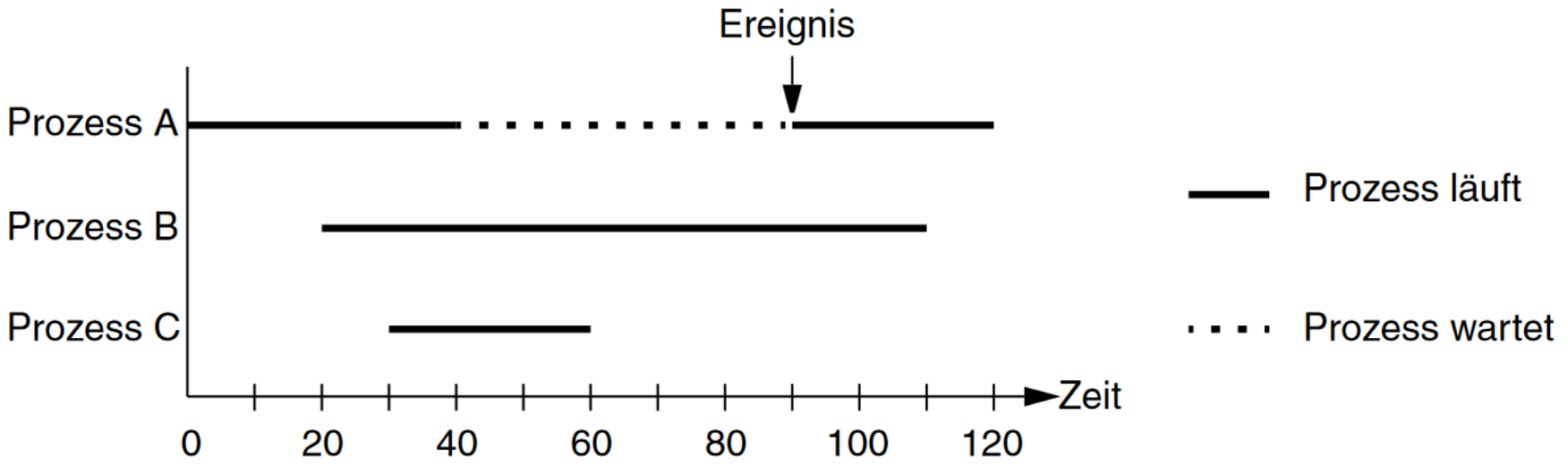
Prozessorzuteilungsstrategien (X)

- **Beispielszenario zur Betrachtung:**

  Die Prozesse reihen sich wie unten abgebildet in die Prozessor-Warteschlange ein.



Quelle: *[BS15]*

Prozessorzuteilungsstrategien (XI)

- FIFO: First In First Out bzw. FCFS (First Come First Served)

  Textuelle Beschreibung:

  Beim FIFO-Scheduling erfolgt die Prozessor-Zuteilung nach dem Senioritätsprinzip.

  Dementsprechend erhält der am längsten wartende Prozess als erster die CPU zugeteilt.

  In einem Szenario, in dem zwei Prozesse zeitgleich ablaufbereit werden, kann das FIFO-Prinzip nicht angewendet werden, sodass die CPU-Zuteilung nach dem Zufallsprinzip erfolgt.

Prozessorzuteilungsstrategien (XII)

- FIFO: First In First Out bzw. FCFS (First Come First Served)

