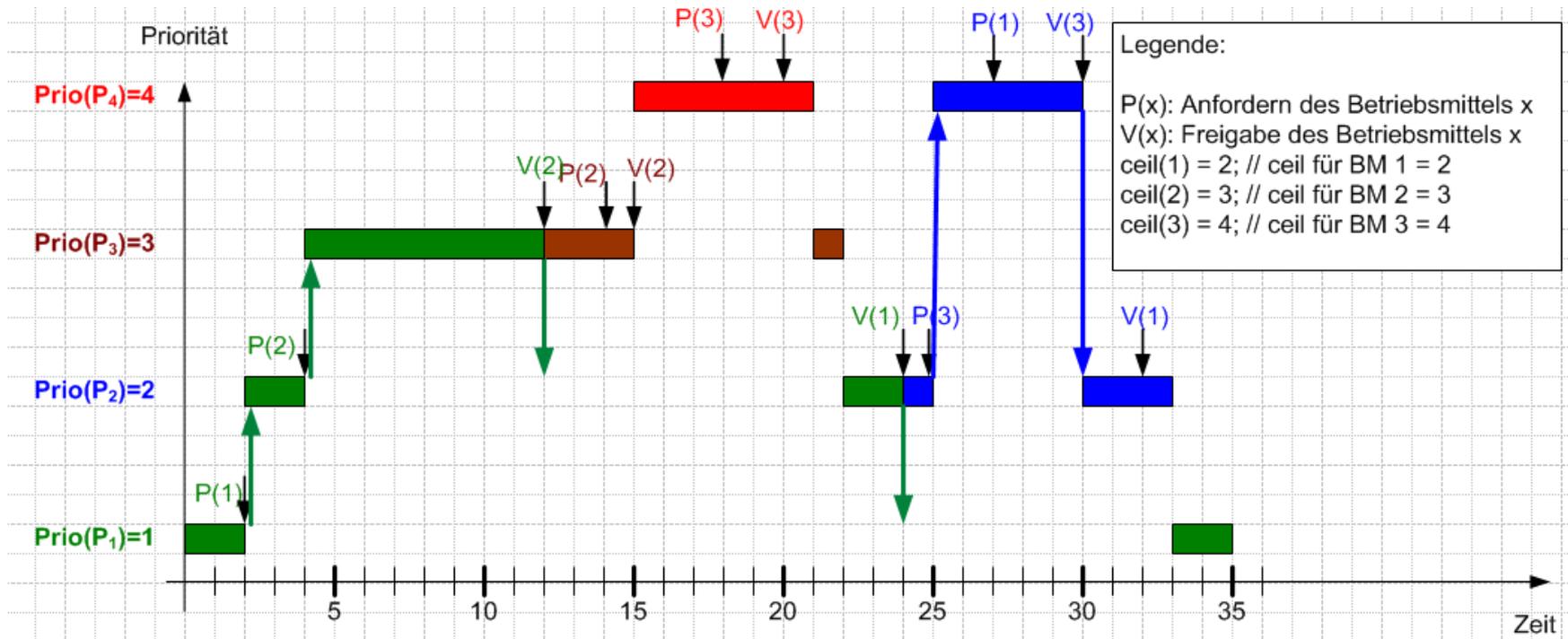


Immediate Priority Ceiling

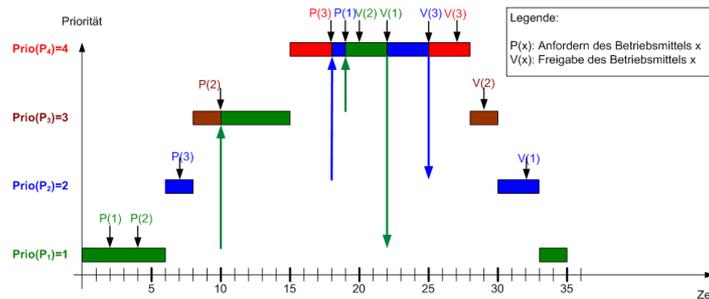
- Vereinfachtes Protokoll: **Immediate priority ceiling**: Prozesse, die ein Betriebsmittel s belegen, bekommen sofort die Priorität $ceil(s)$ zugewiesen.
- Anwendungsgebiet:
 - Verwendung vor allem bei Systemen, in denen sich mehrere Jobs einen gemeinsamen Stack teilen (→ Protokoll wird häufig auch Stack Based Priority Ceiling genannt)
 - Stack wird nach LIFO-Prinzip verwaltet, d.h. ein Job muss sich erst beenden und den Speicherplatz auf den Stack freigeben, bevor ein vorherig unterbrochener Job seine Ausführung fortsetzen kann
 - Notwendigkeit, dass alle benötigten Betriebsmittel frei sind, da es sonst zu Verklemmungen kommen kann

