

# C-Kurs

## “hello, world”

# Die Programmiersprache C

- ❑ Entwickelt zwischen 1969 und 1973 von Dennis Ritchie bei Bell Laboratories
- ❑ ANSI C Standard 1989 ratifiziert durch American National Standards Institute
- ❑ C99 kleinere Erweiterungen zu ANSI C
- ❑ Zusammenfassung von „K&R“:
  - C ist „quirky, flawed, and an enormous success“
  - C und Unix sind eng miteinander verbunden
  - C ist eine kleine und einfache Sprache
  - C ist für praktische Zwecke entworfen worden
  - C ist die Sprache für systemnahe Programmierung

## □ Bestandteile:

- Hardware (Prozessor, Arbeitsspeicher, etc.)
- Software (Betriebssystem, Textverarbeitungsprogramme, etc.)

## □ Aufgabe:

- Anwendungsprogramme ausführen

## □ Vorlesungsziel:

- Anwendungsprogramme programmieren zu können

- „Hello World“ Programm aus  
„C programming language“ Kernighan, Ritchie

```
1 #include <stdio.h>
2
3 int main() {
4     printf ("hello, world\n");
5 }
```

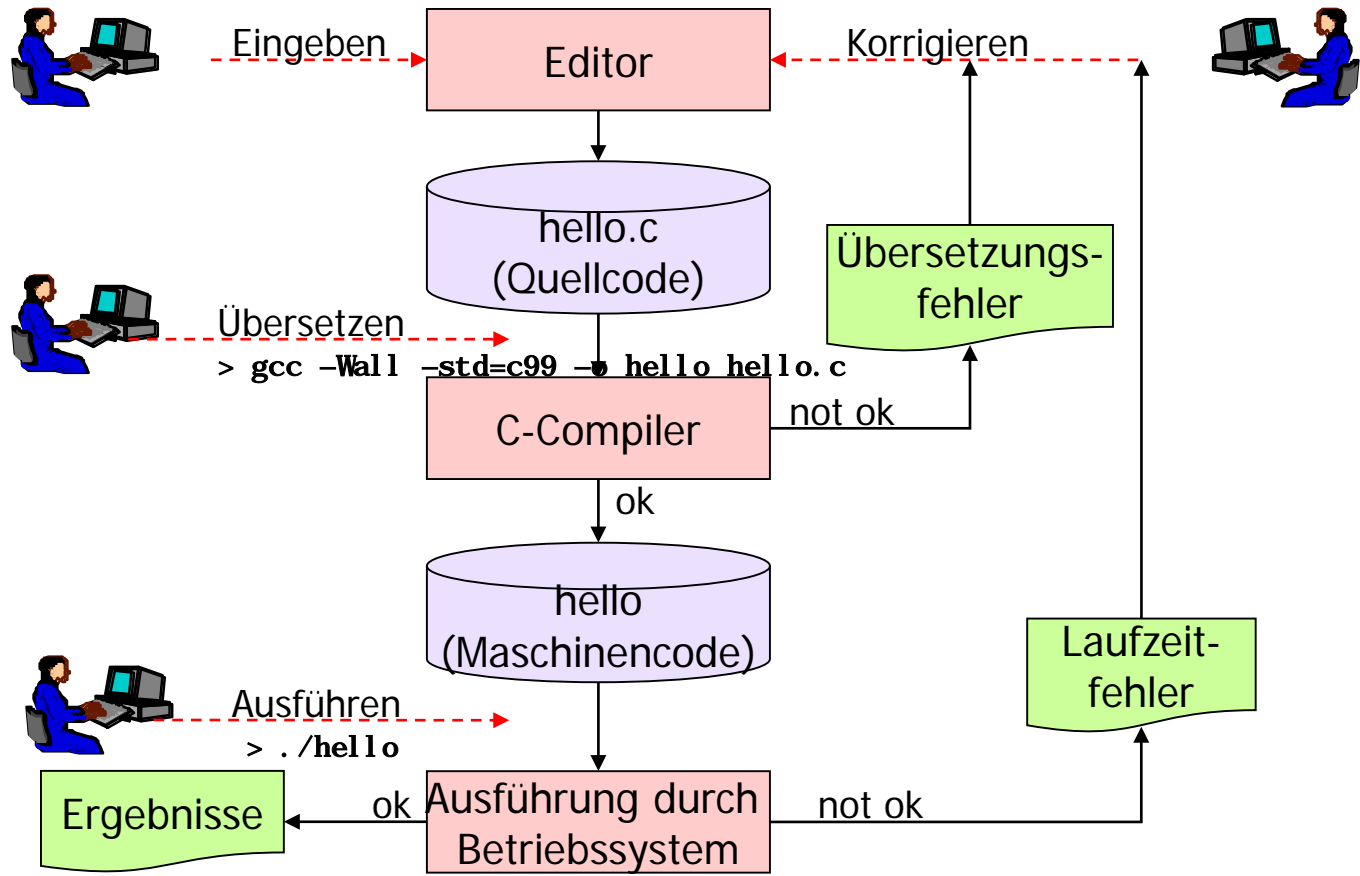
# Quelltext zu Programm

- „Hello World“ Programm aus  
„C programming language“ Kernighan, Ritchie

```
1 #include <stdio.h>