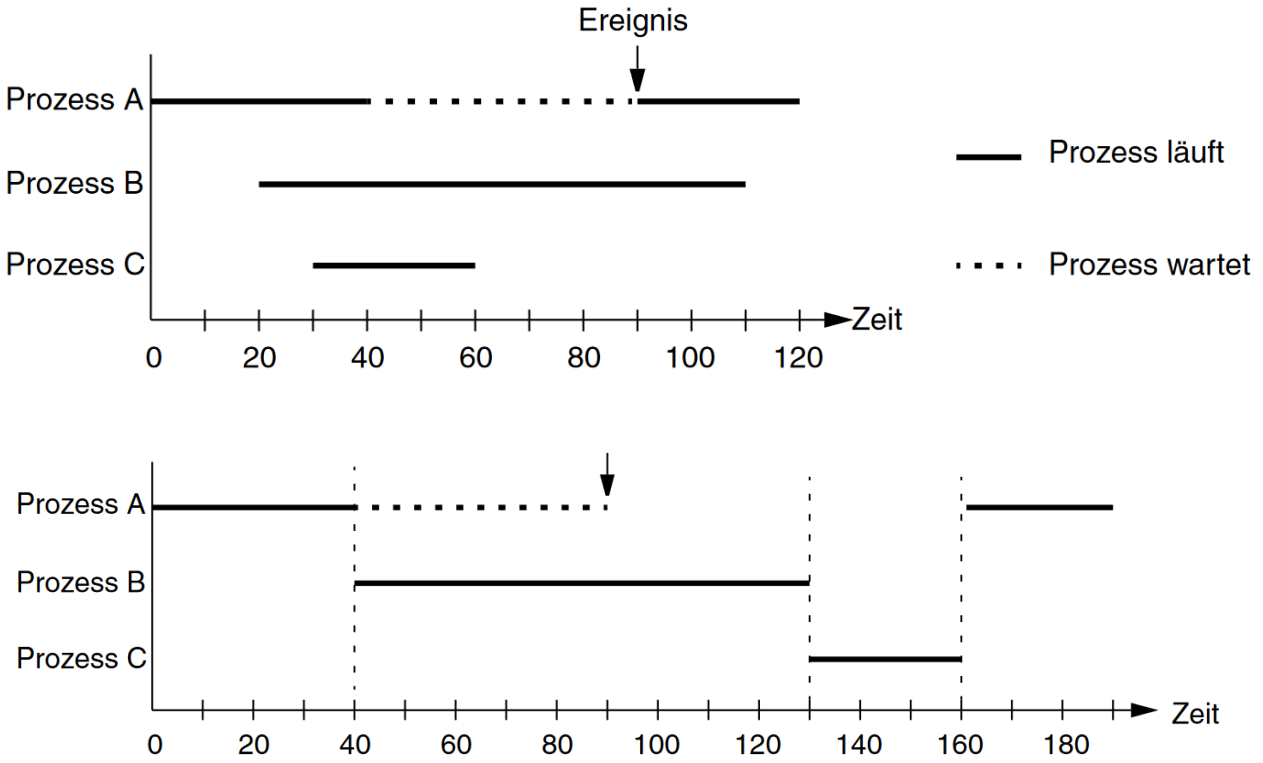
Rahmendaten

Verdrängend?	Zeitpunkt der Prozessor-neuzuteilung	Sonderfälle:	Vorteile	Nachteile
Nein	• Wenn auf CPU laufender Prozess blockiert • Wenn auf CPU laufender Prozess terminiert	Bei zeitgleicher Einreihung von Prozessen: • Zufallsprinzip entscheidet, welcher Prozess auf CPU laufen darf	• Minimaler Verwaltungsaufwand	• Funktioniert nicht mehr, wenn ein Prozess CPU nicht abgibt • Reaktionszeiten abhängig von den Prozessen (Sog. Konvoieffekt)

Prozessorzuteilungsstrategien (XIII)

- FIFO: First In First Out bzw. FCFS (First Come First Served)

Ablaufdiagramm:



Quelle: *[BS15]*

Prozessorzuteilungsstrategien (XIV)

- SJF: Shortest Job First oder Shortest Process Next (SPN)

  Textuelle Beschreibung:

  Bei dieser Art des Schedulings bekommt derjenige Prozess die CPU zugeteilt, der den kleinsten zu erwartenden Rechenbedarf hat. Vor diesem Hintergrund funktioniert der Algorithmus nur, wenn Angaben über den Rechenbedarf der einzelnen Prozesse zugänglich sind. Dies ist der Fall bei Stapelaufträgen oder aufgrund von Schätzwerten, die auf Basis der Vergangenheit eine Vorhersage treffen.