Beispiel: Immediate Priority Ceiling

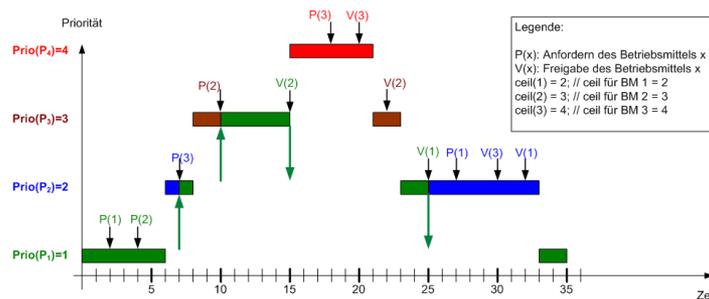


Vergleich der verschiedenen Lösungen

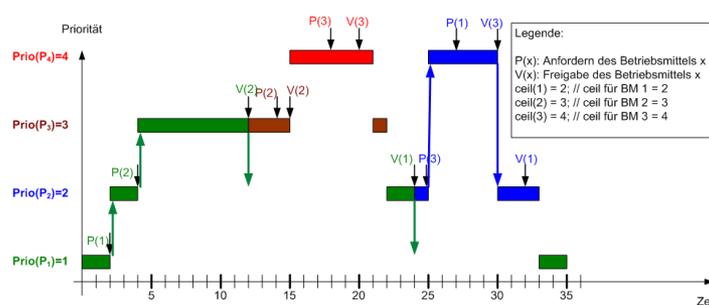
Prioritätsvererbung



Prioritätsobergrenzen

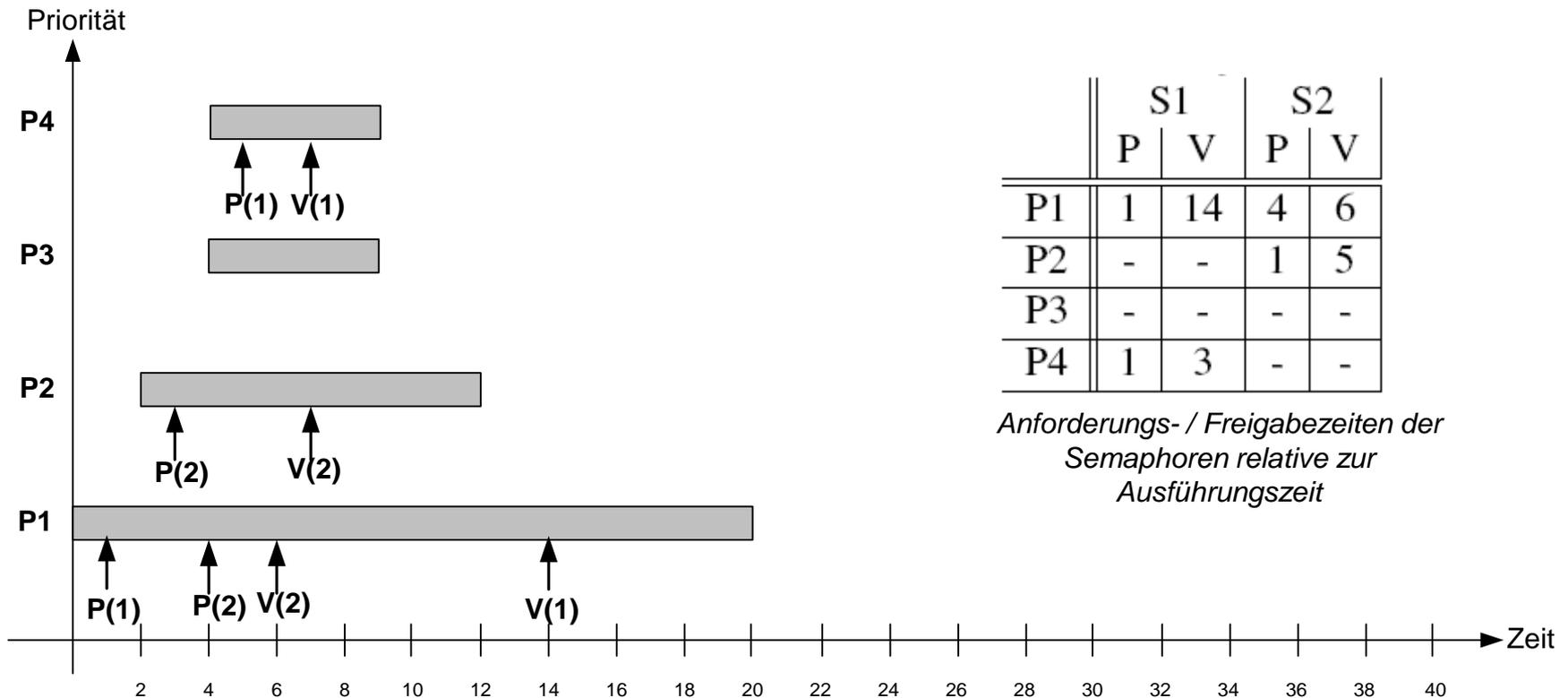


Immediate Priority Ceiling



- Prioritätsvererbung ist sehr einfach und insbesondere Anwendungsunabhängig zu implementieren
