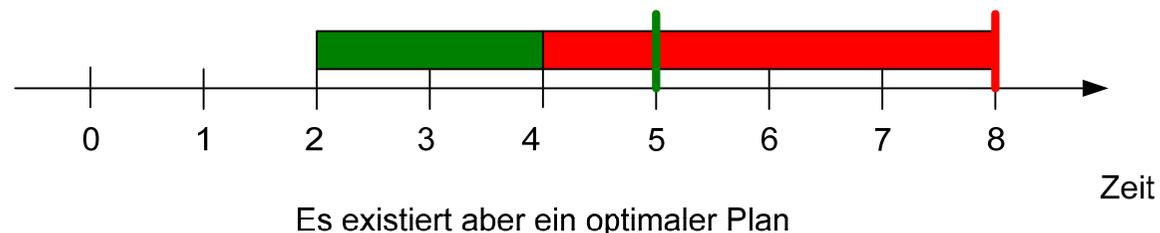
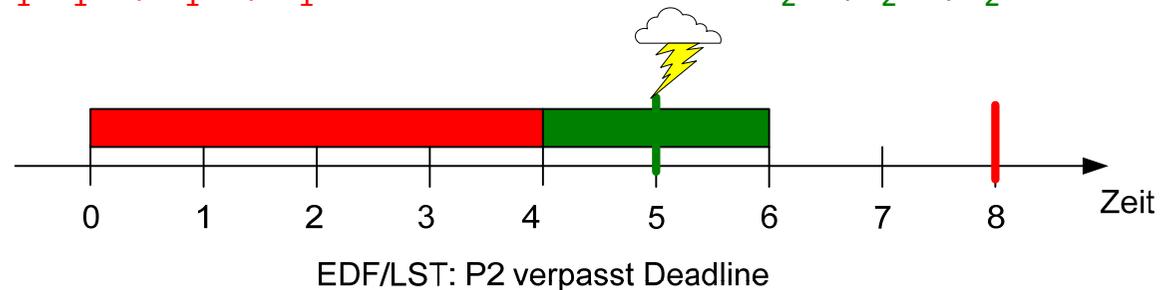


## Versagen von EDF bei unterschiedlichen Bereitzeiten

- Haben die Prozesse unterschiedliche Bereitzeiten, so kann EDF versagen.
- Beispiel:  $P_1: r_1=0; e_1=4; d_1=8$        $P_2: r_2=2; e_2=2; d_2=5$



- **Anmerkung:** Jedes prioritätsgesteuerte, **nicht präemptive** Verfahren versagt bei diesem Beispiel, da ein solches Verfahren nie eine Zuweisung des Prozessors an einen lafbereiten Prozess, falls ein solcher vorhanden ist, unterlässt.

## Modifikationen

- Die Optimalität der Verfahren kann durch folgende Änderungen sichergestellt werden:
  - Präemptive Strategie
  - Neuplanung beim Erreichen einer neuen Bereitzeit
  - Einplanung nur derjenigen Prozesse, deren Bereitzeit erreicht ist  
→ Entspricht einer Neuplanung, falls ein Prozess aktiv wird.
- Bei Least Slack Time müssen zusätzlich Zeitscheiben für Prozesse mit gleichem Spielraum eingeführt werden, um ein ständiges Hin- und Her Schalten zwischen Prozessen zu verhindern.
- Generell kann gezeigt werden, dass die Verwendung von EDF die Anzahl der Kontextwechsel in Bezug auf Online-Scheduling-Verfahren minimiert (siehe Paper von Buttazzo)

## Zeitplanung auf Mehrprozessorsystemen

- Fakten zum Scheduling auf Mehrprozessorsystemen (Beispiele folgen):
  - EDF nicht optimal, egal ob präemptiv oder nicht präemptive Strategie
  - LST ist nur dann optimal, falls alle Bereitzeitpunkte  $r_i$  gleich
  - korrekte Zuteilungsalgorithmen erfordern das Abarbeiten von Suchbäumen mit NP-Aufwand oder geeignete Heuristiken
  - Beweisidee zur Optimalität von LST bei gleichen Bereitzeitpunkten: Der Prozessor wird immer dem Prozess mit geringstem Spielraum zugewiesen, d.h. wenn bei LST eine Zeitüberschreitung auftritt, dann auch, falls die CPU einem Prozess mit größerem Spielraum zugewiesen worden wäre.