Prozessorzuteilungsstrategien (XV)

- SJF: Shortest Job First oder Shortest Process Next (SPN)

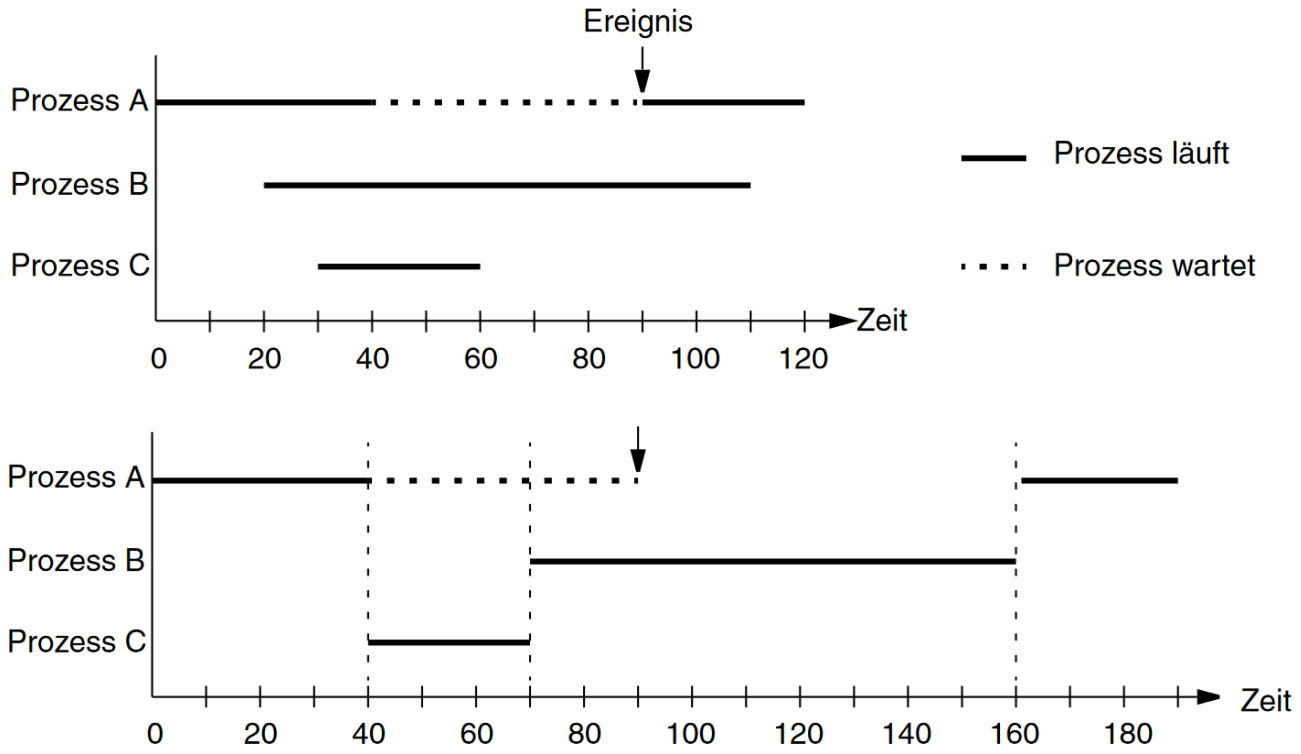
Rahmendaten

Verdrängend?	Zeitpunkt der Prozessor-neuzuteilung	Sonderfälle:	Vorteile	Nachteile
Nein	• Wenn auf CPU laufender Prozess blockiert • Wenn auf CPU laufender Prozess terminiert	-	• Prozesse mit kleinem Rechenbedarf werden bevorzugt und warten nicht auf langläufige Prozesse	• Langläufige Prozesse können verhungern, wenn es genügend Prozesse mit kleinem Rechenbedarf gibt

Prozessorzuteilungsstrategien (XVI)

- SJF: Shortest Job First oder Shortest Process Next (SPN)

Ablaufdiagramm:



Quelle: *[BS15]*

Prozessorzuteilungsstrategien (XVII)

- SRT: Shortest Remaining Time

  Textuelle Beschreibung:

  Dieser Algorithmus ist eine Abwandlung des SJF-Algorithmus dahingehend, als dass die Prozesszuteilung anhand des kleinsten verbleibenden Rechenzeitbedarfs erfolgt. Weiter gilt der SRT-Algorithmus als verdrängend, sodass eine Neuzuteilung zu allen möglichen Zeitpunkten erfolgen kann, wenn ein anderer Prozess weniger Rest-Rechenzeit benötigt.