- Prioritätsobergrenzen können die Verzögerungen der höchstpriorären Prozesse häufig reduzieren, sind jedoch wesentlich aufwändiger zu implementieren und benötigen insbesondere eine Analyse aller Prozesse
- Wesentlicher Vorteil von Prioritätsobergrenzen: Blockierungsketten und damit auch Verklemmungen werden ausgeschlossen

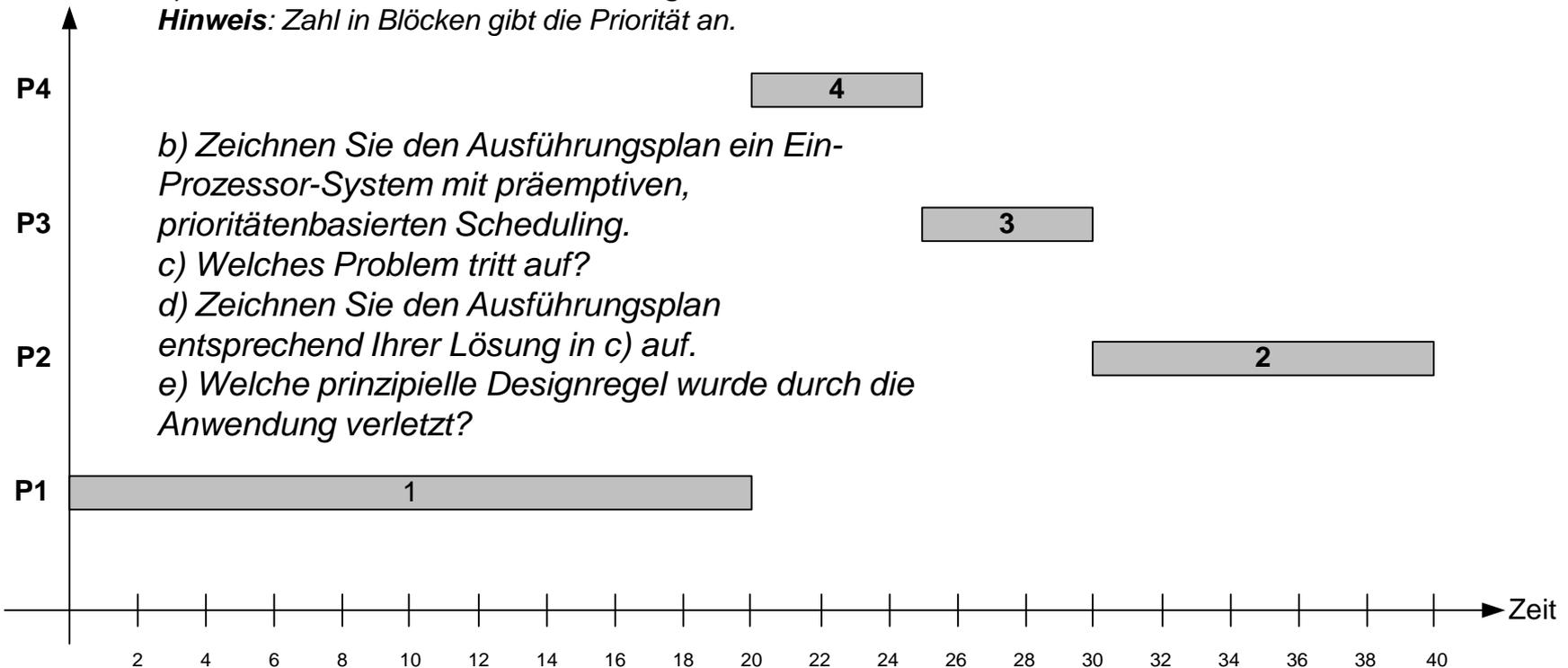
Klausur WS 06/07 – Szenario (15 Punkte = 15min)