2
3 int main() {
4     printf ("hello, world\n");
5 }
```

- Frage: Wie kommt man vom Quelltext  
zum ausführbaren Programm?

# Ablauf des Programmierens



## ☐ Quelltext

```
1 #include <stdio.h>
2
3 int main() {
4     printf ("hello, world\n");
5 }
```

☐ Editor: Programm mit dem man Quelltext, z.B. C-Code, editieren kann

➤ Kate, gedit, nano, emacs, vim

☐ Quelltext ist ASCII Text **ohne** Formatierung, d.h. Word, Openoffice, etc. sind **keine** Editoren für Quelltext

# Quelltext – Quelldatei

## ❑ Quelltext

```
1 #include <stdio.h>
2
3 int main() {
4     printf ("hello, world\n");
5 }
```

## ❑ Quelldatei (source file): **hello.c**

- Sequenz von Bits (0,1)  
eingeteilt in 8-Bit Einheiten, Bytes
- Text Zeichen repräsentiert nach dem ASCII Standard  
Jedes Zeichen, eindeutiger Zahlenwert
- Textdateien: Nur ASCII Zeichen
- Binärdateien: Alle anderen Dateien, die keine Textdateien sind



# Ziel: Ausführbares Programm

## □ Fundamentale Idee:

- Alle Information im System: Folge von Bits
- Typunterscheidung durch Kontext

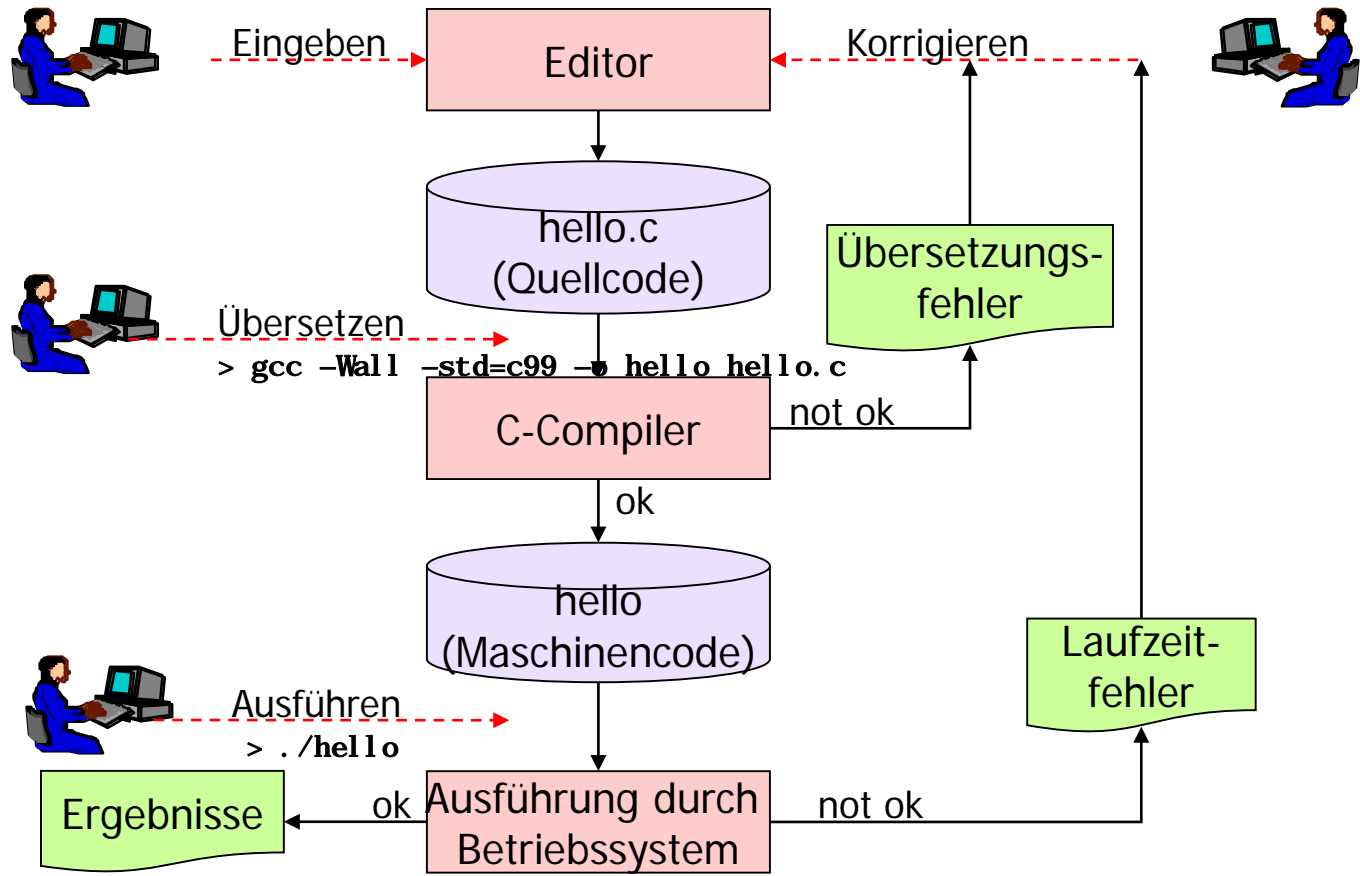
## □ Quellprogramm:

