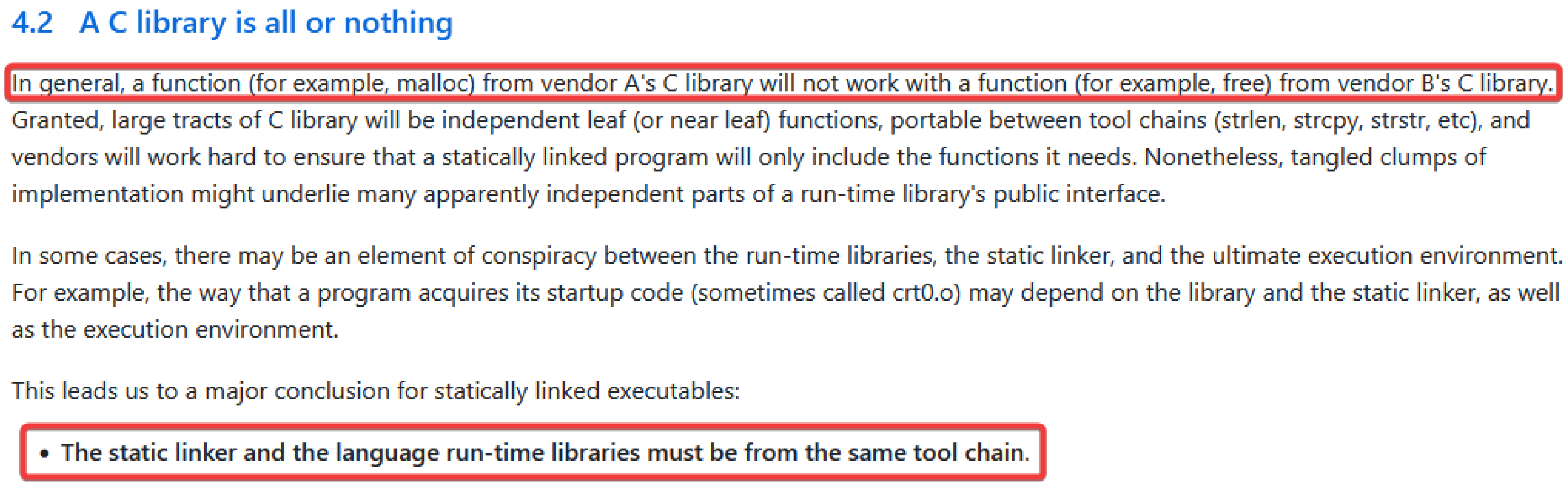
Bilbiotheken

C-Standard Bibliothek

C-Standard Bibliothek

C-Standard Bibliothek/C-Library

- Die C-Standard Bibliothek stellt einen Satz standardisierter C-Funktionen für die unterschiedlichsten Bereiche bereit.
- Z.B. String-Handling, Speicherverwaltung, Konsolen Ein-/Ausgabe etc.



4.2 A C library is all or nothing

In general, a function (for example, malloc) from vendor A's C library will not work with a function (for example, free) from vendor B's C library. Granted, large tracts of C library will be independent leaf (or near leaf) functions, portable between tool chains (strlen, strcpy, strstr, etc), and vendors will work hard to ensure that a statically linked program will only include the functions it needs. Nonetheless, tangled clumps of implementation might underlie many apparently independent parts of a run-time library's public interface.

