Konzepte von Betriebssystem-Komponenten
Ausnahme- / Unterbrechungsbehandlung

Sommersemester 2005
Uni Erlangen
Lehrstuhl für Informatik 4
Verteilte Systeme und Betriebsysteme

Tarek Gasmi
Tarek.Gasmi@informatik.stud.uni-erlangen.de

Gliederung

- ***Einführung***
- ***Traps***
- ***Interrupts***
- ***Der Unterschied zwischen Interrupts und Traps***
- ***Die Behandlung von Interrupts und Traps***
  - ***Termination model***
  - ***Resumption model***
- ***Hardware-Aktivitäten***
- ***Vektor-Tabelle***
- ***Software-Aktivitäten***
- ***Mehrere Unterbrechungsebenen***
- ***Zusammenfassung***

Ziel

- Unterschied zwischen Interrupts und Traps.
- Behandlungen der Unterbrechung

Einführung

- Die Unterbrechungen unterscheiden sich in:
  - Quelle
  - Synchronität
  - Vorhersagbarkeit
  - Reproduzierbarkeit
- Traps und Interrupts.

Traps(1)

- Traps werden durch das gerade laufende Prozess verursacht
- Ohne Behebung der Ausnahmebedingung ist eine Trap-Vermeidung unmöglich.
- Beispiele für Traps :
  - Überlauf bei einer arithmetischen Operation
  - Eine Division durch 0
  - Unbekannter Befehl
  - Systemaufruf
  - Ein Fehler bei einem Speicherzugriff (Z.B falscher Adressbereich oder ein unzulässige Art des Zugriffs)

Traps(2)

- die Hardware oder das Mikroprogramm des Systems entdeckt Traps.
  - Einsparung von Programmieraufwand und Ausführungszeit.
  - Mehr Sicherheit .

Interrupts (1)

- *Interrupts sind externe Unterbrechungen (Z.B. E/A-Geräte) .*
- *Beim Interrupt gibt der Prozessor die Kontrolle an einen Interrupt-Handler ab.*

Interrupts (2)

- *Die Behandlung der Ausnahmesituation muss nebeneffektfrei verlaufen*

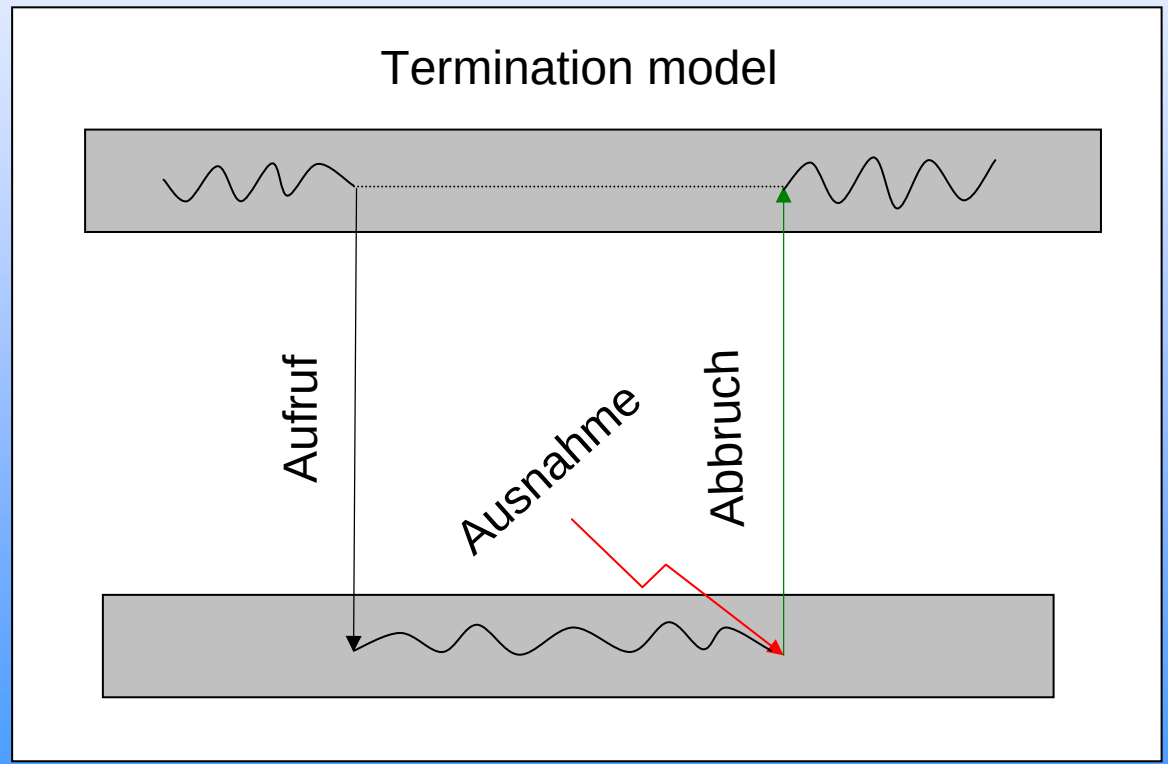
- *Beispiele für Interrupts:*
  - Timer
  - Eine Tast gedrückt
  - E/A-Signale

