

Echtzeitsysteme (SoSe 2020)

Dr.-Ing. ReneTschirley

Beuth Hochschule für Technik Berlin

April.2020

Kapitel 1: Einleitung

Organisatorisches

- ▶ Stellen Sie Fragen!
 - ▶ Unterricht,
 - ▶ E-Mail <mailto:rene.tschirley@beuth-hochschule.de>

- ▶ Vorlesung
- ▶ Übungen
- ▶ Klausur
- ▶ Prüfungen

- ▶ Dozenten sind fehlbar

⇒ Bei Unklarheiten stellen Sie Fragen.

Übung

- ▶ Es gibt zu den behandelten Kapitel Aufgabenblätter.

Compiler, IDE, Makefiles

- ▶ Erfolgreichbestanden: Mindestens 2/3 erfolgreiche Abgaben

Sonstiges

- ▶ Drucken Sie die Folien vor der Veranstaltung aus
- ▶ Machen Sie sich Notizen
- ▶ Ohne Mitarbeit und Bearbeitung der Übungen werden Sie nicht viel lernen
- ▶ Bereiten Sie die Vorlesung und Übungen nach
- ▶ Bitte keine Smartphones in der Vorlesung benutzen
- ▶ Stellen Sie Fragen

Acknowledgement

Der Inhalt der Vorlesung wurde durch die folgenden Werke inspiriert:

- ▶ Vorlesungsunterlagen von Prof. Dr. Alois Knoll, Prof. Dr. Wolfgang Schröder-Preikschat, und Prof. Dr. Christian Baun
- ▶ Bücher
 - ▶ Echtzeitsysteme (ISBN 1613-5216)
– Heinz Wörn und Uwe Brinkschulte
 - ▶ Moderne Betriebssysteme (ISBN 3827-370191)
– Andrew S. Tanenbaum
 - ▶ Linux Unix Systemprogrammierung (ISBN 3-8273-1412-3)
– Helmut Herold
 - ▶ The Linux Programming Interface (ISBN 9 781593 272203)
– Michael Kerrisk

Disclaimer

- ▶ Aus Gründen der Übersichtlichkeit wird auf den **Folien** größtenteils auf **Fehlerbehandlung verzichtet**.

```
char *buf = malloc(9001);
```

- ▶ Die Lösungen der Übungsaufgaben muss die Fehlerbehandlung beinhalten.

```
void error(char *msg, char *file, int line) {  
    fprintf(stderr, "%d:%s: %s: %s\n", line, file,  
        strerror(errno), msg);  
    exit(EXIT_FAILURE);  
}  
// ...  
char *buf;  
if ( !(buf = malloc(9001)) )  
    error(NULL, __FILE__, __LINE__);
```

Agenda

Diese Vorlesung beschäftigt sich mit den folgenden Grundlagenthemen zu **weichen Multitasking-Echtzeitsystemen**

1. Grundlagen (Echtzeitsysteme)
2. Prozesse
3. Fallbeispiel: UNIX-Prozesse
4. Signale
5. Pipes
6. Threads (Grundlagen)
7. Fallbeispiel: POSIX Threads
8. Schedulingstrategien

1.1: Echtzeitsysteme

Frage: Was ist ein Echtzeitsystem (EZS)?

Echtzeitsysteme vs. nicht Echtzeitsysteme

Korrektheit von nicht Echtzeitsystemen

Logische Korrektheit \implies Korrektheit

Ein System ist logisch korrekt falls es die richtigen Ergebnisse liefert.

Korrektheit von Echtzeitsystemen

Logische Korrektheit + zeitliche Korrektheit \implies Korrektheit

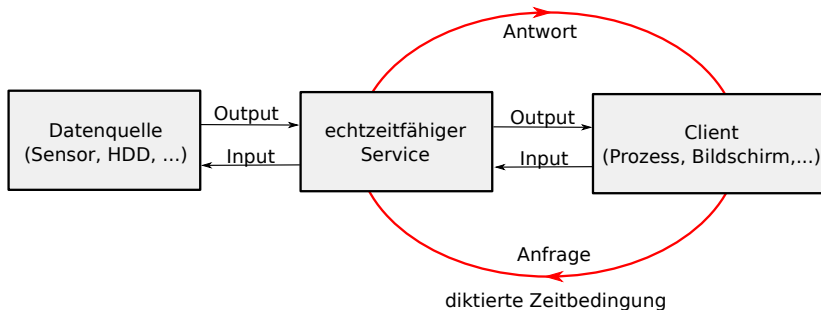
Ein System ist zeitlich korrekt falls es die Ergebnisse innerhalb einer vorgegebenen Zeitspanne liefert.

Frage: Kennen Sie Beispiele für (Nicht-)Echtzeitsysteme?

Definition nach Hermann Kopetz, TU Wien

- ▶ Ein Echtzeit-Computer-System ist ein Computersystem, in dem die **Korrektheit des Systems** vom **logischen Ergebnis** der Berechnung sowie dem **Zeitpunkt** in dem das Ergebnis produziert wird abhängt.
- ▶ Ein Echtzeit-Computer-System ist immer ein Teil eines größeren Systems, welches Echtzeitsystem genannt wird.

