

Probelösung I: Betriebssysteme 2025

Bearbeitungszeit 50 Minuten. Ohne Hilfsmittel.

Von 54 möglichen Punkten werden maximal 50 gewertet.

1. Aufgabe: Richtig oder Falsch

8 Pkt

Sind die folgenden Aussagen richtig oder falsch?

Der Betriebssystemkern arbeitet immer im physischen Adressraum.

Richtig *Der Kern nutzt die MMU auch für sich, wenn es eine gibt.* Falsch

Plugins sind statische Bibliotheken, die eine Instanz zur Laufzeit explizit nachlädt.

Richtig *Dynamische. Statische Bib. sind Teil der ausführbaren Datei.* Falsch

In einem Mehrprozessorsystem kann ein Prozess auf verschiedenen Prozessoren ablaufen, aber nicht gleichzeitig auf mehreren.

Richtig Falsch

Direkt nach dem Kerneintritt werden die Rechen- und die Ablaufumgebung des laufenden Prozesses gesichert.

Richtig *Nur die Rechenumgebung wird gesichert. Die Ablaufumgebung konnte der Prozess nicht ändern.* Falsch

Nicht alle Prozesszustände des 5 State Process Model haben eine direkte Entsprechung im Modell aus der Vorlesung.

Richtig *Die Recherche zum 5SPM ist klausurrelevant.* Falsch

In der V-Aufgabe „Bereitmenge und Aufgreifstrategie“ aus der Vorlesung wird die Bereitmenge als *Multilevel Feedback Queue* verwaltet.

Richtig *Die Aufgabe verwendet eine Multilevel Queue ohne Feedback. Die Recherche zur Multilevel Feedback Queue ist klausurrelevant.* Falsch

Ein symbolischer Link in einem Dateisystem referenziert sein Ziel über eine absolute oder relative Pfadangabe.

Richtig *Die Recherche zu Links ist klausurrelevant.* Falsch

Jedes Objekt in Java darf entweder für `synchronized` oder für `wait` und `notify` eingesetzt werden, aber nicht für beides.

Richtig *Die Recherche zur Synchronisation in Java ist klausurrelevant.* Falsch

2. Aufgabe: Infrastruktur und Hauptspeicher

12 P_{KT}

Diese Teilaufgaben repräsentieren Wissensabfragen allgemein.

- a) Welche Informationen braucht der Kern mindestens, um einen neuen Prozess in einem existierenden Adressraum zu starten?

2 P_{KT}

-
- Adressraum, in dem der Befehlsstrom ablaufen soll
 - Einsprungadresse des Befehlsstroms, d.h. Adresse des ersten Befehls
 - Initialer Stapelzeiger, d.h. oberes Ende des Stapels

Bei bequemeren APIs kann der Stapel automatisch angefordert werden, bevor der Prozess startet. Auch dann kennt der Kern den Stapelzeiger.

- b) Beschreiben Sie den Aufruf einer Kernoperation aus Sicht der Instanz.

3 P_{KT}

Ein Prozess^{0.5} bzw. Befehlsstrom^{0.5} der Instanz bereitet die Parameter^{0.5} des Aufrufs vor. Dann löst er durch einen Software-Interrupt^{0.5} den Kerneintritt^{0.5} aus. Dazu gibt es einen speziellen Prozessor-Befehl.^{0.5}

Wenn der Befehlsstrom mit dem nächsten Befehl fortsetzt, ist die Kernoperation bereits beendet^{0.5} und die Ergebnisse liegen vor.^{0.5} Der Befehlsstrom muss als nächstes prüfen, ob die Kernoperation erfolgreich war oder ein Fehler auftrat.^{0.5}

Höchstens 3 Punkte erreichbar. Auch andere gute Erklärungen können zählen.

- c) Aus welchen Attributen welcher Kernobjekte stammen die Informationen für die Einträge in einer Übersetzungstabelle?

2 P_{KT}

Die logische Seitennummer leitet sich aus der Basisadresse einer *Einblendung* ab, die physische aus der Lage der Seiten des *Segments*. Die Zugriffsflags stammen aus Flags von beiden Kernobjekten.