Fortsetzung – Möglicher Ausführungsplan für ein 1-Prozessor-System

a) Welches Verfahren wird hier angewandt?

Hinweis: Zahl in Blöcken gibt die Priorität an.



b) Zeichnen Sie den Ausführungsplan ein Ein-Prozessor-System mit präemptiven, prioritätenbasierten Scheduling.

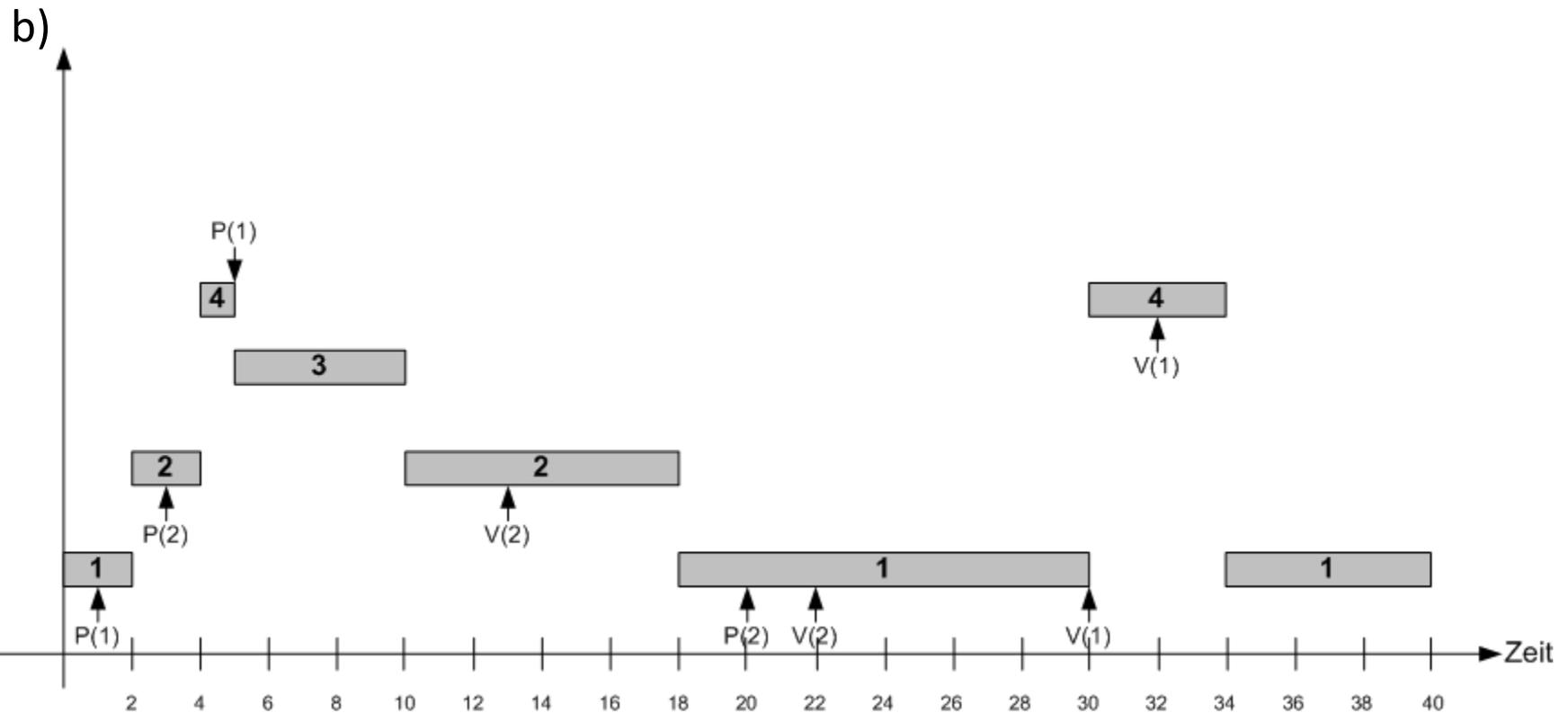
c) Welches Problem tritt auf?

d) Zeichnen Sie den Ausführungsplan entsprechend Ihrer Lösung in c) auf.

e) Welche prinzipielle Designregel wurde durch die Anwendung verletzt?

