
Prüfungsteilnehmer

Prüfungstermin

Einzelprüfungsnummer

Kennzahl: _____

Kennwort: _____

Arbeitsplatz-Nr.: _____

**Herbst
2015**

66114

**Erste Staatsprüfung für ein Lehramt an öffentlichen Schulen
— Prüfungsaufgaben —**

Fach: **Informatik (vertieft studiert)**

Einzelprüfung: **Datenbank- und Betriebssysteme**

Anzahl der gestellten Themen (Aufgaben): **2**

Anzahl der Druckseiten dieser Vorlage: **12**

Zu den zwei Themenschwerpunkten A (Datenbanksysteme) und B (Betriebssysteme) ist jeweils entweder die Teilaufgabe 1 oder 2 zu wählen!
Auf der Vorderseite des Kopfbogens sind im Feld „Gewähltes Thema: Nr.“ die Nummern der beiden gewählten Teilaufgaben anzugeben (z. B. A2, B1)!

Bitte wenden!

Themenschwerpunkt A (Datenbanksysteme)

Teilaufgabe 1

1. Modellierung

Wir wollen eine relationale Datenbankstruktur für ein Online-Auktionshaus modellieren.

Das Auktionshaus hat Mitglieder. Diese Mitglieder haben Kundennummern, Namen und Adressen. Sie können Verkäufer und/oder Käufer sein. Verkäufer können eine Laden-URL (eine Subdomain) erhalten, die zu einer Seite mit ihren aktuellen Auktionen führt. Verkäufer können neue Auktionen starten. Die Auktionen eines Verkäufers werden durchnummeriert. Diese Auktionsnummer ist nur je Verkäufer eindeutig. Jede Auktion hat ein Mindestgebot und eine Ablaufzeit.

Auf Auktionen können Gebote abgegeben werden. Die Gebote auf eine Auktion werden nach ihrem Eintreffen nummeriert. Diese Nummer ist nur innerhalb einer Auktion eindeutig. Zu einem Gebot werden noch die Zeit des Gebots sowie der gebotene Geldbetrag angegeben. Die Gebote werden von Käufern abgegeben. Jedes Gebot muss einem Käufer zugeordnet sein.

Jede Auktion besteht aus einer Menge von Artikeln, aber aus mindestens einem. Artikel haben eine Beschreibung und können über ihre Artikel-ID identifiziert werden. Zur Katalogisierung der Artikel gibt es Kategorien. Kategorien haben eine eindeutige ID und einen Namen. Jede Kategorie kann Subkategorien besitzen. Auch Subkategorien sind Kategorien (und können damit weitere Subkategorien besitzen). Jeder Artikel kann beliebig vielen Kategorien zugeordnet werden.

Entwerfen Sie für das beschriebene Szenario ein ER-Diagramm. Bestimmen Sie hierzu:

- die Entity-Typen, die Relationship-Typen und jeweils deren Attribute,
- ein passendes ER-Diagramm,
- die Primärschlüssel der Entity-Typen, welche Sie anschließend in das ER-Diagramm eintragen,
- die Funktionalitäten der Relationship-Typen, welche Sie ebenfalls in das ER-Diagramm eintragen.

Fortsetzung nächste Seite!

2. Normalformen

Gegeben sei folgendes verallgemeinerte Relationenschema in 1. Normalform:

$$R: \{[A, B, C, D, E, F, G, H]\}$$

Für R soll die folgende Menge FD von funktionalen Abhängigkeiten gelten:

- $F \rightarrow E$
- $A \rightarrow BD$
- $AE \rightarrow D$
- $A \rightarrow EF$
- $AG \rightarrow H$

Bearbeiten Sie mit diesen Informationen folgende Teilaufgaben. Vergessen Sie dabei nicht Ihr Vorgehen stichpunktartig zu dokumentieren und zu begründen.