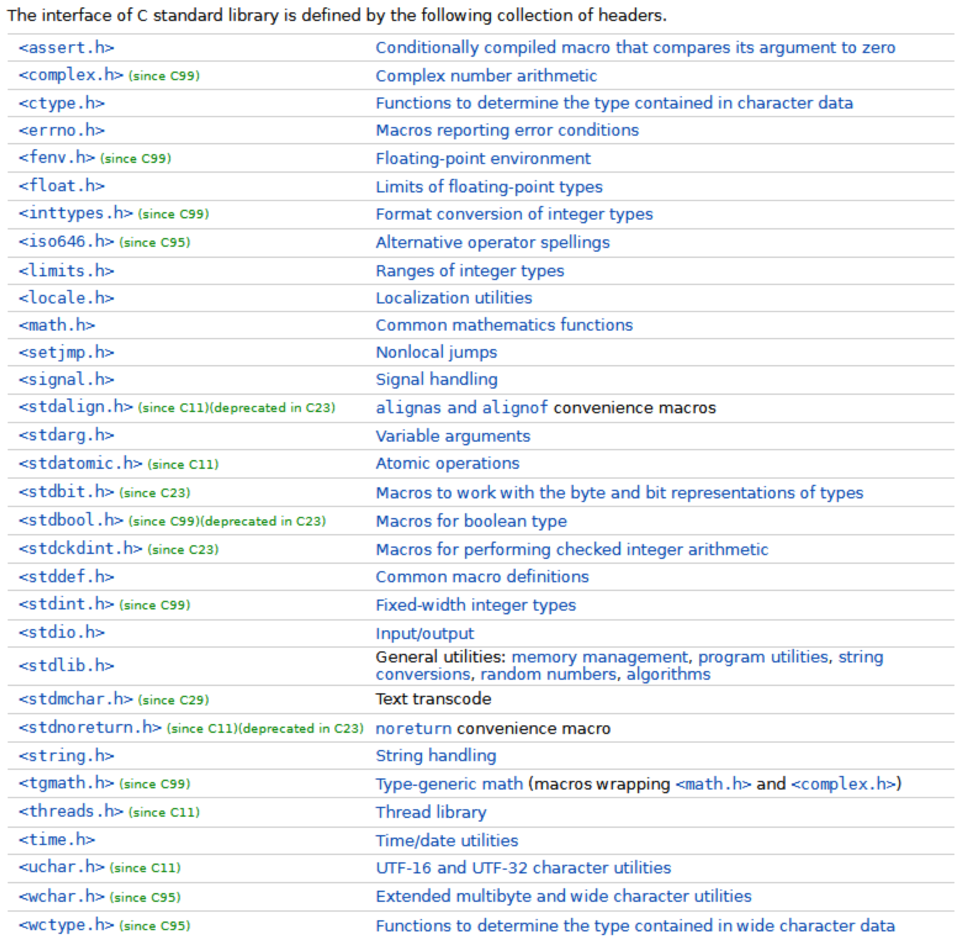
In some cases, there may be an element of conspiracy between the run-time libraries, the static linker, and the ultimate execution environment. For example, the way that a program acquires its startup code (sometimes called crt0.o) may depend on the library and the static linker, as well as the execution environment.

This leads us to a major conclusion for statically linked executables:

- **The static linker and the language run-time libraries must be from the same tool chain.**

Quelle: intern

C-Standard Library Headers



The interface of C standard library is defined by the following collection of headers.