Zeichenkette (unformatierter Text)

## □ Ausführbares Programm:

Instruktionen für den Prozessor

# Ablauf des Programmierens



# Compilieren – Übersetzen

## □ Hello Programm:

### ➤ Anfang:

C Quelltext auch oft als C Programm bezeichnet

C Programm – verständlich für Menschen

### ➤ Ziel:

Maschinensprache – verständlich für Computer

- Maschinensprache besteht aus Maschinenbefehlen, den Instruktionen
- Gespeichert in ausführbaren Programmdateien

### ➤ Mittel:

Compilersystem

□ Ein Compiler hat die Aufgabe Quellcode in ausführbaren Maschinencode zu übersetzen. Ein Compiler ist selbst wieder ein Programm.

# Anwendung: C-Compiler

- ❑ Beispiel: GNU Compiler Collection (gcc)

```
unix> gcc -Wall -std=c99 -o hello hello.c
```

- ❑ Benennen der Ausgabedatei: Compilerflag `-o <name>`  
(Ansonsten wird die Datei `a.out` generiert)
- ❑ Wir benutzen den Programmierstandard C99  
(Compilerflag: `-std=c99`)
- ❑ Wir empfehlen mit Warnungen zu kompilieren  
(Compilerflag: `-Wall`)

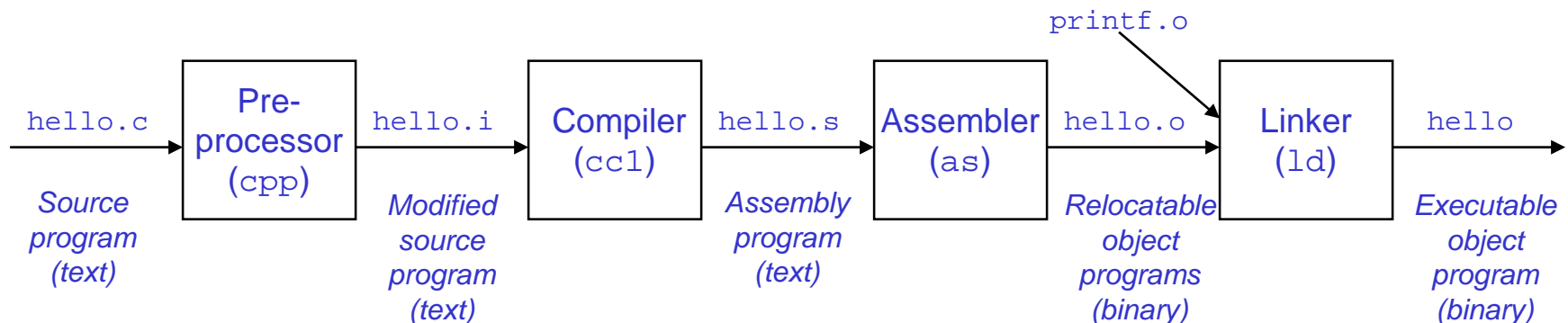
# Überblick: C-Compiler

## □ Beispiel: GNU Compiler Collection (gcc)

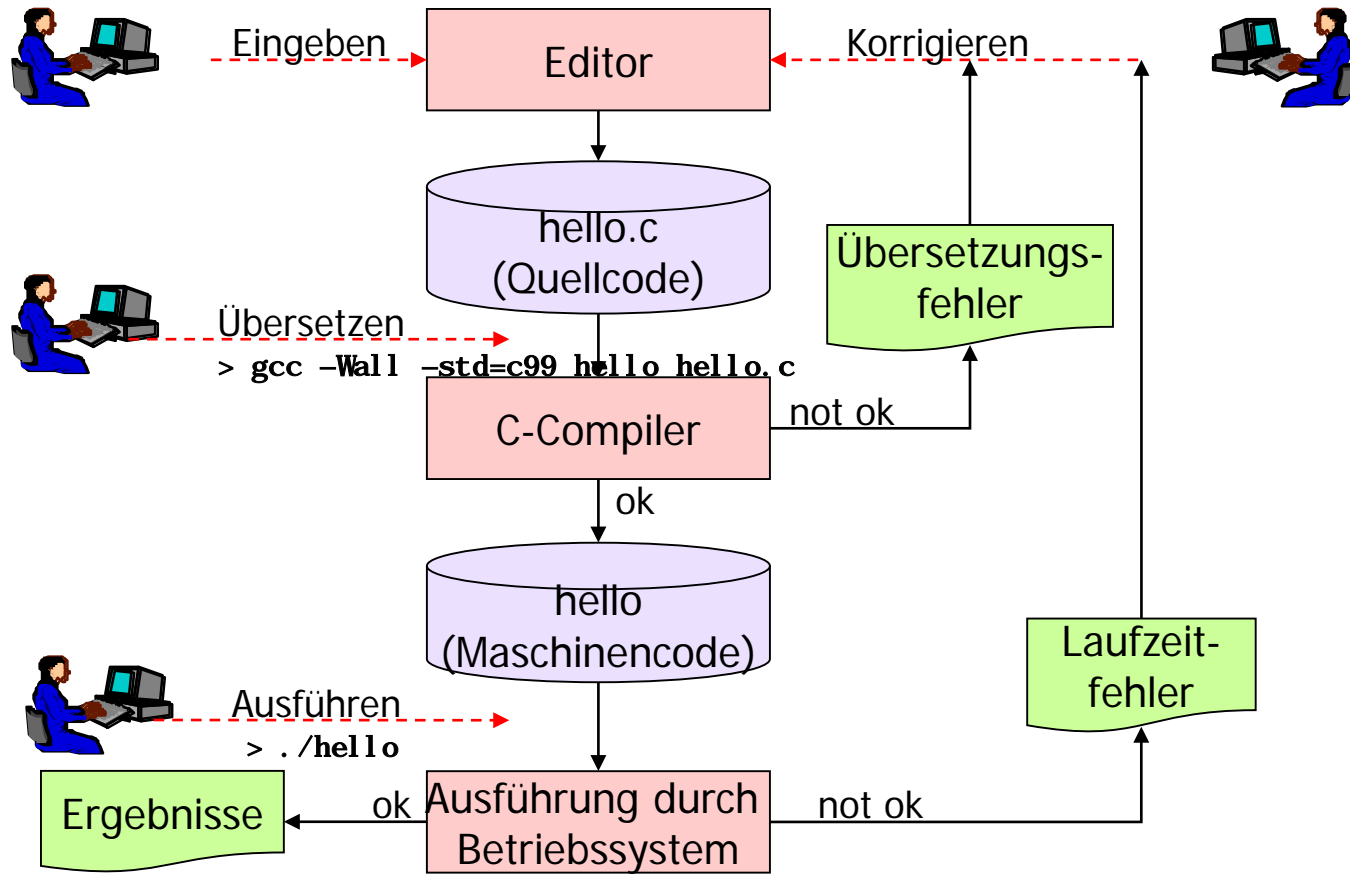
```
unix> gcc -Wall -std=c99 -o hello hello.c
```