- a) Bestimmen Sie alle Schlüsselkandidaten von R. Begründen Sie stichpunktartig, warum es außer den von Ihnen gefundenen Schlüsselkandidaten keine weiteren geben kann.
- b) Ist R in 2NF, 3NF?
- c) Berechnen Sie eine kanonische Überdeckung von FD. Es genügt, wenn Sie für jeden der vier Einzelschritte die Menge der funktionalen Abhängigkeiten als Zwischenergebnis angeben.
- d) Bestimmen Sie eine Zerlegung von R in 3NF. Wenden Sie hierfür den Synthesalgorithmus an.

3. SQL

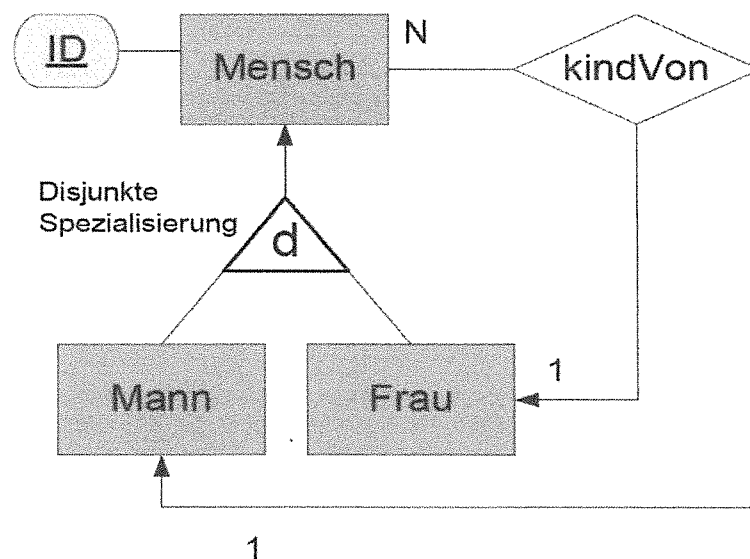
Gegeben seien folgende Relationen:

Mensch(ID, MutterID, VaterID)

Mann(ID)

Frau(ID)

Das zugehörige ER-Modell für dieses relationale Datenbankschema sieht folgendermaßen aus:



Fortsetzung nächste Seite!

Bearbeiten Sie folgende Teilaufgaben:

- a) Finden Sie die Töchter der Frau mit *ID* 42.
- b) Gibt es Männer, die ihre eigenen Großväter sind? Formulieren Sie eine geeignete SQL-Anfrage.
- c) Definieren Sie eine View *VaterKind* (*VaterID*; *KindID*), die allen Vätern (*VaterID*) ihre Kinder (*KindID*) zuordnet. Diese View darf keine NULL-Werte enthalten.
- d) Verwenden Sie die View aus c), um alle Väter zurückzugeben, absteigend geordnet nach der Anzahl ihrer Kinder.
- e) Hugo möchte mit folgender Anfrage auf Basis der View aus c) alle kinderlosen Männer erhalten:

```
SELECT VaterID
FROM VaterKind
GROUP BY VaterID
HAVING COUNT(KindID) = 0
```

- i. Was ist das Ergebnis von Hugos Anfrage und warum?
- ii. Formulieren Sie eine Anfrage, die tatsächlich alle kinderlosen Männer zurückliefert.

Hinweis: Denken Sie daran, dass SQL auch Mengenoperationen kennt.

Teilaufgabe 2

1. Modellierung

Das Fitnessstudio „Bavariogym“ plant zur Unterstützung des Trainings eine App für ihre Mitglieder. Betrachten Sie dazu folgendes Szenario:

- Zu einem Mitglied werden Vorname, Nachname, Eintrittsdatum und der Tarif gespeichert.
- Im Fitnessstudio arbeiten speziell ausgebildete Fitnesstrainer. Deren Aufgabe ist es, Trainingspläne zu erstellen. Ein Trainer besitzt immer eine eindeutige Personal-Nummer. Gehen Sie davon aus, dass ein Trainer kein Mitglied ist.
- Ein Trainingsplan ist nur in einem bestimmten Zeitraum gültig. Sehen Sie hierzu zwei Daten für die zeitlichen Grenzen vor. Um einen Trainingsplan zu identifizieren soll eine fortlaufende Nummer verwendet werden.
- Jeder Trainingsplan schreibt das Training an unterschiedlichen Geräten vor. Dabei sollen jeweils die Anzahl der Wiederholungen pro Satz, die Anzahl der Sätze und die Gewichtseinstellung berücksichtigt werden.
- Ein Mitglied kann nach mehreren Trainingsplänen gleichzeitig trainieren. Ebenso findet ein Trainingsplan bei mehreren verschiedenen Mitgliedern Anwendung.
- Ein Gerät besitzt einen eindeutigen Namen, sowie eine Information über die Muskelgruppe, die trainiert wird.