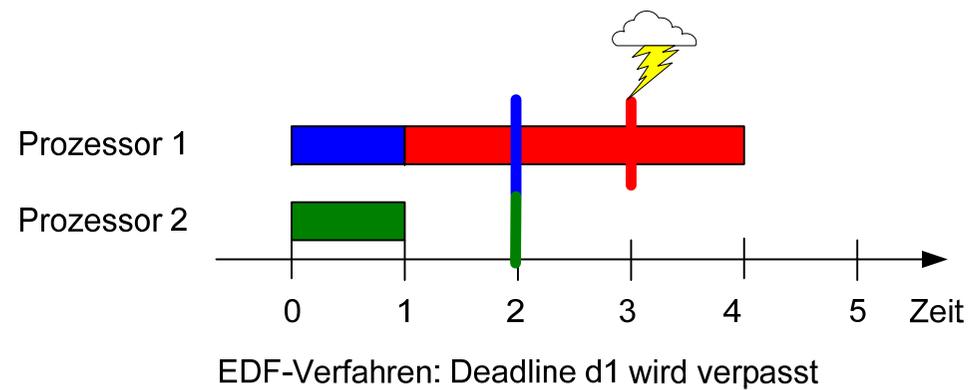
## Beispiel: Versagen von EDF

- 2 Prozessoren, 3 Prozesse:

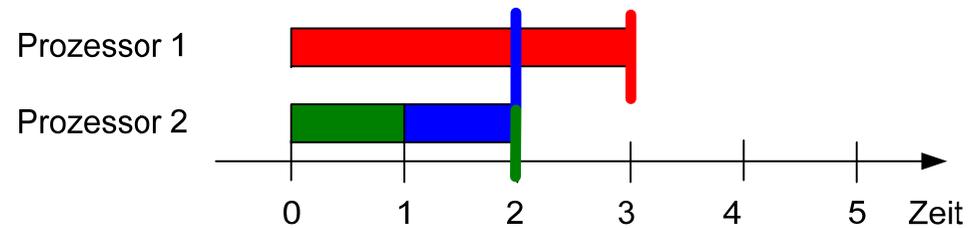
$P_1: r_1=0; e_1=3; d_1=3;$

$P_2: r_2=0; e_2=1; d_2=2;$

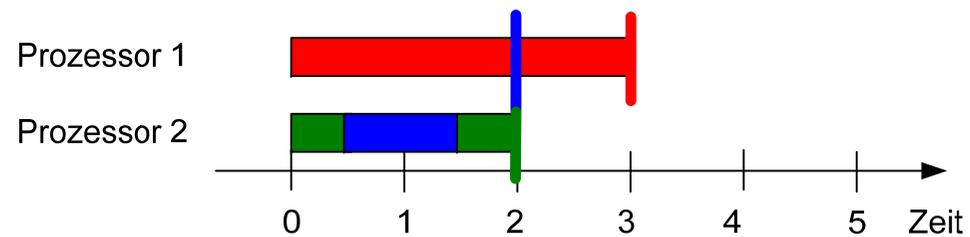
$P_3: r_3=0; e_3=1; d_3=2;$



## Beispiel: Optimaler Plan und LST-Verfahren



Optimaler Plan



LST-Verfahren mit  $\Delta t = 0.5$

## Beispiel: Versagen von LST

- 2 Prozessoren, 5 Prozesse,  $\Delta_t=0,5$ :