Skizze: Echtzeit-Datenverarbeitung

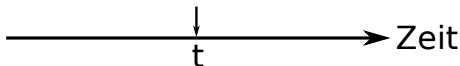


Probleme der Echtzeitfähigkeit

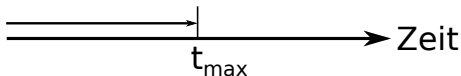
1. Geschwindigkeit liefert keine Garantie auf eine rechtzeitige Antwort.
 - ▶ Asynchrone Programmunterbrechungen (engl. *interrupts*) können **unvorhersagbare Laufzeitvarianzen** verursachen.
 - ▶ Schnelle Programmausführung ist notwendig für die rechtzeitige Bearbeitung einer Aufgabe.
2. Zeit ist keine intrinsische Eigenschaft des Rechensystems, daher muss die Zeitskala des EZS hinreichend hochauflösend und genau sein.

Arten von Rechtzeitigkeit

- ▶ Angabe eines **genauen Zeitpunktes**



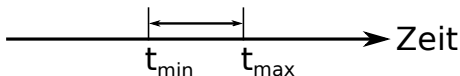
- ▶ Angabe eines **spätesten Zeitpunktes**



- ▶ Angabe eines **frühesten Zeitpunktes**



- ▶ Angabe eines **Zeitintervalls**



Zeitbedingungen

- ▶ **Periodische Zeitbedingungen**

Die Zeitbedingung wiederholt sich in **regelmäßigen** Abständen

- ▶ **Aperiodische Zeitbedingungen**

Die Zeitbedingung wiederholt sich in **unregelmäßigen** Abständen

- ▶ **Absolute Zeitbedingungen**

Hier wird die absolute Uhrzeit als Zeitbedingung definiert.

- ▶ **Relative Zeitbedingungen**

Hier wird die Zeitbedingung relativ zu einem anderen Zeitbedingungen oder Ereignis definiert.

Echtzeitbedingungen

- ▶ Weich (engl. *soft*)
 - ▶ Verspätete Antwort kann eintreten
 - ▶ Terminverletzung ist tolerierbar
- ▶ Fest (engl. *firm*)
 - ▶ Das Ergebnis einer zu einem vorgegebenen Termin nicht geleisteten Arbeit ist wertlos und wird verworfen
 - ▶ Terminverletzung ist noch tolerierbar
- ▶ Hart (engl. *hard*)
 - ▶ Das Versäumnis eines fest vorgegebenen Termins kann eine „Katastrophe“ hervorrufen
 - ▶ Terminverletzung ist keinesfalls tolerierbar

Ausführungsmodelle

▶ Zeitgesteuerte Systeme

- ▶ Gesamter zeitlicher Systemablauf wird zur Compile-Zeit festgelegt
- ▶ Präzise globale Uhr: Uhrensynchronisation notwendig
- ▶ Für die einzelnen Berechnungen ist jeweils ein Zeitslot reserviert: maximale Laufzeiten (**worst case execution times – WCET**) werden abgeschätzt
- ▶ **Vorteil:** Statisches Scheduling möglich und damit ein vorhersagbares (deterministisches) Verhalten

▶ Ereignisgesteuerte Systeme

- ▶ Alle Ausführungen werden durch Ereignisse ausgelöst
- ▶ Garantierte Antwortzeiten
- ▶ Das zeitliche Einplanen von Prozessen (Scheduling) erfolgt dynamisch

Einzelprogrammbetrieb und Mehrprogrammbetrieb

▶ Einzelprogrammbetrieb (engl. Singletasking)

Zu jedem Zeitpunkt läuft nur ein einziges Programm

▶ Mehrprogrammbetrieb (engl. Multitasking)

- ▶ Mehrere Programme können gleichzeitig (bei mehreren CPUs) oder zeitlich verschachtelt (quasi-parallel) ausgeführt werden
- ▶ Prozesse werden in kurzen Abständen abwechselnd aktiviert
- ▶ Dadurch entsteht der Eindruck der Gleichzeitigkeit

Echtzeitbetriebssystem

- ▶ Ein Echtzeitbetriebssystem (engl. *real-time operating system*, *RTOS*) ist in der Lage Echtzeitanforderungen zu erfüllen.
- ▶ Das Zeitverhalten von RTOS muss dem von harten Echtzeitbedingungen genügen.
- ▶ RTOS haben daher spezielle Prozess-Scheduler und Speicherverwaltungs-Strategien
- ▶ RTOS unterstützen Fehlererkennungsmechanismen.
- ▶ Beispiele: FreeRTOS, PikeOS, QNX, RTEMS, ...

Zusammenfassung

Nach diesem Kapitel sollten Sie...

- ▶ ... wissen was ein Echtzeitsystem ist.
- ▶ ... die 4 Arten der Rechtzeitigkeit kennen.
- ▶ ... die unterschiedlichen Zeitbedingungen kennen.
- ▶ ... die 3 Echtzeitbedingungen kennen.