- d) Welche Indikatoren setzt die MMU in Einträgen der Übersetzungstabellen?

1 P_{KT}

Zugriffsindikator

Modifikationsindikator

- e) *Übersetzungstabelle, für Aufgabenstellung siehe Klausur*

Logisch	Physisch	Flags
0x10	0x80	C L
0x11	0x81	D L
0x12	0x82	D S
0x13	0x83	D S
0xfd	0x84	D S
0xfe	0x85	D S
0x20	0x86	D S
0x21	0x87	D S

0.5 Punkte pro Zeile

-1 für Adressen statt Seitennummern

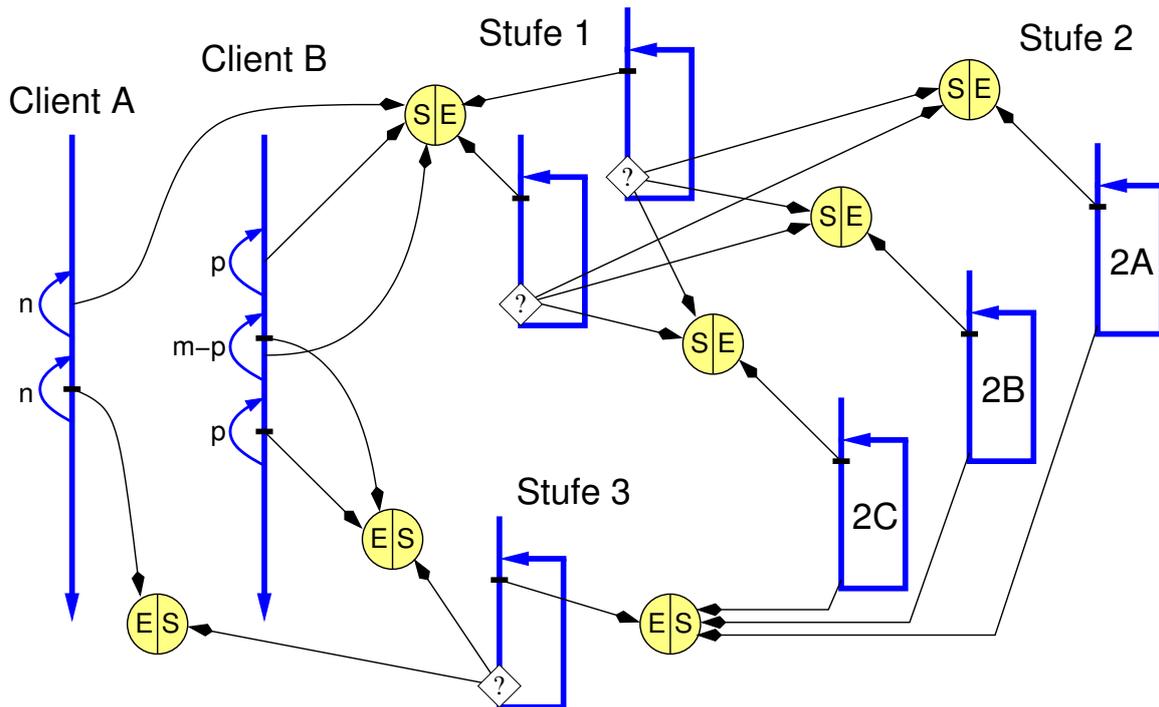
3. Aufgabe: Interaktionsdiagramm

8 PKT

Erstellen Sie ein Interaktionsdiagramm nach der folgenden Beschreibung:

Zwei Clients schicken mehrere Aufträge an ein offenes, dreistufiges Fließband. Client A schickt n Aufträge mit unbeschränkter Pufferung, Client B m Aufträge mit beschränkter Pufferung und Limit p .