Prozessorzuteilungsstrategien (XVIII)

- SRT: Shortest Remaining Time

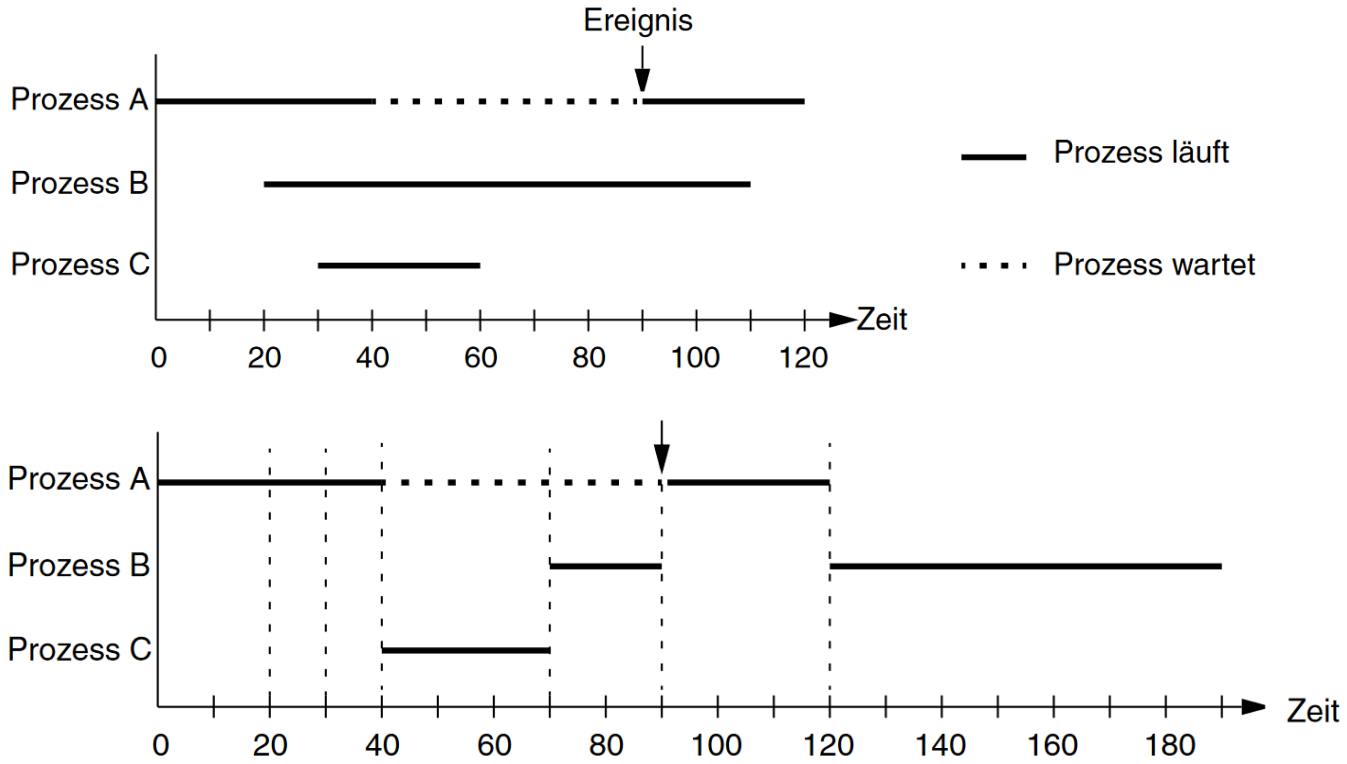
Rahmendaten

Verdrängend?	Zeitpunkt der Prozessor-neuzuteilung	Sonderfälle:	Vorteile	Nachteile
Ja	• Alle möglichen Zeitpunkte entsprechend dieser Folie.	-	• Prozesse mit kleinem Rest-Rechenbedarf werden bevorzugt und warten nicht auf langläufige Prozesse	• Langläufige Prozesse können verhungern, wenn es genügend Prozesse mit kleinem Rest-Rechenbedarf gibt

Prozessorzuteilungsstrategien (XIX)

- SRT: Shortest Remaining Time

Ablaufdiagramm:



Quelle: *[BS15]*

Prozessorzuteilungsstrategien (XX)

- RR: Round Robin

  Textuelle Beschreibung:

  Das Round Robin Verfahren baut auf dem FIFO-Algorithmus auf und erweitert diesen um Timesharing. Dazu erhält jeder Prozess ein Zeitquantum an Rechenzeit zugeteilt, indem er ununterbrochen die CPU nutzen darf. Es handelt sich hierbei um ein verdrängendes Scheduling-Verfahren, bei dem einem Prozess die Rechenzeit auch dann entzogen wird, wenn sein Zeitquantum abgelaufen ist.