Der Unterschied zwischen Interrupts und Traps

- Ein Trap :
  - synchron
  - vorhersagbar
  - reproduzierbar
  - kein Interrupt
- Ein Interrupt
  - Asynchron
  - unvorhersagbar
  - nicht reproduzierbar
  - kein Trap

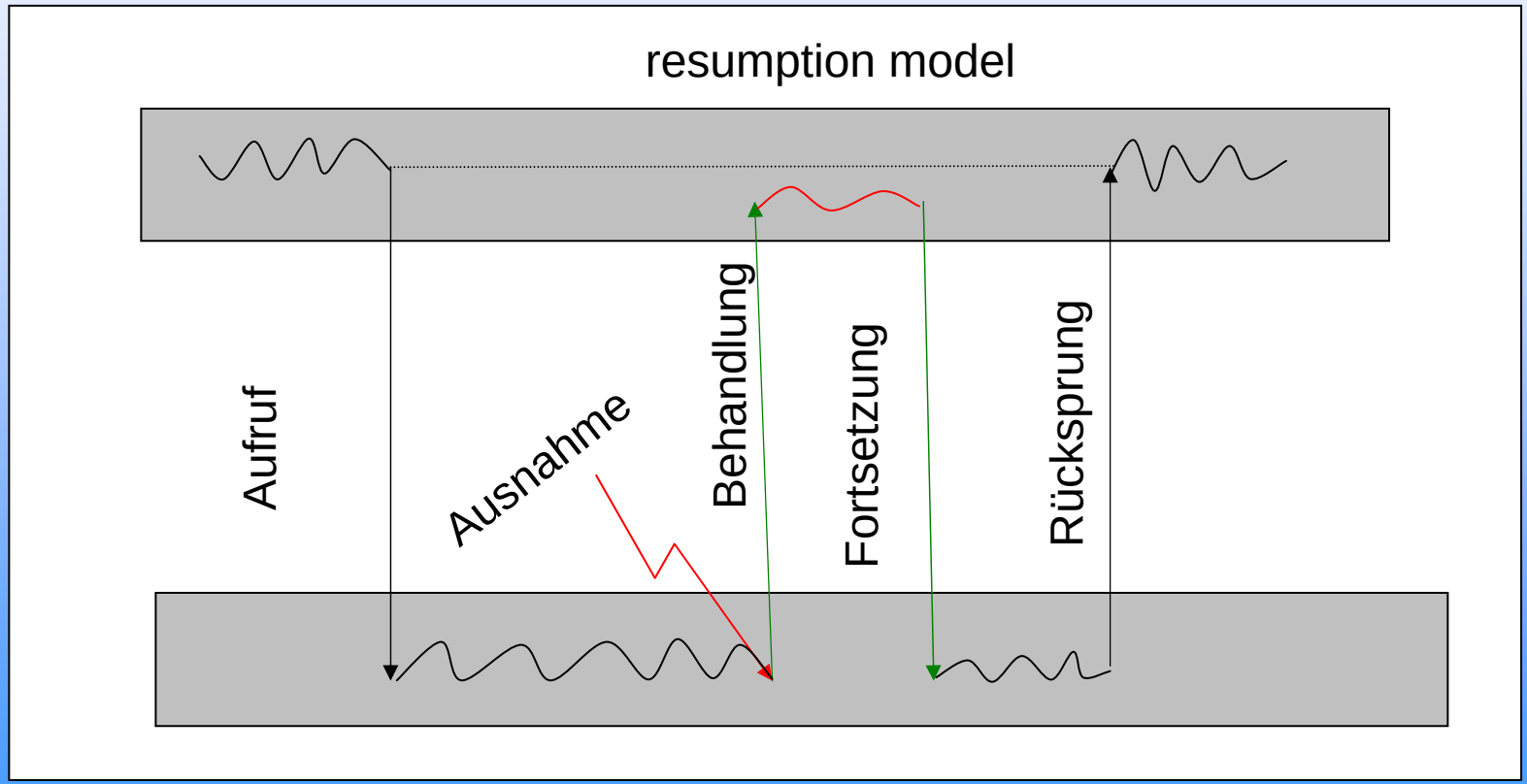
Die Behandlung der Interrupts und Traps(1)