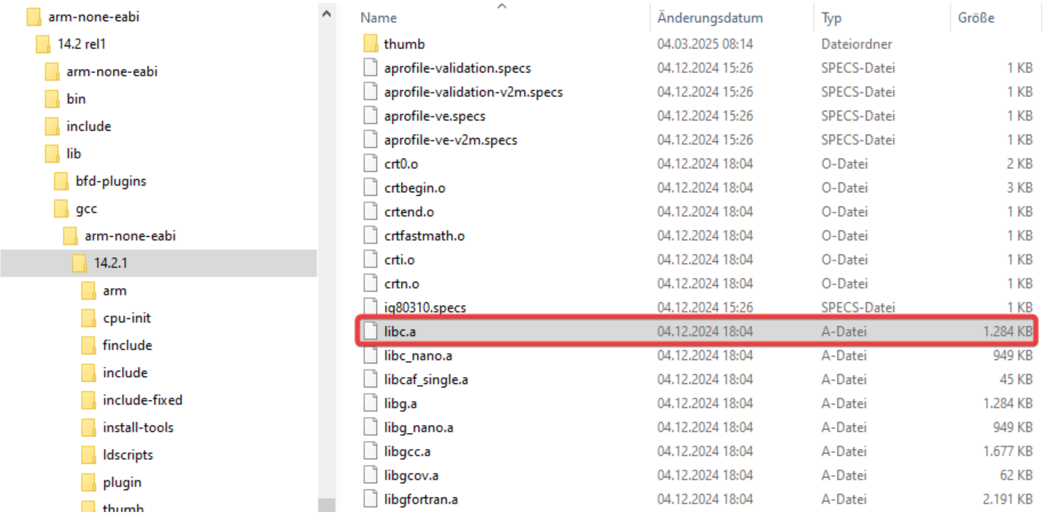
<assert.h>	Conditionally compiled macro that compares its argument to zero
<complex.h> (since C99)	Complex number arithmetic
<ctype.h>	Functions to determine the type contained in character data
<errno.h>	Macros reporting error conditions
<fenv.h> (since C99)	Floating-point environment
<float.h>	Limits of floating-point types
<inttypes.h> (since C99)	Format conversion of integer types
<iso646.h> (since C95)	Alternative operator spellings
<limits.h>	Ranges of integer types
<locale.h>	Localization utilities
<math.h>	Common mathematics functions
<setjmp.h>	Nonlocal jumps
<signal.h>	Signal handling
<stdalign.h> (since C11)(deprecated in C23)	alignas and alignof convenience macros
<stdarg.h>	Variable arguments
<stdatomic.h> (since C11)	Atomic operations
<stdbit.h> (since C23)	Macros to work with the byte and bit representations of types
<stdbool.h> (since C99)(deprecated in C23)	Macros for boolean type
<stdckdint.h> (since C23)	Macros for performing checked integer arithmetic
<stddef.h>	Common macro definitions
<stdint.h> (since C99)	Fixed-width integer types
<stdio.h>	Input/output
<stdlib.h>	General utilities: memory management, program utilities, string conversions, random numbers, algorithms
<stdmchar.h> (since C29)	Text transcode
<stdnoreturn.h> (since C11)(deprecated in C23)	noreturn convenience macro
<string.h>	String handling
<tgmath.h> (since C99)	Type-generic math (macros wrapping <math.h> and <complex.h>)
<threads.h> (since C11)	Thread library
<time.h>	Time/date utilities
<uchar.h> (since C11)	UTF-16 and UTF-32 character utilities
<wchar.h> (since C95)	Extended multibyte and wide character utilities
<wctype.h> (since C95)	Functions to determine the type contained in wide character data

Quelle: https://en.cppreference.com/w/c/header

- Um Funktionen aus der C-Standard Bibliothek zu verwenden, muss zum einen das entsprechende Header-File inkludiert werden und zum anderen die korrekte Bibliothek dem Linker mitgeteilt werden.
- Die C-Standard Bibliothek wird i.d.R. durch den Linker implizit eingebunden
- Das keine Standard-Bibliothek verwendet werden soll, muss häufig explizit als Kommandozeilenparameter dem Linker mitgeteilt werden.

