

Betriebssysteme (BS)

VL 9.1 – Fadenverwaltung – Betriebssystemfäden

Volkmar Sieh / Daniel Lohmann

Lehrstuhl für Informatik 4
Verteilte Systeme und Betriebssysteme

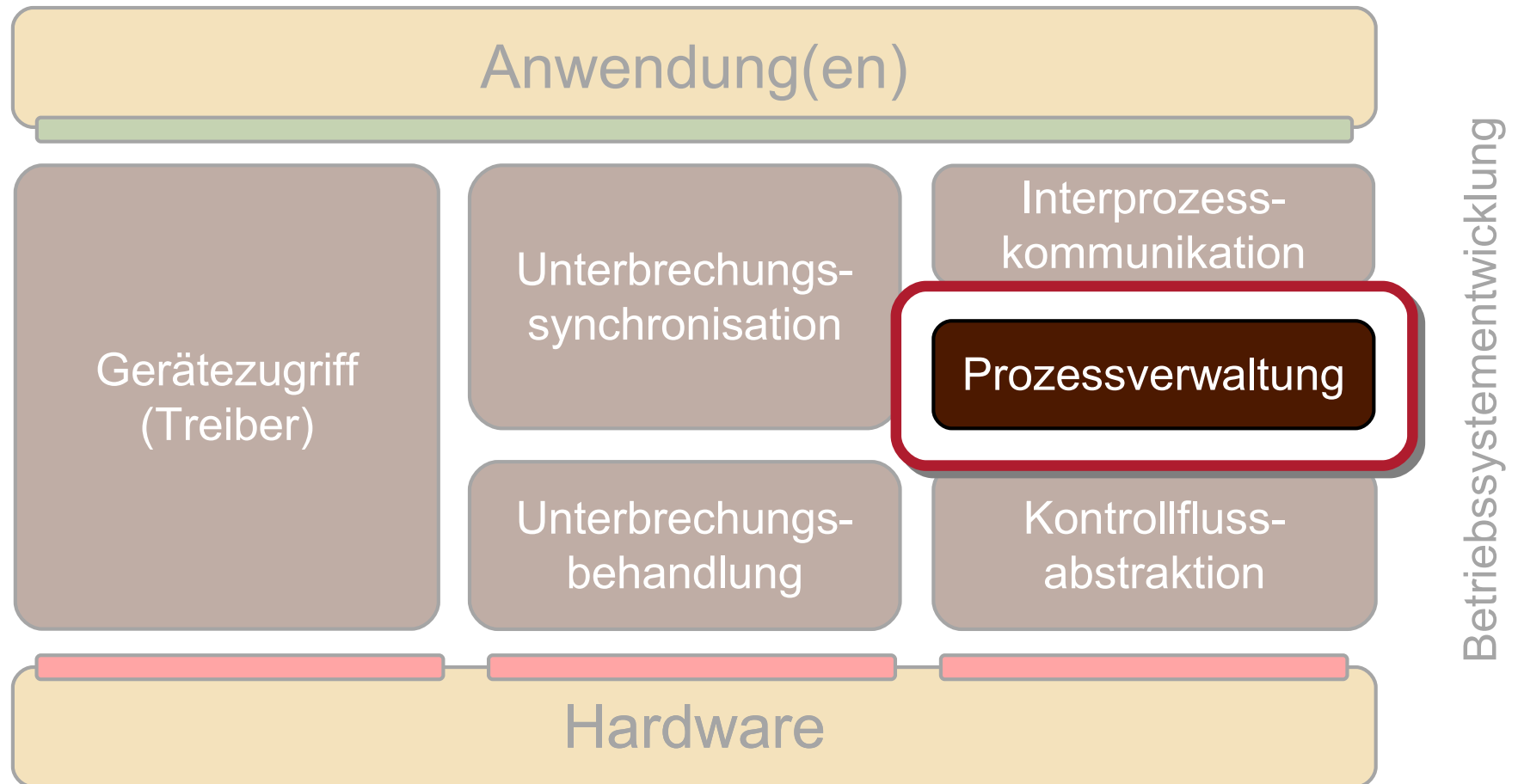
Friedrich-Alexander-Universität
Erlangen Nürnberg