Lösung Teilaufgabe a) + b)

a) Nicht-präemptives, prioritätenbasiertes Scheduling

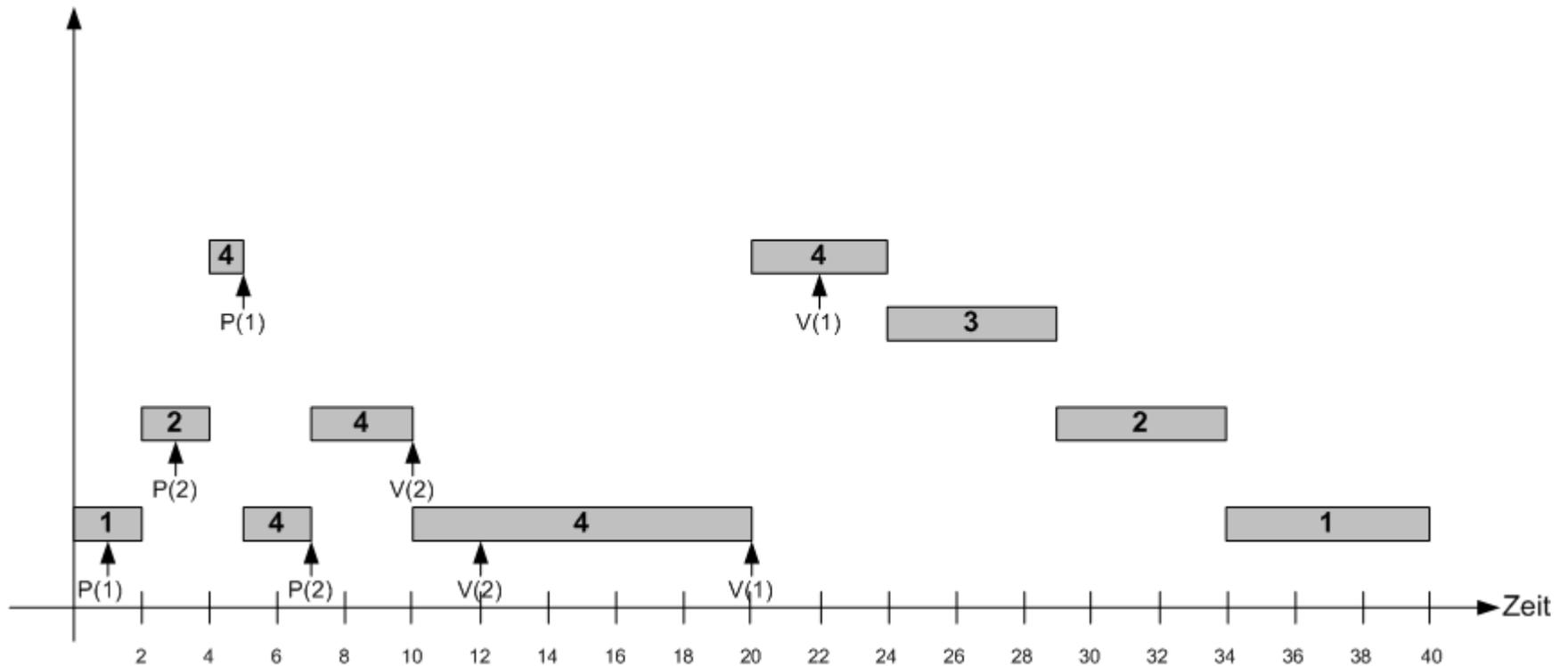


Echtzeitsysteme

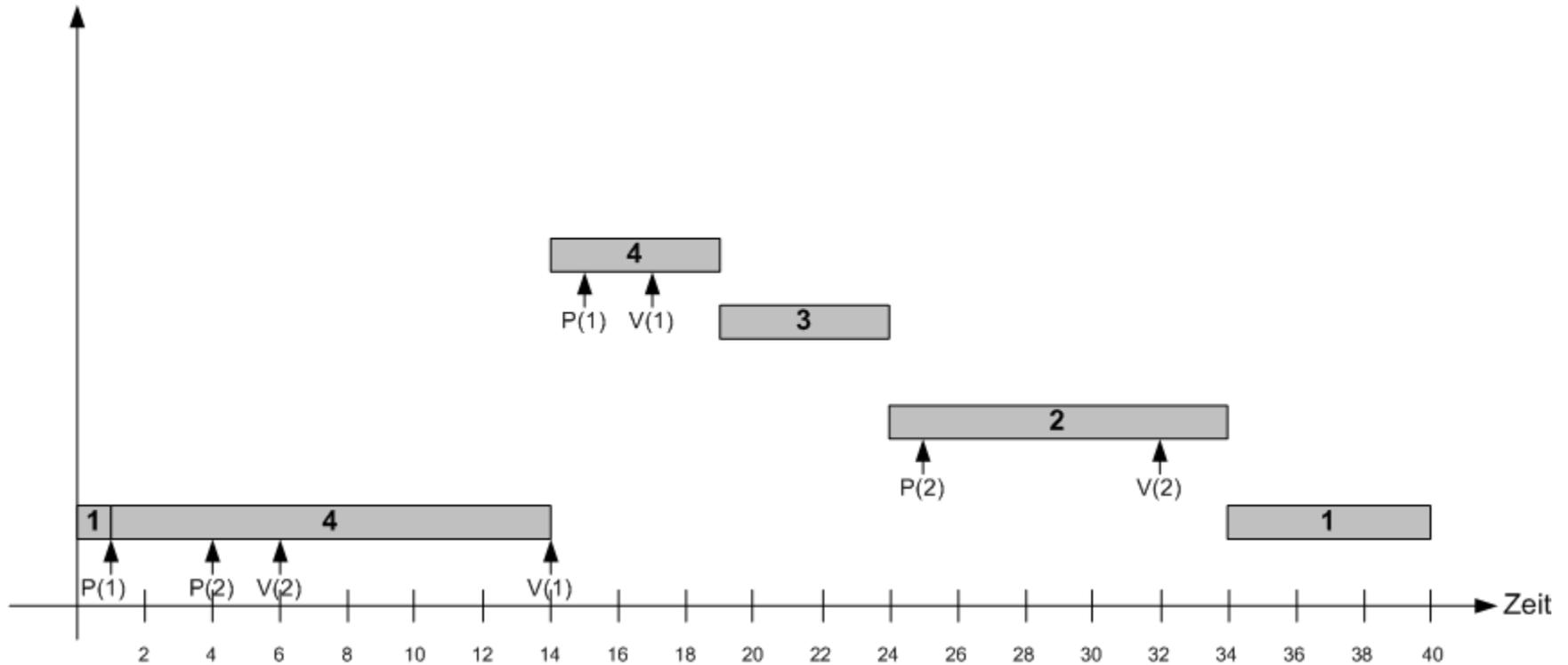
Lösung

- c) Prioritätsinversion, Möglichkeiten zur Behebung Prioritätsvererbung, (sofortige) Prioritätsgrenzen
- d) Siehe folgende Folien
- e) Prozesse mit geringerer Priorität sollten keine Betriebsmittel so lange blockieren, wenn diese auch von Prozessen mit höherer Priorität benötigt werden.