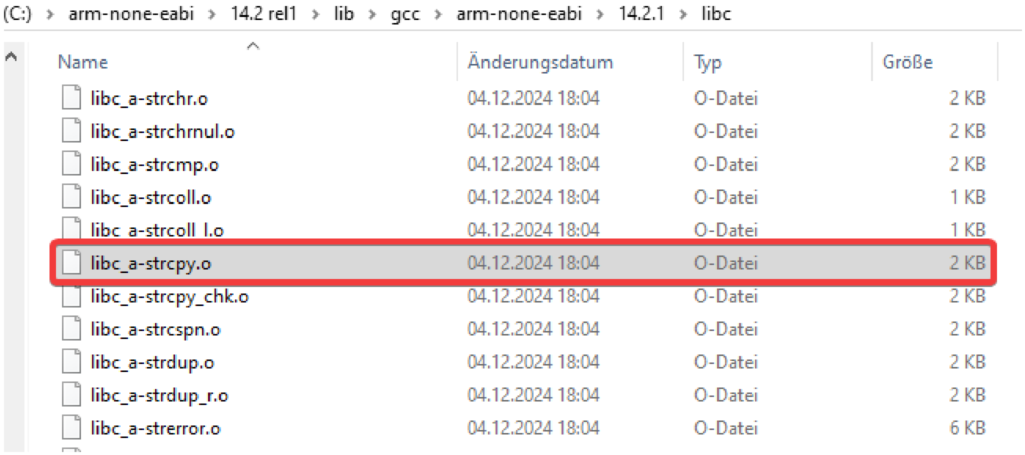
C-Standard Bibliothek

C-Standard Library



- arm-none-eabi
  - 14.2 rel1
    - arm-none-eabi
    - bin
    - include
    - lib
      - bfd-plugins
      - gcc
        - arm-none-eabi
          - 14.2.1
            - arm
            - cpu-init
            - finclude
            - include
            - include-fixed
            - install-tools
            - ldscripts
            - plugin
            - thumb

Name	Änderungsdatum	Typ	Größe
thumb	04.03.2025 08:14	Dateiordner	
aprofile-validation.specs	04.12.2024 15:26	SPECS-Datei	1 KB
aprofile-validation-v2m.specs	04.12.2024 15:26	SPECS-Datei	1 KB
aprofile-ve.specs	04.12.2024 15:26	SPECS-Datei	1 KB
aprofile-ve-v2m.specs	04.12.2024 15:26	SPECS-Datei	1 KB
crt0.o	04.12.2024 18:04	O-Datei	2 KB
crtbegin.o	04.12.2024 18:04	O-Datei	3 KB
crtend.o	04.12.2024 18:04	O-Datei	1 KB
crtfastmath.o	04.12.2024 18:04	O-Datei	1 KB
crti.o	04.12.2024 18:04	O-Datei	1 KB
crtn.o	04.12.2024 18:04	O-Datei	1 KB
iq80310.specs	04.12.2024 15:26	SPECS-Datei	1 KB
libc.a	04.12.2024 18:04	A-Datei	1.284 KB
libc_nano.a	04.12.2024 18:04	A-Datei	949 KB
libcaf_single.a	04.12.2024 18:04	A-Datei	45 KB
libg.a	04.12.2024 18:04	A-Datei	1.284 KB
libg_nano.a	04.12.2024 18:04	A-Datei	949 KB
libgcc.a	04.12.2024 18:04	A-Datei	1.677 KB
libgcov.a	04.12.2024 18:04	A-Datei	62 KB
libgfortran.a	04.12.2024 18:04	A-Datei	2.191 KB



(C:) › arm-none-eabi › 14.2 rel1 › lib › gcc › arm-none-eabi › 14.2.1 › libc

Name	Änderungsdatum	Typ	Größe
libc_a-strchr.o	04.12.2024 18:04	O-Datei	2 KB
libc_a-strchrnul.o	04.12.2024 18:04	O-Datei	2 KB
libc_a-strcmp.o	04.12.2024 18:04	O-Datei	2 KB
libc_a-strcoll.o	04.12.2024 18:04	O-Datei	1 KB
libc_a-strcoll_l.o	04.12.2024 18:04	O-Datei	1 KB
libc_a-strcpy.o	04.12.2024 18:04	O-Datei	2 KB
libc_a-strcpy_chk.o	04.12.2024 18:04	O-Datei	2 KB
libc_a-strcspn.o	04.12.2024 18:04	O-Datei	2 KB
libc_a-strdup.o	04.12.2024 18:04	O-Datei	2 KB
libc_a-strdup_r.o	04.12.2024 18:04	O-Datei	2 KB
libc_a-strerror.o	04.12.2024 18:04	O-Datei	6 KB