Prozessorzuteilungsstrategien (XXI)

- RR: Round Robin

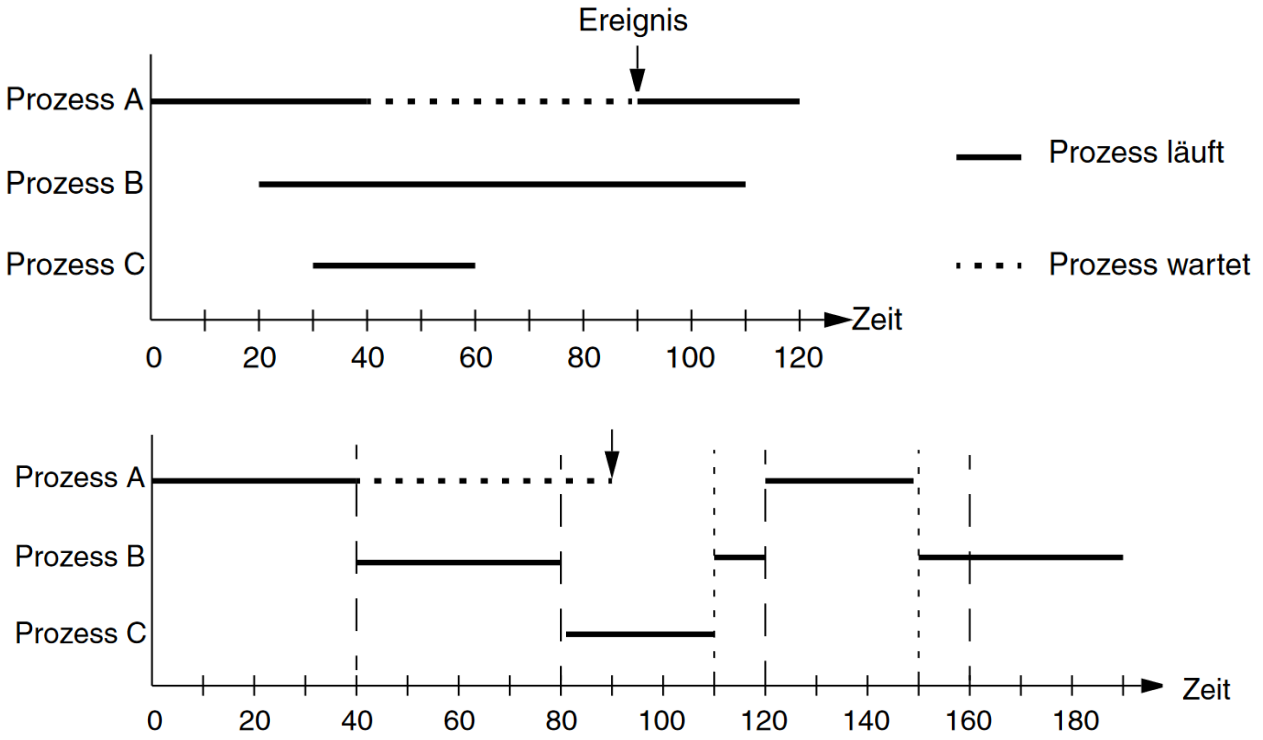
Rahmendaten

Verdrängend?	Zeitpunkt der Prozessor-neuzuteilung	Sonderfälle:	Vorteile	Nachteile
Ja	• Wenn auf CPU laufender Prozess blockiert • Wenn auf CPU laufender Prozess terminiert • Wenn Zeitscheibe abgelaufen	-	• Gleichbehandlung aller Prozesse • Busy-Waits können System nicht lahmlegen	• Langläufige Prozesse benötigen viel Zeit bis zu ihrer Terminierung

Prozessorzuteilungsstrategien (XXII)

- RR: Round Robin

Ablaufdiagramm:



(Zeitscheibengröße: 40 Zeiteinheiten)

Quelle: *[BS15]*

Prozessorzuteilungsstrategien (XXIII)

- ML: Multi-Level Priority

  Textuelle Beschreibung:

  Das Multi-Level Priority Scheduling basiert primär auf der Vergabe von Prioritäten an die einzelnen Prozesse. Die Priorität wird dabei i. d. R. mit Hilfe von Zahlenwerten beschrieben. Die Strategie ist verdrängend, sodass zu jedem Zeitpunkt der Prozess mit der höchsten Priorität die CPU zugeteilt hat. Sind mehrere Prozesse mit gleicher Priorität ablaufbereit, gilt sekundär das FIFO-Prinzip zur Auswahl des nächsten Prozesses.

