

Systemnahe Programmierung SS 2024

Die Programmierumgebung

Helmut Lindner

Struktur eines C-Programms

Erstes C-Programm

```
#include <stdio.h>
```

-> Headerdatei `stdio.h` einbinden
Deklarationen externer Funktionen und Werte
für Standard I/O (Konsole, Dateien..)

```
/*  
    Mehrzeiliger Kommentar
```

```
*/  
int main(int argc, char *argv[]) {  
    int anzahl = 99;
```

-> Definition der Funktion *main*

-> Definition der integer Variable *anzahl*

```
    // Einzeiliger Kommentar
```

```
    printf("%d Luftballons.\n", anzahl);
```

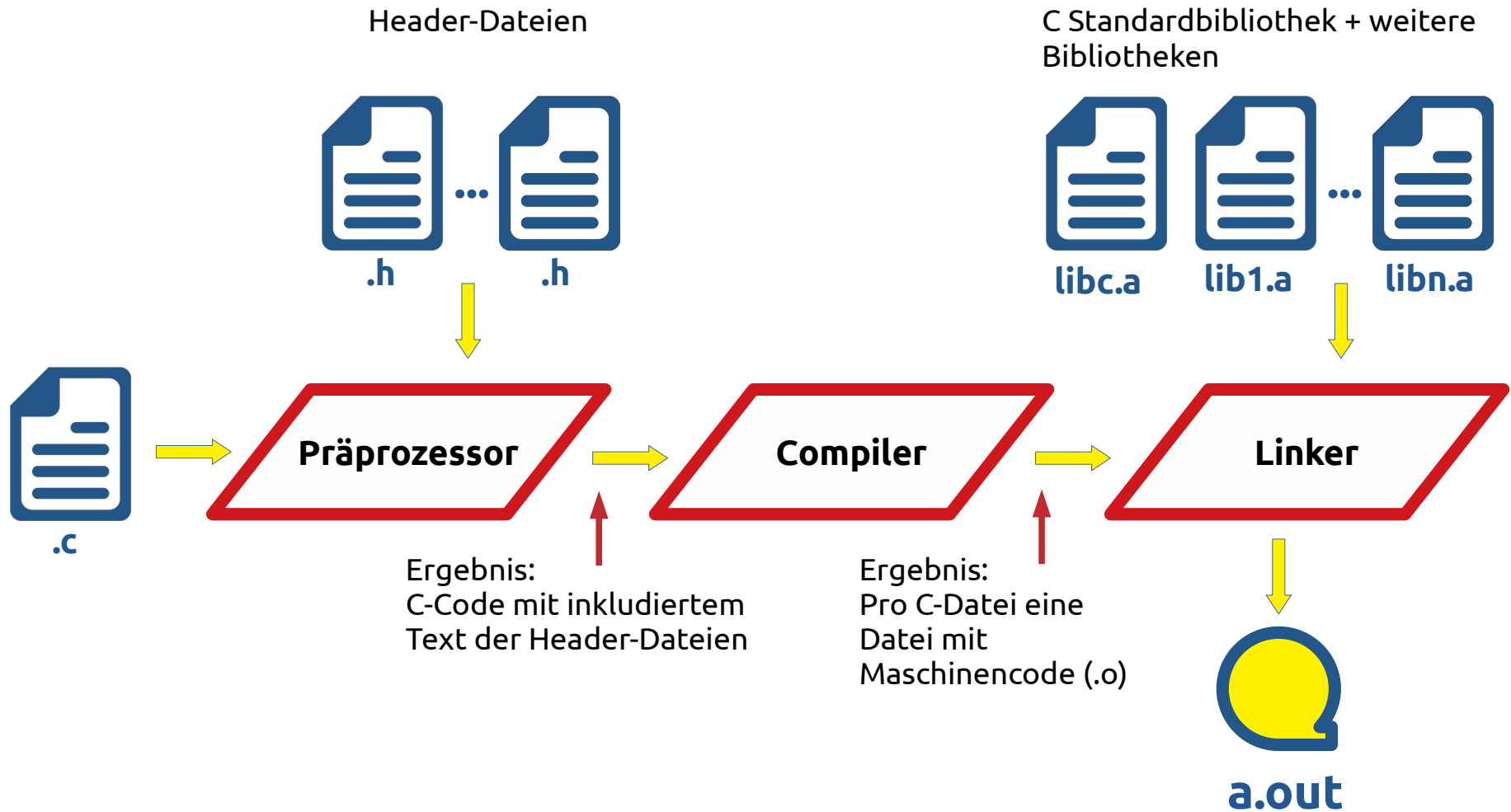
-> Aufruf der Funktion *printf*

```
    return 0;
```