WS 21 – 20. Dezember 2021



https://www4.cs.fau.de/Lehre/WS21/V_BS

Überblick: Einordnung dieser VL



Agenda

Einordnung
Betriebssystemfäden
Motivation
Kooperativer Fadenwechsel
Präemptiver Fadenwechsel
Ablaufplanung
Zusammenfassung



Agenda

Einordnung

Betriebssystemfäden

Motivation

Kooperativer Fadenwechsel

Präemptiver Fadenwechsel

Ablaufplanung

Zusammenfassung



Betriebssystemfäden: Motivation

- Ansatz: Anwendungen „unbemerkt“ als eigenständige Fäden ausführen
 - eine BS-Koroutine pro Anwendung
 - Aktivierung der Anwendung erfolgt durch Aufruf
 - Koroutinenwechsel erfolgt indirekt durch Systemaufruf
- Vorteile
 - unabhängige Anwendungsentwicklung
 - Ablaufplanung (*Scheduling*) wird zentral implementiert
 - bei E/A kann eine Anwendung einfach vom BS „blockiert“ und später wieder „geweckt“ werden
 - zusätzlicher Entzugsmechanismus (*preemption mechanism*) kann die Monopolisierung der CPU verhindern



Kooperativer Fadenwechsel

Anwendungen

<app1>

<app2>

Betriebssystem

<kickoff>

<kickoff'>

<resume>

<schedule>

upcall app1

resume

upcall app2

resume

resume

ref

ref

ref

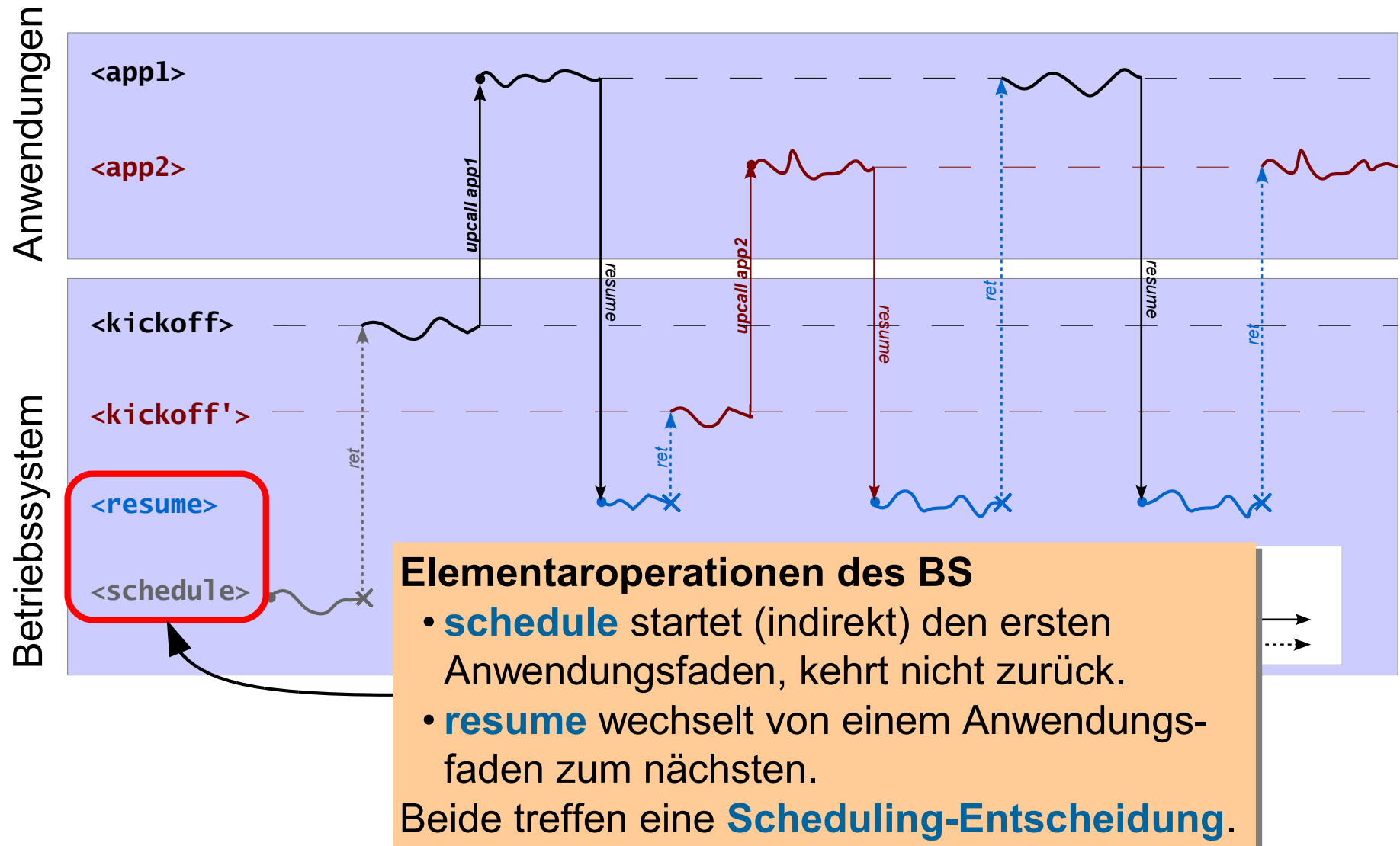
ref

Legende

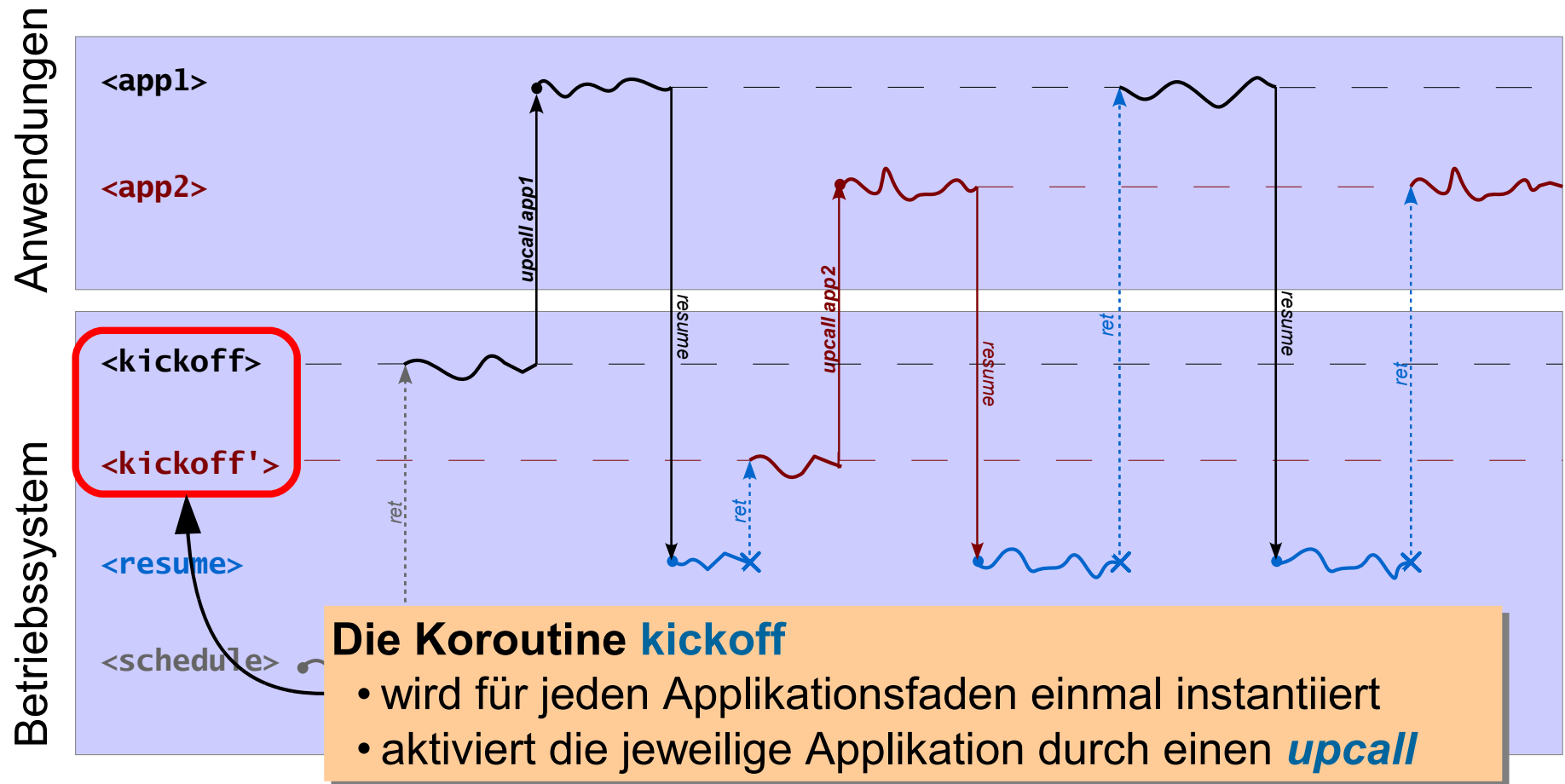
Kontrollflusserzeugung:	•
Kontrollflusszerstörung:	X
explizite (Re-)Aktivierung:	→
implizite (Re-)Aktivierung:	- - - - ->