Prozessorzuteilungsstrategien (XXIV)

- ML: Multi-Level Priority

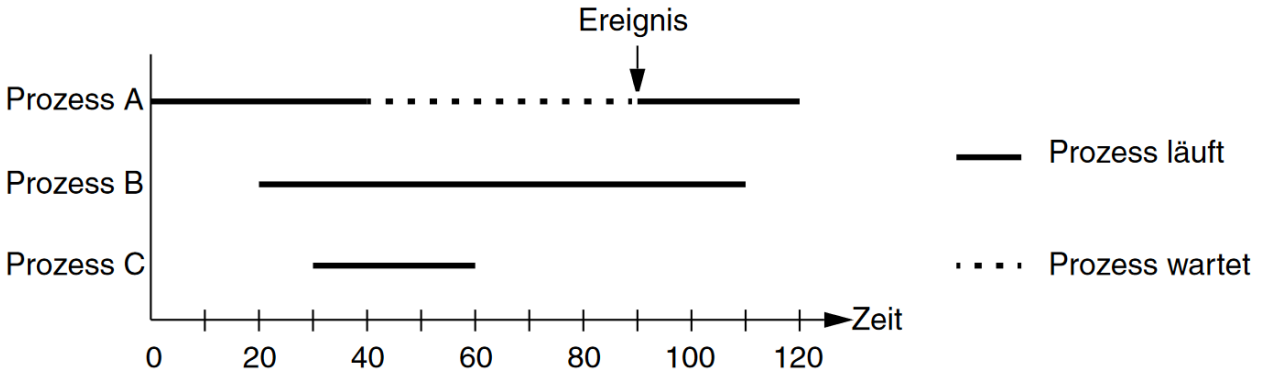
Rahmendaten

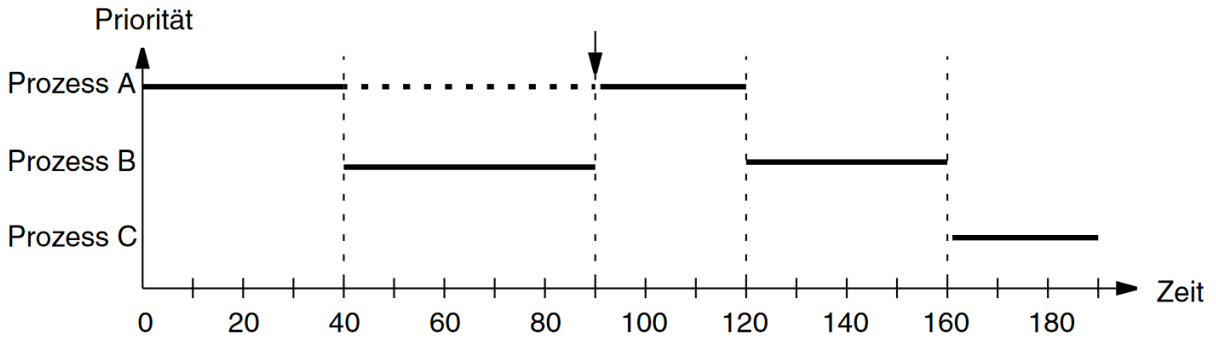
Verdrängend?	Zeitpunkt der Prozessor-neuzuteilung	Sonderfälle:	Vorteile	Nachteile
Ja	• Alle möglichen Zeitpunkte entsprechend dieser Folie.	• Mehrere Prozesse mit gleicher Priorität ablaufbereit: in diesem Fall Auswahl des nächsten Prozesses anhand FIFO-Prinzip.	• Anwender kann anhand der Prozess-Prioritäten CPU-Zuteilung besser steuern	• „Monopolisierung“ der CPU durch höherpriorisierte Prozesse

Prozessorzuteilungsstrategien (XXV)

- ML: Multi-Level Priority

Ablaufdiagramm:





(A höchste Priorität, C niedrigste Priorität)