### ➤ 4 Phasen:

- Preprocessor      Aufbereitung
- Compiler      Übersetzt C in Assemblercode
- Assembler      Übersetzt Assemblercode in Maschinsprache
- Linker      Nachbearbeitung / Kombination  
                 verschiedener Module



# Ablauf des Programmierens



# Ausführung eines Programms

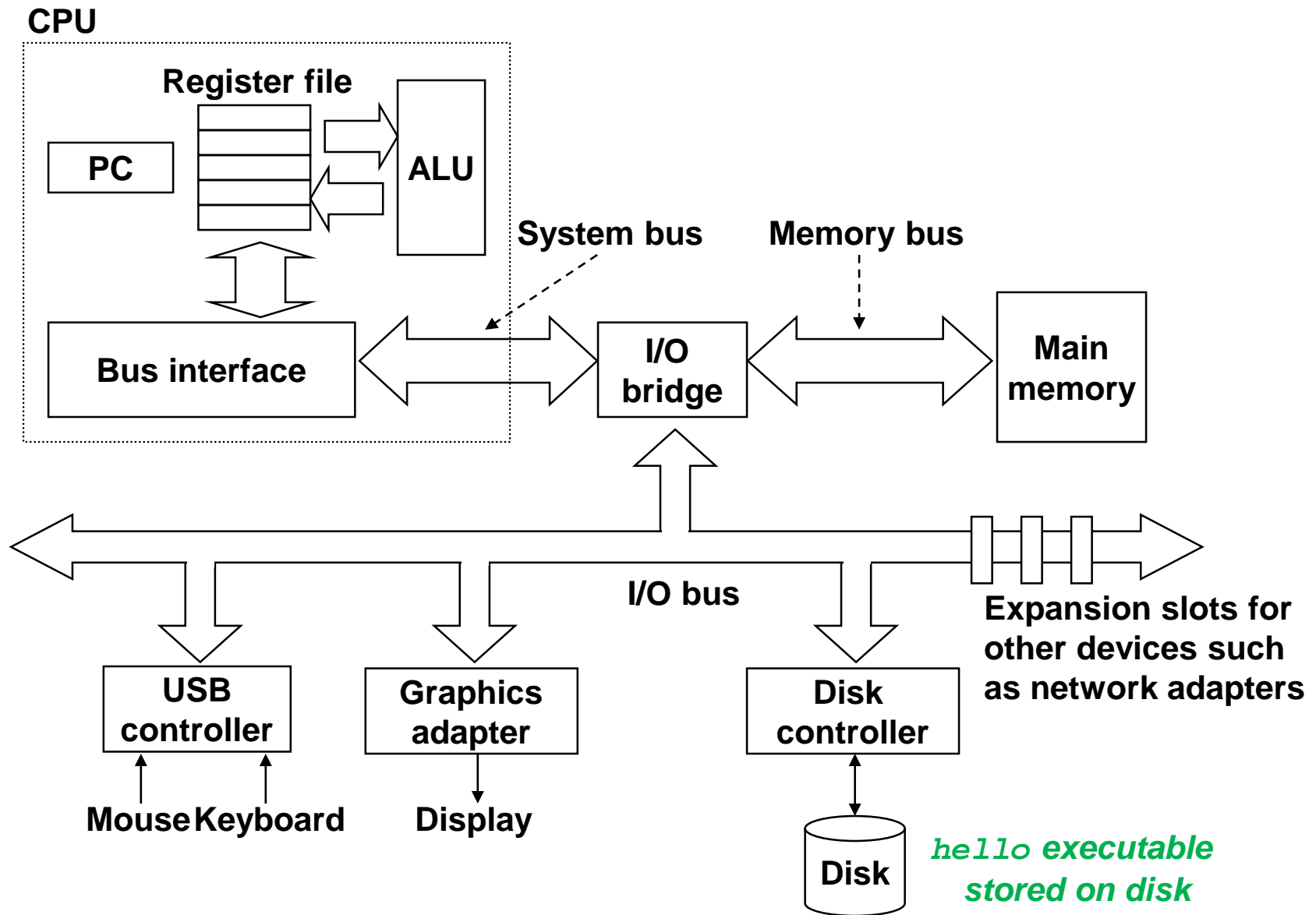
□ Programme werden von anderen Programmen gestartet