-> Wert 0 zurückgeben

```
}
```

Compilieren eines C-Programms



Compilieren eines C-Programms

Der C-Compiler

Der C-Compiler ist standardmäßig als `cc` Befehl verfügbar:

```
cc ex1.c
```

Der Compiler ruft auch den Linker auf und erzeugt so gleich eine ausführbare Datei mit dem Default-Namen `a.out`.

Mit dem Flag `-o` wird der Name der ausführbaren Datei angegeben:

```
cc ex1.c -o luftballon
```

Nützliche Compiler-Flags

- Wall ...alle Warnings anzeigen
- O0 ... alle Optimierungen deaktivieren
- g ... Debuginformationen erzeugen
- l ... zusätzliche Bibliothek einbinden

Default Einstellungen des Compilers anzeigen:

```
cc -v -x c -E /dev/null
```

Struktur eines C-Programms

Header Dateien

`stdio.h` ist ein Teil der C Standard Bibliothek

weitere wichtige Headerdateien:

`ctype.h`, `math.h`, `stdlib.h`, `string.h`, `time.h`

Die Header-Dateien müssen sich im Include-Pfad befinden. Dieser zeigt normalerweise auf `/usr/include` und Unterverzeichnisse.

Mit dem Compiler-Flag `-I<Pfad>` können weitere Include-Verzeichnisse angegeben werden.

Dokumentation der GNU C Library:

https://www.gnu.org/software/libc/manual/html_node/index.html

Übersichtliche Funktionsreferenz:

https://www.tutorialspoint.com/c_standard_library/index.htm

Allgemeines

Programmierungsumgebung und Tools

Ein virtuelle Maschine (Oracle VirtualBox) mit allen benötigten Tools steht zum Download unter

https://fhjoanneum-my.sharepoint.com/:u:/g/personal/helmut_lindner_fh-joanneum_at/EcuYYEVI5otHg8SnHrPucsUBniR6fkJMYi6B2Q50UurhGw?e=NeoLfh

bereit.

Ich empfehle diese VM zu verwenden, grundsätzlich können sie auch ihr eigenes Linux verwenden.

Wir verwenden folgende Tools:

- gcc - GNU C-Compiler
- make
- Texteditor oder IDE (nach Vorliebe)

Installation

Tools installieren (ubuntu Linux)

Compiler und Tools installieren

```
sudo apt install build-essential
```

Splint – statische Checks

```
sudo apt install splint
```

Valgrind – Laufzeit Prüfung

```
sudo apt install valgrind
```

man Pages

Manpages installieren

```
sudo apt-get install manpages-dev
```

Wichtige Abschnitte:

SYNOPSIS

Beschreibt die Funktionssignatur und notwendige Include-Dateien.

Beispiel:

```
#include <unistd.h>
ssize_t write(int fd, const void *buf, size_t count);
```

DESCRIPTION

RETURN VALUE – wichtig, was wird zurückgegeben (Datentyp!)

ERRORS – Wie wird errno im Fehlerfall gesetzt.

SEE ALSO – Referenz zu ähnlichen Funktionen

Übung

Programm compilieren

1. Erstellen sie das erste C-Programm im Moodle VPL und compilieren Sie es.

Verändern sie das Programm so, dass es anstelle der Luftballons die Quadratwurzel der Zahl 3638 berechnet und ausgibt.

Verwenden sie dazu die Bibliotheksfunktion `sqrt()`.

Verwenden sie

`man sqrt`

um die Definition der `sqrt()` Funktion zu sehen.

Verwenden sie als Ausgabeformat (`printf`): `%.2lf`

Ausgabe: Die Wurzel von 3638 ist <Ergebnis>

2. Compilieren sie das Programm lokal auf ihrem Rechner und führen sie es aus.