Die erste Stufe des Fließbands ist mit insgesamt zwei Prozessen reproduziert. Die zweite Stufe besteht aus einem Team ohne Verteiler, das drei Arten von Aufträgen bearbeitet. Um jede Art von Auftrag kümmert sich genau ein Prozess. Die dritte Stufe ist ein einzelner Prozess.

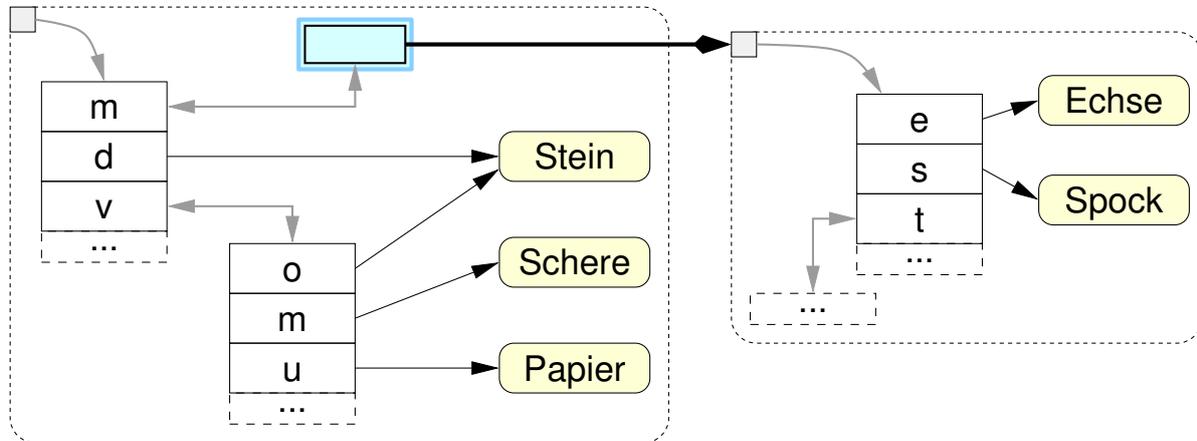


Punkte: Pufferung unbeschränkt 1, beschränkt 2, Fließband 1.5,
Repro 1, Team 1.5, ohne Verteiler 1

4. Aufgabe: Dateien und Verzeichnisse

8 PKT

Die folgende Abbildung zeigt zwei Dateisysteme mit insgesamt fünf Dateien. Das zweite ist beim ersten mit dem Pfad `/m` eingehängt. Die Dateiinhalte bestehen jeweils aus einem Wort: Stein, Schere, Papier, Echse, Spock.



- a) Welcher absolute Pfad führt zur Datei, in der „Spock“ steht? 1 PKT
`/m/s`
- b) Welche absoluten Pfade führen zur Datei, in der „Stein“ steht? 1 PKT
`/d` und `/v/o`
- c) Welche relativen Pfade führen aus dem Verzeichnis `v` zu „Stein“? 1 PKT
`o` und `../d`
- d) Welcher relative Pfad führt aus `v` zu „Echse“? 1 PKT
`../m/e`
- e) Welcher relative Pfad führt aus dem Verzeichnis `t` zu „Schere“? 1 PKT
`../..v/m`
- f) Warum legt man Mount Points bevorzugt auf leere Verzeichnisse? 1 PKT
 Ein Mount Point verdeckt den Inhalt des Verzeichnisses, auf dem er liegt.
- g) Wie können symbolische Links brechen? 2 PKT

1. Eine Komponente des verlinkten Pfads kann verschwinden.
2. Ein verlinkter relativer Pfad kann durch Verschieben des Links brechen.