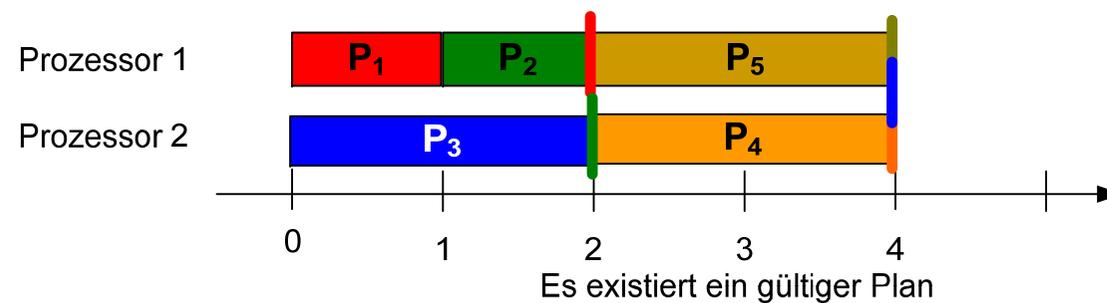
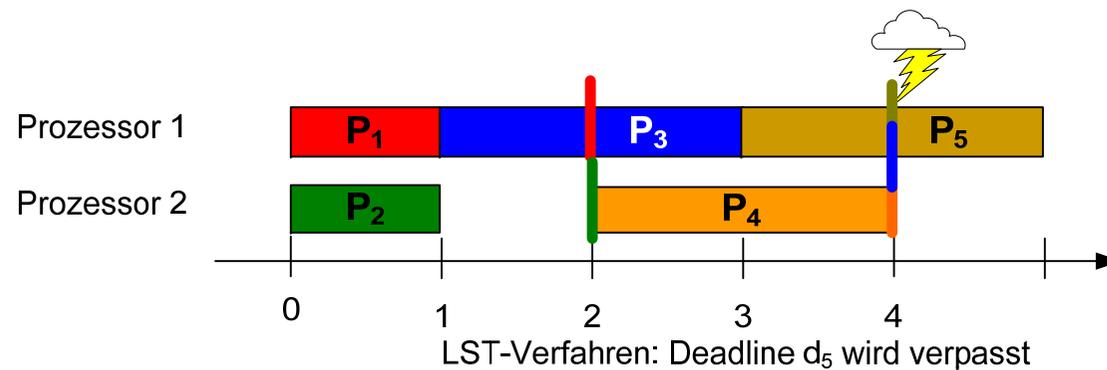
$P_1$ :  $r_1=0$ ;  $e_1=1$ ;  $d_1=2$ ;

$P_2$ :  $r_2=0$ ;  $e_2=1$ ;  $d_2=2$ ;

$P_3$ :  $r_3=0$ ;  $e_3=2$ ;  $d_3=4$ ;

$P_4$ :  $r_4=2$ ;  $e_4=2$ ;  $d_4=4$ ;

$P_5$ :  $r_5=2$ ;  $e_5=2$ ;  $d_5=4$ ;



## Versagen von präemptiven Schedulingverfahren

- Jeder präemptiver Algorithmus versagt, wenn die Bereitstellzeiten unterschiedlich sind und nicht im Voraus bekannt sind.

Beweis:

- $n$  CPUs und  $n-2$  Prozesse ohne Spielraum ( $n-2$  Prozesse müssen sofort auf  $n-2$  Prozessoren ausgeführt werden)  $\Rightarrow$  Reduzierung des Problems auf 2-Prozessor-Problem
- Drei weitere Prozesse sind vorhanden und müssen eingeplant werden.
- Die Reihenfolge der Abarbeitung ist von der Strategie abhängig, in jedem Fall kann aber folgender Fall konstruiert werden, so dass:
  - es zu einer Fristverletzung kommt,
  - aber ein gültiger Plan existiert.

## Fortsetzung Beweis

- Szenario:

$P_1: r_1=0; e_1=1; d_1=1;$

$P_2: r_2=0; e_2=2; d_2=4;$

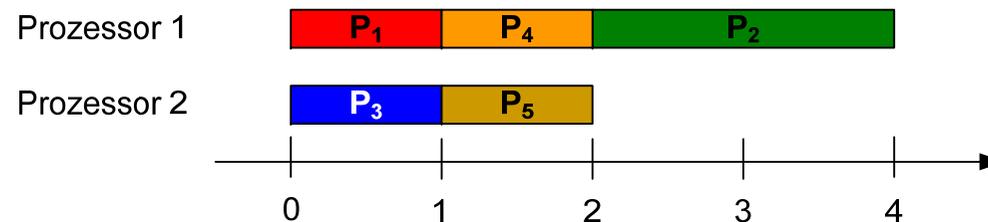
$P_3: r_3=0; e_3=1; d_3=2;$

→ Prozess  $P_1$  (kein Spielraum) muss sofort auf CPU1 ausgeführt werden.