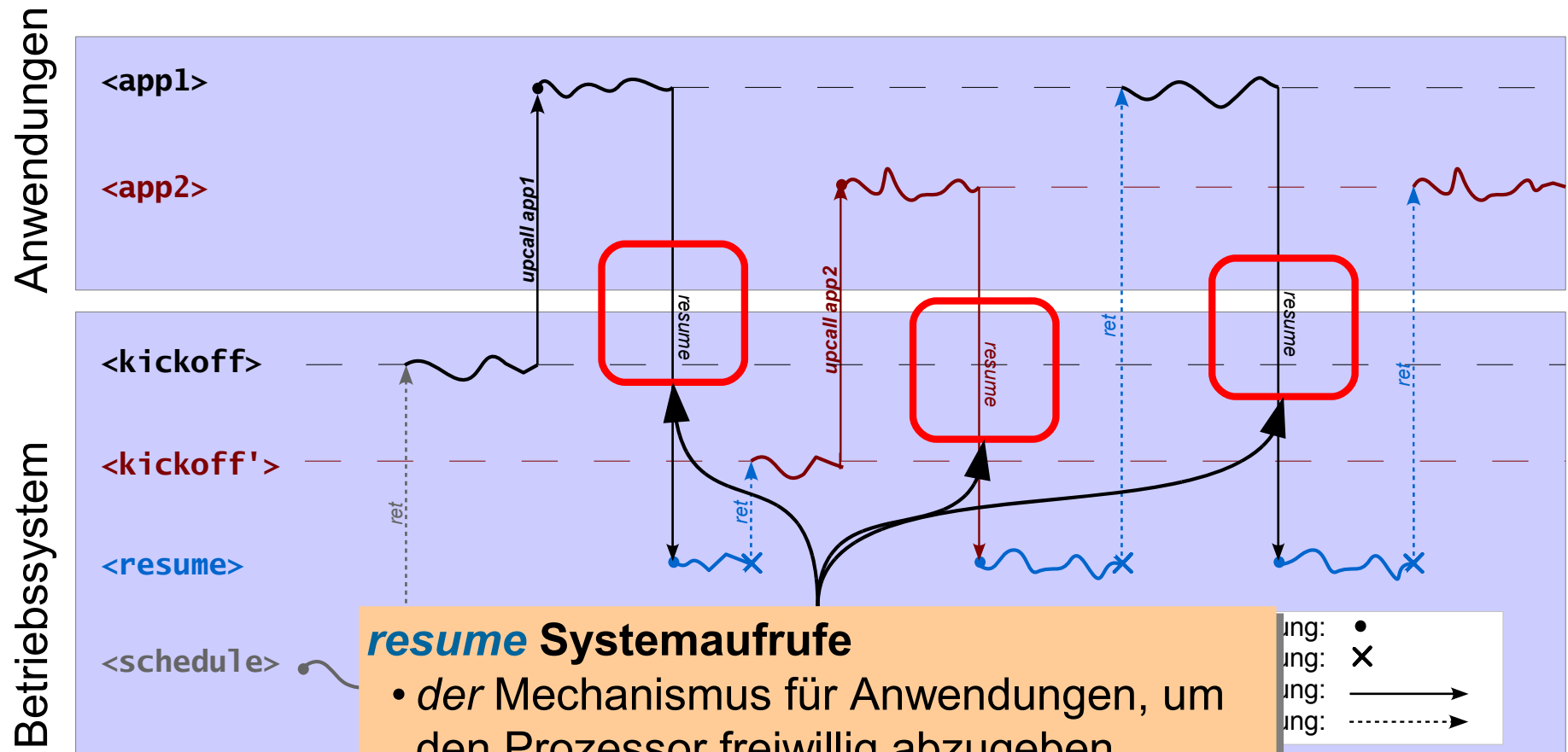
Kooperativer Fadenwechsel



Kooperativer Fadenwechsel



Kooperativer Fadenwechsel



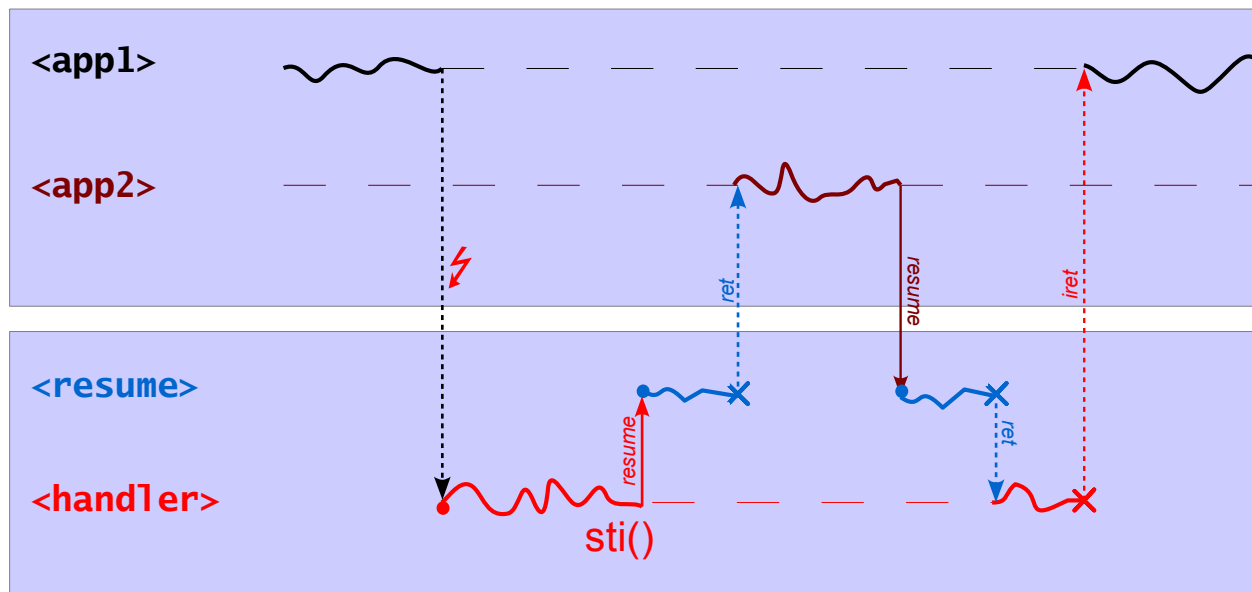
resume Systemaufrufe

- der Mechanismus für Anwendungen, um den Prozessor freiwillig abzugeben
- ggf. verbunden mit einem Moduswechsel der CPU (in diesem Fall wird noch ein *Wrapper* benötigt)



Präemptiver Fadenwechsel

- CPU-Entzug durch Zeitgeberunterbrechung
 - die Unterbrechung ist „nur“ ein impliziter Aufruf
 - Behandlungsroutine kann *resume* aufrufen



Achtung: So geht es normalerweise *nicht*, denn *resume* trifft eine **Scheduling-Entscheidung**. Bei den notwendigen Datenstrukturen ist **Unterbrechungssynchronisation** zu beachten!