□ Hello Programm:

- Anfang: Ausführbares Programm
- Ziel: Ergebnis des Programms
- Mittel: Spezielle Anwendung: **shell (terminal)**
  - Kommandozeileninterpreter
    - Druckt Eingabeaufforderung (Prompt)
    - Wartet auf Eingabe einer Kommandozeile
    - Führt Kommando aus  
(Annahme: 1. Wort: Shell Kommando oder ausführbares Programm)

```
unix> ./hello
hello, world
unix>
```

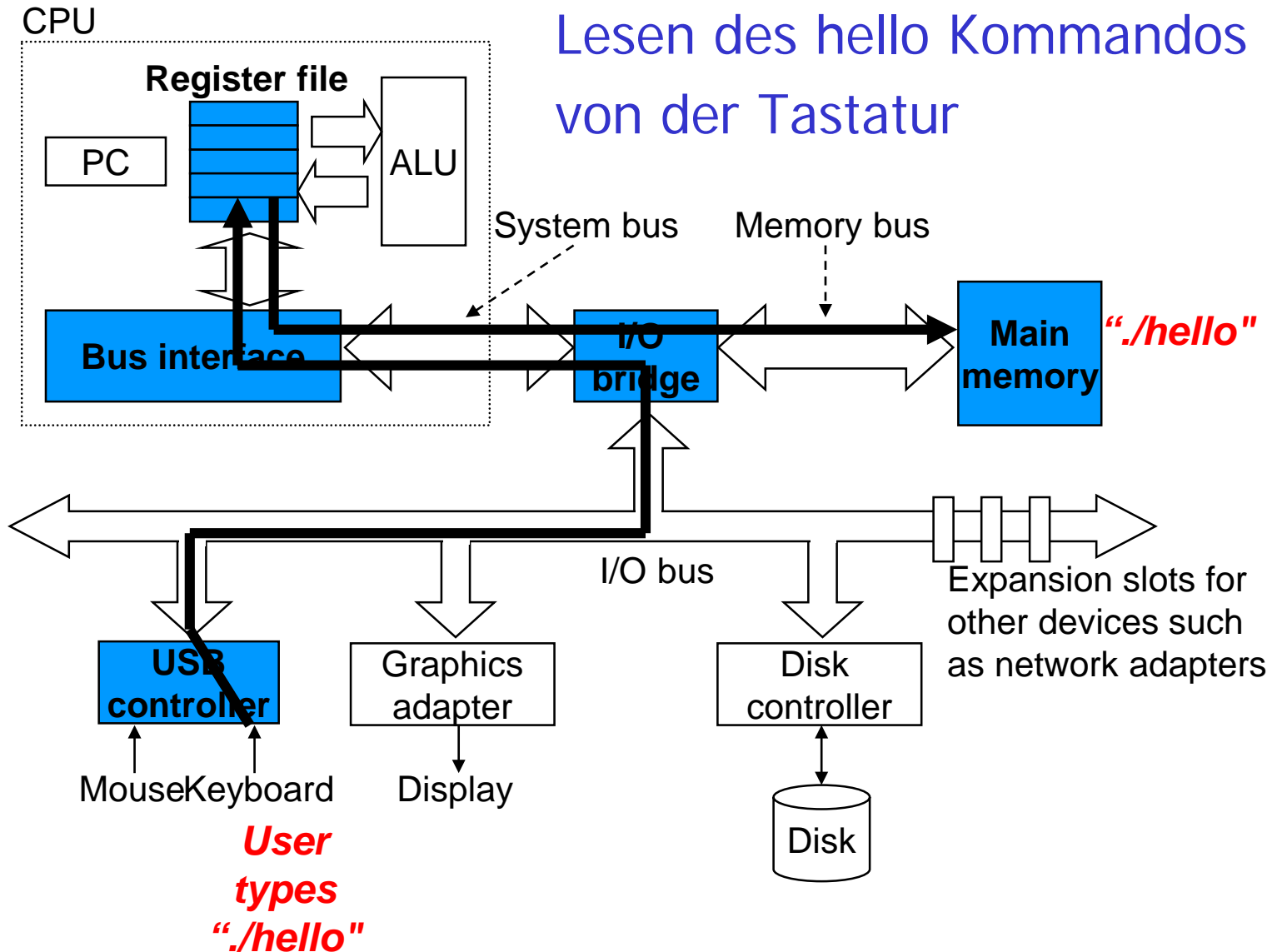
# Ausflug: Die Rechnerhardware





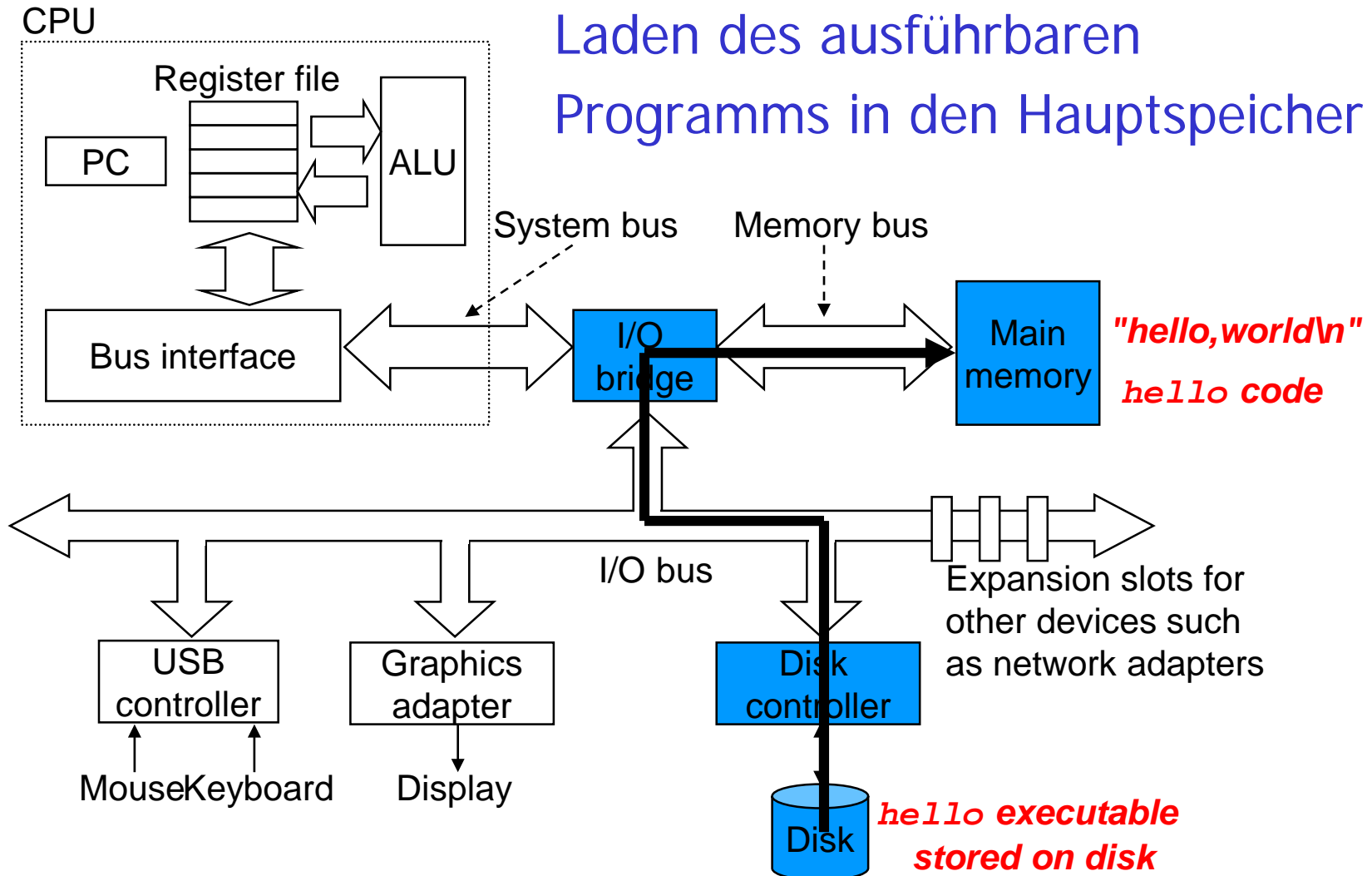
# Ablauf: Ausführung

Lesen des hello Kommandos  
von der Tastatur



# Ablauf: Ausführung

Laden des ausführbaren  
Programms in den Hauptspeicher



# Ablauf: Ausführung

Schreiben der Ausgabe vom  
Speicher auf das Display