***Quelle:** [BS15]*

Prozessorzuteilungsstrategien (XXVI)

- MLF: Multi-Level Feedback

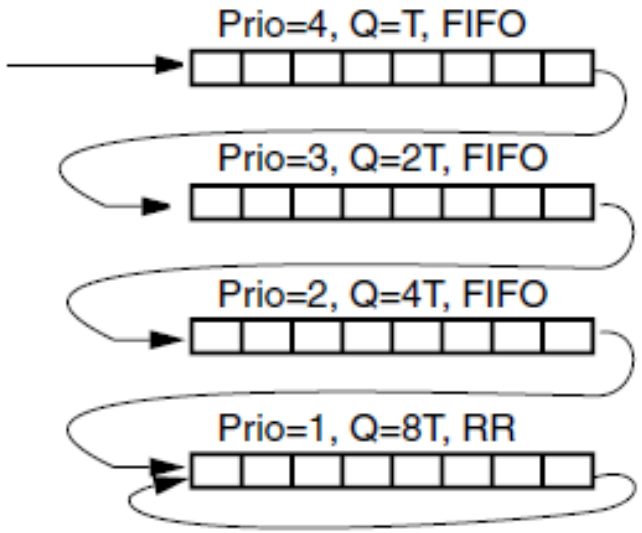
  Textuelle Beschreibung:

  Bei diesem Scheduling-Algorithmus werden Prozesse anhand ihrer bereits aufgelaufenen Rechenzeit beurteilt. Hierzu existieren mehrere Prioritätsstufen mit jeweils einer eigenen Prozess-Warteschlange. Jeder Prioritätswarteschlange ist dabei ein festes Zeitquantum zugeordnet. Beim Prozess-Start erhält ein Prozess die höchste Priorität. Existieren innerhalb einer Prioritätswarteschlange mehrere ablaufbereite Prozesse, wird das FIFO-Prinzip angewendet. Sobald ein Prozess innerhalb einer Prioritätswarteschlange sein Zeitquantum ausgeschöpft hat, wird er eine Prioritätswarteschlange niedriger eingereiht und erhält ein größeres Zeitquantum. Erreicht ein Prozess die niedrigste Prioritätsstufe, läuft seine CPU-Zuteilung nach dem Round-Robin Verfahren ab.

Prozessorzuteilungsstrategien (XXVII)

- MLF: Multi-Level Feedback

  Schematische Darstellung:



Prio: Priorität
Q: Zeitquantum
T: Zeitscheibengröße

Quelle: *[BS15]*

Prozessorzuteilungsstrategien (XXVIII)