5. Aufgabe: Interaktion — Yin–Yang–Sperre

18 Pkt

Ein Kernobjekt vom Typ „Yin–Yang–Sperre“ unterscheidet zwei Arten von Aufrufern, nämlich Yin– und Yang–Prozesse. Der gesperrte Bereich kann nur paarweise, also gemeinsam von einem Yin– und einem Yang–Prozess, betreten werden. Beim Verlassen gilt diese Einschränkung nicht, jeder Prozess geht alleine hinaus. Die Sperre erlaubt nur eine bestimmte Anzahl von Prozesspaaren gleichzeitig im gesperrten Bereich. Ist diese Grenze erreicht, darf ein neues Paar erst hinein, nachdem ein Yin– und ein beliebiger Yang–Prozess den gesperrten Bereich verlassen haben. Das Entsperren erfolgt also unabhängig von der Paarung beim Sperren. Der Kernobjekttyp bietet dazu folgende Operationen:

enterYinSyn: ein Yin–Prozess betritt den gesperrten Bereich

enterYangSyn: ein Yang–Prozess betritt den gesperrten Bereich

leaveYinAsyn: ein Yin–Prozess verlässt den gesperrten Bereich

leaveYangAsyn: ein Yang–Prozess verlässt den gesperrten Bereich

setCapacity: legt die erlaubte Anzahl von Paaren im gesperrten Bereich fest

Die Implementierung der Yin–Yang–Sperre verwendet folgende Attribute:

Z: aktuelle Anzahl der Paare im gesperrten Bereich, initial 0

K: erlaubte Anzahl der Paare im gesperrten Bereich, initial 1

Y: Anzahl und Art der Prozesse ohne Partner, die sich im gesperrten Bereich befinden. Initial 0, positiv bei Überzahl der Yin–Prozesse, negativ bei Überzahl der Yang–Prozesse.

WM/Yin: Wartemenge für Yin–Prozesse, initial leer

WM/Yang: Wartemenge für Yang–Prozesse, initial leer

Wenn ein einzelner Prozess den gesperrten Bereich verlässt, bleibt der Zähler Z unverändert. Erst wenn ein Prozess der anderen Art ebenfalls den gesperrten Bereich verlässt (bzw. vorher verlassen hat) und somit ein Paar komplettiert, wird ein anderes Paar eingelassen oder Z vermindert. Der Hilfszähler Y führt Buch über die alleingelassenen Prozesse im gesperrten Bereich, deren Partner schon gegangen ist.

Lösungen auf der nächsten Seite

- a) Beschreiben Sie den Ablauf von `enterYinSyn`. Der Aufrufer muss warten, bis ein Partnerprozess bereitsteht und Platz im gesperrten Bereich frei ist. 5 Pkt
- WM/Yang nicht leer und $Z < K$, d.h. Paar darf eintreten? +2
 - Ja:
 - Prozess aus WM/Yang nehmen und deblockieren +1
 - Z um 1 erhöhen +1
 - Nein: Aufrufer blockieren und in WM/Yin legen +1
- b) Beschreiben Sie den Ablauf von `leaveYinAsyn`. Der Aufrufer muss nicht warten. Y vermindert sich und zeigt an, ob der gehende Yin-Prozess ein Paar komplettiert oder nicht. Bedenken Sie, dass K durch einen Aufruf von `setCapacity` vermindert sein könnte. 7 Pkt
- Y um 1 vermindern +1
 - falls $Y \geq 0$, d.h. komplettes Paar ist raus: +1
 - $Z \leq K$ und WM/Yin nicht leer und WM/Yang nicht leer? +2
 - Ja: *ein Paar wartet und darf den Platz übernehmen*
 - * Prozess aus WM/Yin nehmen und deblockieren +1
 - * Prozess aus WM/Yang nehmen und deblockieren +1
 - Nein: *kein Paar wartet, oder Schranke noch überschritten*
 - * Z um 1 vermindern +1
- c) Beschreiben Sie den Ablauf von `setCapacity`. Der neue Wert von K wird sofort übernommen, auch wenn sich gerade mehr Prozesspaare im gesperrten Bereich befinden. Weitere Paare dürfen aber erst wieder eintreten, wenn die Schranke dabei eingehalten wird. 6 Pkt
- K auf neuen Wert setzen +1
 - solange $Z < K$ und WM/Yin nicht leer und WM/Yang nicht leer +2
 - Z um 1 erhöhen +1
 - Prozess aus WM/Yin nehmen und deblockieren +1
 - Prozess aus WM/Yang nehmen und deblockieren +1