- Termination model



Die Behandlung von Interrupts und Traps(2)

- Resumption model



Hardware-Aktivitäten

- Eine Komponente meldet ihren Unterbrechungswunsch an.
- Der Prioritätsencoder leitet immer nur den Interrupt mit der jeweils höchsten Priorität zur CPU weiter.
- Die CPU beendet normal den aktuellen Befehl und rettet sowohl ihr Statusregister wie auch den Befehlszähler in einen bestimmen Speicherbereich (Stack).
- Zu jedem Interrupt soll nun die entsprechenden Interrupt-Service-Routine aufgerufen werden. Der Anfang eines Unterprogramm , ist in der Vektor-Tabelle (fester Bereich in Hauptspeicher) abgelegt.

Vektor-Tabelle

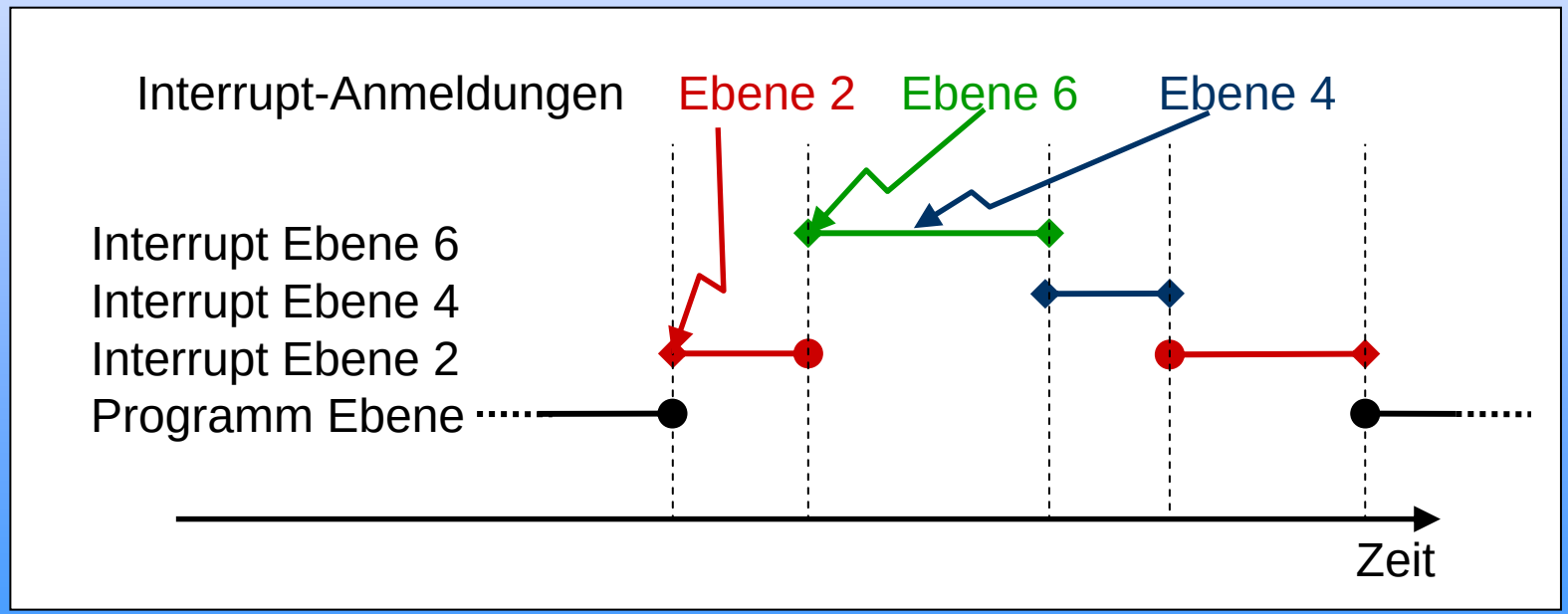
Adresse	Periorität	Interrupt-Quelle
20	15 (höchste)	Timer
24	14	Tastatur
28	-	Interrupt-Controller2
02C	5	Serielle Schnittstelle 2
30	4	Serielle Schnittstelle 1
34	3	Parallele Schnittstelle 2
38	2	Floppy Disk
03C	1 (niedrigste)	Parallele Schnittstelle 1
1C0	13	Echtzeituhr
1C4 bis 1D0	12-11-10-9	I/O-Kanal
1D4	8	Co-Prozessor
1D8	7	Harddisk
1DC	6	I/O-Kanal

Software-Aktivitäten

- Die Interrupt Service Routine stellt ein Unterprogramm dar. Also müssen zu nächst (wie bei jedem Unterprogramm) alle Register, die im Unterprogramm zum Einsatz kommen, auf den Stack gerettet werden.