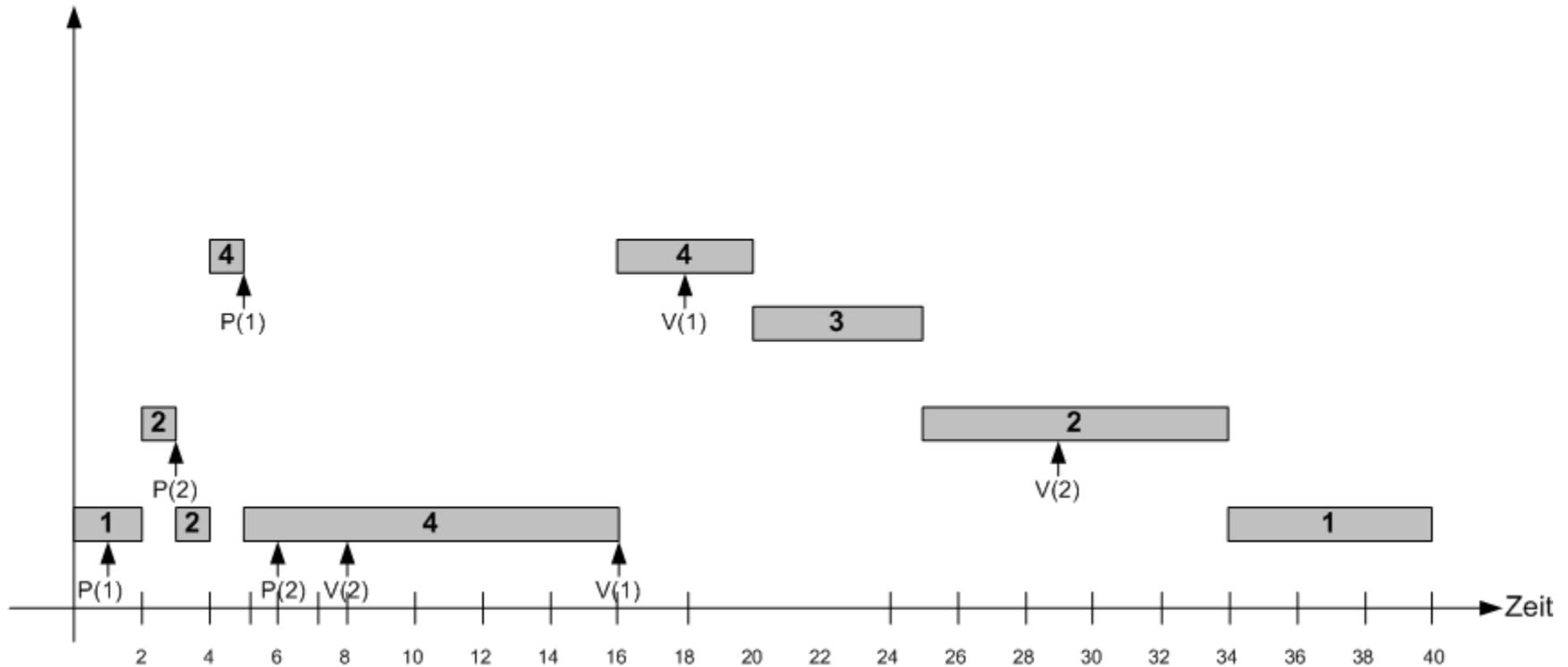
Lösung Teilaufgabe d) mit Prioritätsvererbung



Lösung Teilaufgabe d) mit sofortigen Prioritätsgrenzen



Lösung Teilaufgabe d) mit Prioritätsgrenzen



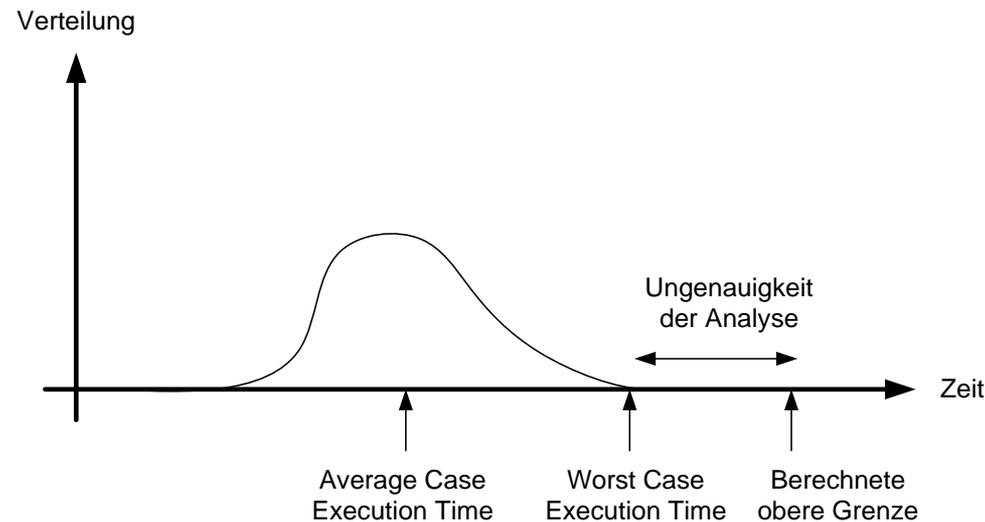


Scheduling

Exkurs: WCET (Worst Case Execution Time) - Analyse

WCET Analyse

- Ziel der Worst Case Execution Time Analyse ist die Abschätzung der maximalen Ausführungszeit einer Funktion



- Die Laufzeit ist abhängig von den Eingabedaten, dem Prozessorzustand, der Hardware,...