→ Es gibt je nach Strategie zwei Fälle zu betrachten: P2 oder P3 wird zunächst auf CPU2 ausgeführt.

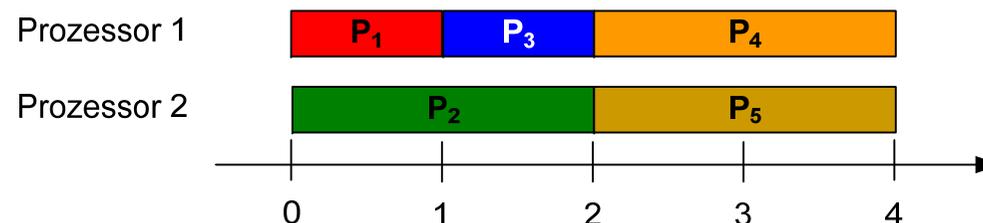
## 1. Fall

- $P_2$  wird zum Zeitpunkt 0 auf CPU2 ausgeführt.
  - Zum Zeitpunkt 1 muss dann  $P_3$  (ohne Spielraum) ausgeführt werden.
  - Zum Zeitpunkt 1 treffen aber zwei weitere Prozesse  $P_4$  und  $P_5$  mit Frist 2 und Ausführungsdauer 1 ein.
- Es gibt drei Prozesse ohne Spielraum, aber nur zwei Prozessoren.
- Aber es gibt einen gültigen Ausführungsplan:



## 2. Fall

- $P_3$  wird zum Zeitpunkt 0 auf CPU2 ausgeführt.
    - Zum Zeitpunkt 1 sind  $P_1$  und  $P_3$  beendet.
    - Zum Zeitpunkt 1 beginnt  $P_2$  seine Ausführung.
    - Zum Zeitpunkt 2 treffen aber zwei weitere Prozesse  $P_4$  und  $P_5$  mit Deadline 4 und Ausführungsdauer 2 ein.
- ⇒ Anstelle der zum Zeitpunkt 2 noch notwendigen 5 Ausführungseinheiten sind nur 4 vorhanden.
- Aber es gibt einen gültigen Ausführungsplan:



## Strategien in der Praxis

- Die Strategien EDF und LST werden in der Praxis selten angewandt. Gründe:
  - In der Realität sind keine abgeschlossenen Systeme vorhanden (Alarmer, Unterbrechungen erfordern eine dynamische Planung)
  - Bereitzeiten sind nur bei zyklischen Prozessen oder Terminprozessen bekannt.
  - Die Abschätzung der Laufzeit sehr schwierig ist (siehe Exkurs).
  - Synchronisation, Kommunikation und gemeinsame Betriebsmittel verletzen die Forderung nach Unabhängigkeit der Prozesse.

## Ansatz in der Praxis

- Zumeist basiert das Scheduling auf der Zuweisung von statischen Prioritäten.
- Prioritäten werden zumeist durch natürliche Zahlen zwischen 0 und 255 ausgedrückt. Die höchste Priorität kann dabei sowohl 0 (z.B. in VxWorks) als auch 255 (z.B. in POSIX) sein.
- Die Priorität ergibt sich aus der Wichtigkeit des technischen Prozesses und der Abschätzung der Laufzeiten und Spielräume. Die Festlegung erfolgt dabei durch den Entwickler.
- Bei gleicher Priorität wird zumeist eine FIFO-Strategie (d.h. ein Prozess läuft solange, bis er entweder beendet ist oder aber ein Prozess höherer Priorität eintrifft) angewandt.  
**Alternative** Round Robin: Alle lafbereiten Prozesse mit der höchsten Priorität erhalten jeweils für eine im Voraus festgelegte Zeitdauer die CPU.