- In der Interrupt Service Routine wird der Interrupt bearbeitet.
- Vor dem Rücksprung zum unterbrochenen Programm lädt die CPU aus dem Stack die geretteten Register (Status-Register und Befehlszähler).

Mehrere Unterbrechungsebenen

- Ein Interrupt kann den Interrupt einer niedrigeren Ebene (Priorität) unterbrechen, aber nicht umgekehrt.



Zusammenfassung

- *Der Unterschied zwischen Interrupt und Trap*
  - Ein Trap ist synchron, vorhersagbar, reproduzierbar und kein Interrupt.
  - Ein Interrupt ist Asynchron, unvorhersagbar, reproduzierbar und kein Trap.
- Die Behandlung der Interrupts und Traps
  - Termination model
  - Resumption model
- Hardware-Aktivitäten
- Software-Aktivitäten
- Mehrere Unterbrechungsebenen

Literatur

- *Jürgen Nehmer and Peter Sturm, März 2001, Systemsoftware: Grundlagen Moderner Betriebsystem, Dpunkt.*
- *Helmut Malz, 2004, Rechnerarchitektur, Broschiert-Vieueg-Verlagsgesellschaft.*
- *Carsten Vogt, April 2001, Betriebsysteme,Spektrum Akademischer Verlag.*
- *Andrew s. Tannenbaum James Goodman,15.Januar, Computerarchitektur: Strukturen Konzepte Grundlagen, Pearson Studium.*
- Wolfgang Schröder-Preikschat, S.S. 2004, Skript der Softwaresysteme I, URL: http://www4.informarik.uni-erlangen.de/Lehre/SS04/V_SOS1/Folien, 16-Juni-2004

Frage?