Quelle: intern

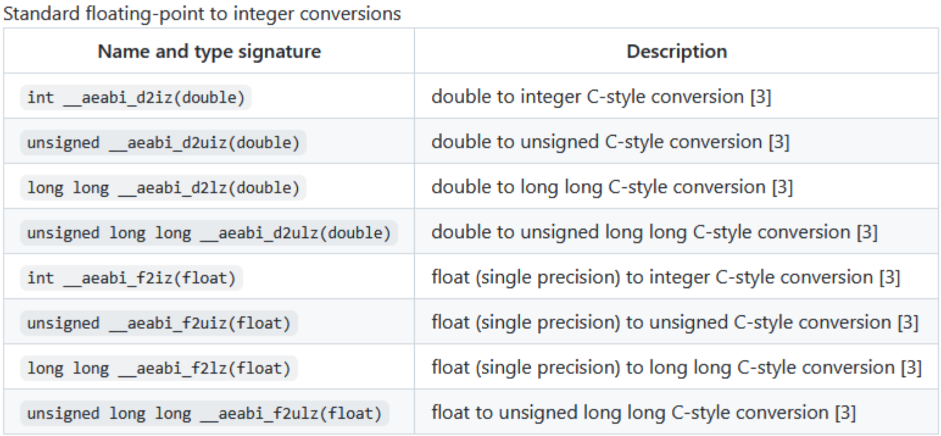
Bilbiotheken

C-Runtime Bibliothek

C-Standard Bibliothek

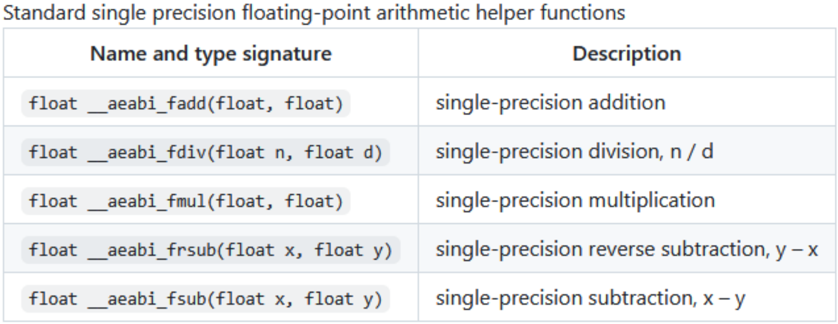
C-Runtime Bibliothek

- Die C-Runtime Bibliothek enthält, im Gegensatz zur C-Standard Bibliothek, keine durch den Benutzer explizit aufrufbaren Funktionen.
- Diese Bibliothek stellt standardisierte Funktionen bereit, welche der Compiler in den generierten Assembler-Code integriert, um bestimmte Funktionalitäten auf allen unterstützten Plattformen bereitzustellen.
- Wichtigster Teil der C-Runtime sind die Floating-Point Funktionen



Standard floating-point to integer conversions

Name and type signature	Description
`int __aeabi_d2iz(double)`	double to integer C-style conversion [3]
`unsigned __aeabi_d2uiz(double)`	double to unsigned C-style conversion [3]
`long long __aeabi_d2lz(double)`	double to long long C-style conversion [3]
`unsigned long long __aeabi_d2ulz(double)`	double to unsigned long long C-style conversion [3]
`int __aeabi_f2iz(float)`	float (single precision) to integer C-style conversion [3]
`unsigned __aeabi_f2uiz(float)`	float (single precision) to unsigned C-style conversion [3]
`long long __aeabi_f2lz(float)`	float (single precision) to long long C-style conversion [3]
`unsigned long long __aeabi_f2ulz(float)`	float to unsigned long long C-style conversion [3]



Standard single precision floating-point arithmetic helper functions

Name and type signature	Description
`float __aeabi_fadd(float, float)`	single-precision addition
`float __aeabi_fdiv(float n, float d)`	single-precision division, n / d
`float __aeabi_fmul(float, float)`	single-precision multiplication
`float __aeabi_frsub(float x, float y)`	single-precision reverse subtraction, y – x
`float __aeabi_fsub(float x, float y)`	single-precision subtraction, x – y

Quelle: intern

C-Standard Bibliothek

C-Runtime Bibliothek

- Neben den Floating-Pointer Helper Functions, stellt die C-Runtime Bibliothek auch noch Funktionen für folgende Bereiche bereit
  - Integer Division (32/32 →32)
  - Division by Zero
  - Unalinged Memory Access
  - Memory copying, clearing and setting
  - C++ Support (e.g. Constructor/Destructor Calls)

Quelle: intern