

Probeklausur 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?

Achtung: Falsche Antworten führen zu Punktabzug, fehlende nicht.

Der Betriebssystemkern arbeitet immer im physischen Adressraum.

☐ Richtig

Falsch ☐

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

☐ Richtig

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

Falsch ☐

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

☐ Richtig

Falsch ☐

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

☐ Richtig

Falsch ☐

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

☐ Richtig

Falsch ☐

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

☐ Richtig

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}

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

3 P_{KT}

bei Bedarf auf einem anderem Blatt weiterschreiben

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

2 P_{KT}

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

1 P_{KT}

- e) Der Programmlader startet ein bestimmtes Programm. Dazu legt er einen neuen Adressraum an und stellt darin folgende Bereiche bereit:

Zweck	Anfang	Größe
Code	0x10000	0x1000
Konstanten	0x11000	0x1000
Statische Daten	0x12000	0x2000
Stapel	0xfd000	0x2000
Halde	0x20000	0x2000

Die Seitengröße für die Adressübersetzung ist 0x1000 (4096 Byte). Die Bereiche werden in der oben angeführten Reihenfolge angefordert, die Seiten innerhalb eines Bereichs nach aufsteigenden Adressen. Alle Seiten liegen im Hauptspeicher. Dort sind genügend Seiten ab Adresse 0x80000 frei. Sie werden ebenfalls nach aufsteigenden Adressen belegt.

Erstellen Sie eine Übersetzungstabelle für die Instanz, unmittelbar nachdem der Programmlader den Adressraum eingerichtet hat. In der Tabelle erscheint nur der umzurechnende Teil der Adressen. Ordnen Sie jedem Eintrag die passenden Gültigkeitsflags zu.

4 PKT

Flags: C = Code, D = Daten, L = nur Lesen, S = Lesen und Schreiben

Logische Seitennummer	Physische Seitennummer	Flags

Diese Tabelle enthält mehr Zeilen, als die Antwort benötigt.

3. Aufgabe: Interaktionsdiagramm

8 PKT

Erstellen Sie ein Interaktionsdiagramm nach der folgenden Beschreibung:

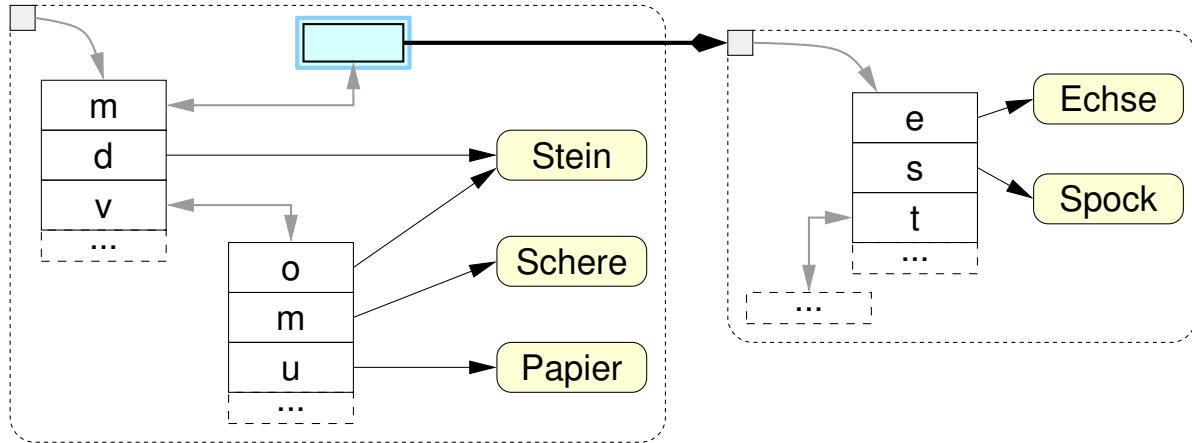
Zwei Clients schicken mehrere Aufträge an ein offenes, dreistufiges Fließband. Client A schickt n Aufträge mit unbeschränkter Pufferung, Client B m Aufträge mit beschränkter Pufferung und Limit p .

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.

4. Aufgabe: Dateien und Verzeichnisse

8 P_{KT}

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. *Geben Sie Pfade ohne Umwege an, also ohne überflüssige `../` oder `./`*



- Welcher absolute Pfad führt zur Datei, in der „Spock“ steht? 1 P_{KT}

- Welche absoluten Pfade führen zur Datei, in der „Stein“ steht? 1 P_{KT}

- Welche relativen Pfade führen aus dem Verzeichnis `v` zu „Stein“? 1 P_{KT}

- Welcher relative Pfad führt aus `v` zu „Echse“? 1 P_{KT}

- Welcher relative Pfad führt aus dem Verzeichnis `t` zu „Schere“? 1 P_{KT}

- Warum legt man Mount Points bevorzugt auf leere Verzeichnisse? 1 P_{KT}

- Wie können symbolische Links brechen? 2 P_{KT}

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 Kernobjektyp 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.

- a) Beschreiben Sie den Ablauf von `enterYinSyn`. Der Aufrufer muss warten, bis ein Partnerprozess bereitsteht und Platz im gesperrten Bereich frei ist.

5 P_{KT}

Hinweis: `enterYangSyn` läuft analog mit jeweils anderer Wartemenge ab.

- 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 P_{KT}

Hinweis: `leaveYangAsyn` läuft analog ab, wobei sich Y erhöht.

- 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