- MLF: Multi-Level Feedback

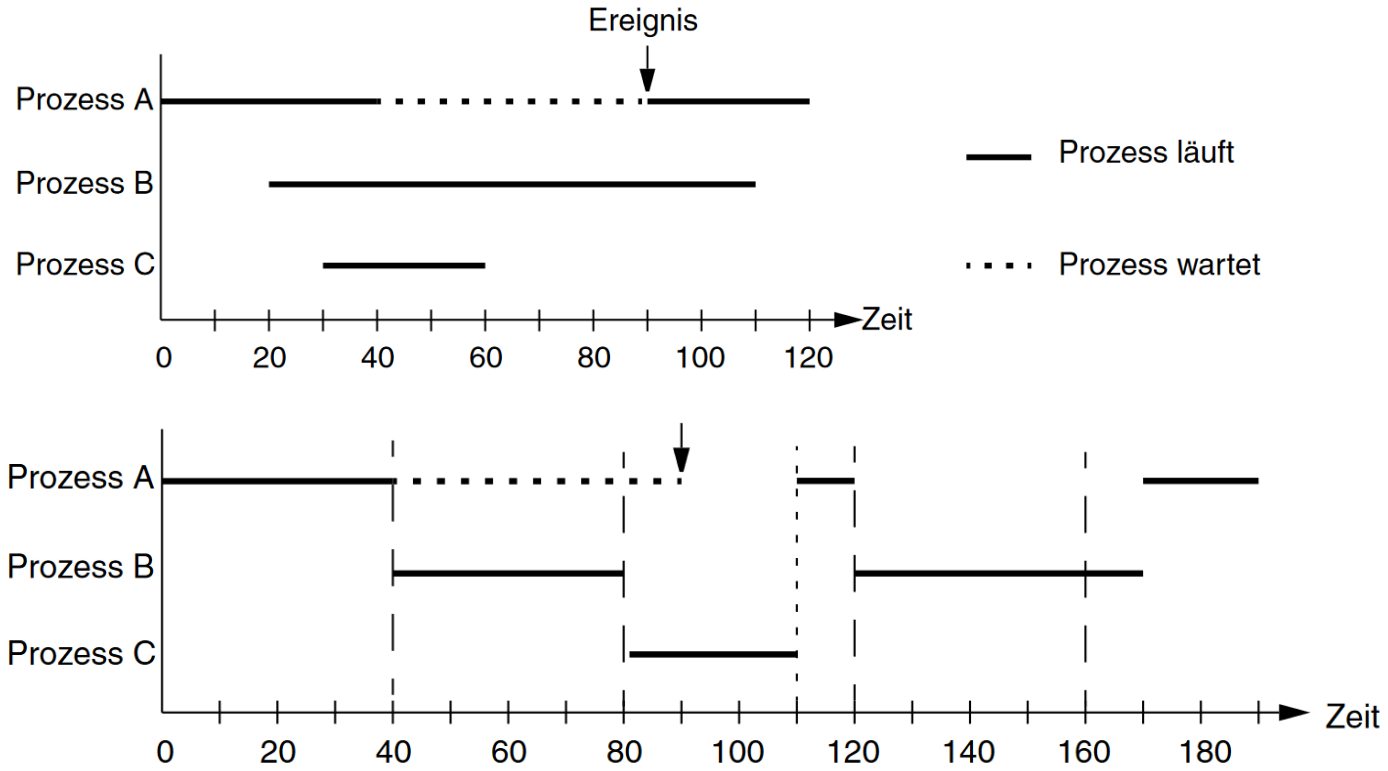
Rahmendaten

Verdrängend?	Zeitpunkt der Prozessor-neuzuteilung	Sonderfälle:	Vorteile	Nachteile
Ja	• Wenn auf CPU laufender Prozess blockiert • Wenn auf CPU laufender Prozess terminiert • Wenn Zeitscheibe abgelaufen	• Mehrere Prozesse in gleicher Prioritäts-warteschlange: in diesem Fall Auswahl des nächsten Prozesses anhand FIFO-Prinzip. • Prozess in niedrigster Prioritäts-warteschlange: Zuteilung anhand RR-Verfahren	• E/A-lastige Prozesse werden bevorzugt → Vorteil für interaktive Systeme • Keine Vorkenntnisse über erwartete Rechenzeiten notwendig	• Langlaufende Prozesse erhalten schnell niedrigste Priorität

Prozessorzuteilungsstrategien (XXIX)

- MLF: Multi-Level Feedback

Ablaufdiagramm:



(Zeitscheibengröße T: 40 Zeiteinheiten, 4 Prioritäten: 4 höchste, 1 niedrigste, Zeitquanten: Prio 4: T, Prio 3: 2T, Prio 2: 4T, Prio 1: 8T)

***Quelle:** [BS15]*

Abbildungsverzeichnis

- Alle Abbildungen, sofern nicht anders angegeben aus [MB17]

Literatur

- [BS17] Betriebssysteme – Grundlagen und Konzepte, Rüdiger Brause, 4. Auflage
  Springer Vieweg Verlag, 2017
  ISBN: 978-3-662-54099-2

- [GB14] Grundkurs Betriebssysteme, Peter Mandl, 4. Auflage
  Springer Vieweg Verlag, 2014
  ISBN: 978-3-658-06217-0

- [BK17] Betriebssysteme Kompakt, Christian Baun, 1. Auflage
  Springer Vieweg Verlag, 2017
  ISBN: 978-3-662-53142-6

Literatur

- [MB17] Moderne Betriebssysteme, Andrew S. Tanenbaum & Herbert Bos, 4. Auflage
  Pearson Studium, 2017
  ISBN: 978-3-86894-270-5

- [MS12] Multicore-Software, Urs Gleim & Tobias Schüle
  dpunkt.verlag, 2012
  ISBN: 978-3-89864-758-8

- [BS15] Betriebssysteme, Eduard Glatz, 3. Auflage
  dpunkt.verlag, 2015
  ISBN: 978-3-86490-222-2