Probleme bei der WCET Analyse

- Bei der Abschätzung der maximalen Ausführungszeiten stößt man auf einige Probleme:
 - Es müssen unter anderem die Auswirkungen der Hardwarearchitektur, des Compilers und des Betriebssystems untersucht werden. Dadurch erschwert sich eine Vorhersage.
 - Zudem dienen viele Eigenschaften der Beschleunigung des allgemeinen Verhaltens, jedoch nicht des Verhaltens im schlechtesten Fall, z.B.:

- Caches, Pipelines, Virtual Memory
- Interruptbehandlung, Präemption
- Compileroptimierungen
- Rekursion

	Zugriffszeit	Größe
Register	0.25 ns	500 bytes
Cache	1 ns	64 KB
Hauptspeicher	100 ns	512 MB
Festplatte	5 ms	100 GB

Zugriffszeiten für verschiedene Speicherarten

- Noch schwieriger wird die Abschätzung falls der betrachtete Prozess von der Umgebung abhängig ist.

Unterscheidungen bei der WCET-Analyse

- Die Analyse muss auf unterschiedlichen Ebenen erfolgen:
 - Was macht das Programm?
 - Was passiert im Prozessor?
- Bei der Analyse werden zwei Methoden unterschieden:
 - **statische** WCET Analyse: Untersuchung des Programmcodes
 - **dynamische** Analyse: Bestimmung der Ausführungszeit anhand von verschiedenen repräsentativen Durchläufen

Statische Analyse

- **Aufgaben:**
 - Bestimmung von Ausführungspfaden in der Hochsprache
 - Transformation der Pfade in Maschinencode
 - Bestimmung der Laufzeit einzelner Maschinencodesequenzen
- **Probleme:**
 - Ausführungspfade lassen sich oft schlecht vollautomatisch ableiten (zu pessimistisch, zu komplex)
 - Ausführungspfade häufig abhängig von Eingabedaten
- **Lösungsansatz: Annotierung der Pfade mit Beschränkungen (wie z.B. maximale Schleifendurchläufe)**
 - Abhängigkeiten zwischen verschiedenen Prozessen werden häufig nicht berücksichtigt oder es werden hierzu strikte Annahmen getroffen (Beispiel hochkritische Software in der Avionik: zeitgesteuertes Scheduling und Löschen des Caches zu Beginn jedes Zeitslots)

Dynamische Analyse

- Statische Analysen können zumeist die folgenden Wechselwirkungen nicht berücksichtigen:
 - Wechselwirkungen mit anderen Programmen (siehe z.B. wechselseitiger Ausschluss)
 - Wechselwirkungen mit dem Betriebssystem (siehe z.B. Caches)
 - Wechselwirkungen mit der Umgebung (siehe z.B. Interrupts)
 - Wechselwirkungen mit anderen Rechnern (siehe z.B. Synchronisation)
- Durch dynamische Analysen können diese Wechselwirkungen abgeschätzt werden.
- Problem: Wie können die Testläufe sinnvoll ausgewählt werden?

Dimensionierung der Rechenleistungen

- Aufstellen der Worst-Case Analyse:
 - Rechenaufwand für bekannte periodische Anforderungen
 - Rechenaufwand für erwartete sporadische Anforderungen
 - Zuschlag von 100% oder mehr zum Abfangen von Lastspitzen
- Unterschied zu konventionellen Systemen:
 - keine maximale Auslastung des Prozessors
 - keine Durchsatzoptimierung
 - Abläufe sollen determiniert abschätzbar sein



Scheduling

Zusammenfassung

Erfolgskontrolle: Was Sie aus dem Kapitel mitgenommen haben?

- Kenntniss der Schedulingkriterien (Einhalten von Fristen, Fairness,...) , sowie der verschiedenen Prozessparameter (Startzeit, Laufzeit, Deadline, Priorität).
- Klassische Verfahren (EDF, LST, RM) und Anforderungen an die Optimalität dieser Verfahren
- Zeitgesteuertes Scheduling / Planen hat den Vorteil, dass es robuster gegenüber Einzelfehlern ist
- Bei prioritätsbasierten Verfahren (sowohl EDF, LST, Prioritäten) wird es umso schwieriger Aussagen zu machen, desto niedriger die Priorität ist
- Problematik der Abhängigkeiten zwischen Prozessen
- Problem der Prioritätsinversion, sowie Lösungsverfahren
- Problematik der WCET-Analyse