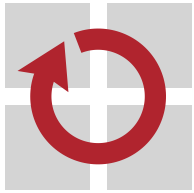
Übungen zu Systemnahe Programmierung in C

Abschnitt 10.5: Die Funktion main()

29.06.2020

Tim Rheinfels
Benedict Herzog
Bernhard Heinloth

Lehrstuhl für Informatik 4
Friedrich-Alexander-Universität Erlangen-Nürnberg



Lehrstuhl für Verteilte Systeme
und Betriebssysteme

FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG

TECHNISCHE FAKULTÄT

Die Funktion main()

- Funktion `main( )`: Einsprungstelle für ein C Programm
- Signatur nach Anwendungszweck:
  - AVR: Nur ein Programm
    - ⇒ `void main(void)`
  - Linux: Mehrere Programme
    - ⇒ `int main(void)`
    - ⇒ `int main(int argc, char *argv[])`
- Parameter und Rückgabewert zur Kommunikation

Kommandozeilenparameter

- Kommandozeilenparameter: Argumente für Programme
- `main()` erhält sie als Funktionsparameter:
  - `argc`: Anzahl der Argumente
  - `argv`: Array aus Zeigern auf Argumente

  $\Rightarrow$ Array von Strings
- Erstes Argument: Programmname

Kommandozeilenparameter – Beispiel

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

int main(int argc, char *argv[]) {
  for(int i = 0; i < argc; ++i) {
    printf("argv[%d]: %s\n", i, argv[i]);
  }

  return EXIT_SUCCESS;
}
```

```
$ ./commandline
argv[0]: ./commandline
$ ./commandline Hallo Welt
argv[0]: ./commandline
argv[1]: Hallo
argv[2]: Welt
```

Rückgabestatus

- Rückgabestatus: Information für den Aufrufenden
- Übliche Codes:
  - `EXIT_SUCCESS`: Ausführung erfolgreich
  - `EXIT_FAILURE`: Fehler aufgetreten

Rückgabestatus – Beispiel

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

int main(int argc, char *argv[]) {
  if(argc == 1) {
    fprintf(stderr, "No parameters given!\n");
    return EXIT_FAILURE;
  }

  // [...]

  return EXIT_SUCCESS;
}
```

```
$ ./exitcode
No parameters given!
$ echo $?
1
$ ./exitcode Hallo Welt
$ echo $?
0
```