Fortsetzung nächste Seite!

- a) Entwerfen Sie für das beschriebene Szenario ein ER-Modell in Chen-Notation. Bestimmen Sie hierzu:
- Die Entity-Typen, die Relationship-Typen und jeweils deren Attribute.
 - Die Funktionalitäten der Relationship-Typen.
- b) Wie würde sich ihr Modell aus Teilaufgabe a) ändern, wenn die Trainer den Mitgliedern individuelle, jedoch nicht exklusive Trainingspläne anfertigen? Zeichnen Sie ein aktualisiertes ER-Modell in Chen-Notation. Bestimmen Sie hierzu:
- Die Entity-Typen, die Relationship-Typen, jeweils **ohne** deren Attribute.
 - Die Funktionalitäten der Relationship-Typen.
- c) Wie würden Sie vorgehen, um ihr Modell aus Teilaufgabe a) so zu ändern, dass Sie für jede Person im Fitnessstudio den Vornamen, den Nachnamen, die Straße, die Hausnummer, die PLZ und den Ort speichern können. Beschränken Sie ihre Antwort auf maximal drei Sätze.

2. SQL-Anfragen

Gegeben ist das folgende relationale Schema:

EQUIPE $\{\{id, name\}\}$

FAHRER $\{\{id, name, equipe_id\}\}$

ETAPE $\{\{id, date, NbKm\}\}$

TEILNAHME $\{\{fahrer_id, etape_id, zeit\}\}$

Zur Modellierung der Tour de France im letzten Jahr (2014) wird folgendes relationales Schema verwendet. EQUIPE enthält Informationen zur Mannschaft (der EQUIPE), zu denen jeder Fahrer eindeutig zugeordnet ist. FAHRER enthält die Informationen zum Fahrer mit dem Fremdschlüssel *equipe_id*. ETAPE enthält die Information zu den Etappen, die gefahren werden, das Attribut *NbKm* ist die Länge der Etape in km. TEILNAHME stellt die Verbindung von Etappen und Fahrern her, das Attribut *zeit* ist die individuelle Zeit eines Fahrers *fahrer_id* in der Etape *etape_id*. Fällt der Fahrer für die Etape *id* aus, so erscheint er in der Relation TEILNAHME für dieses *etape_id* nicht.

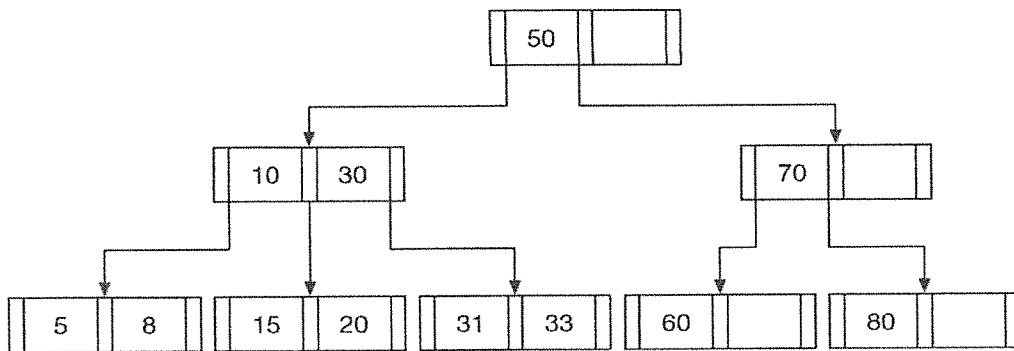
Formulieren Sie folgende Anfragen in SQL. Achten Sie falls nötig auf duplikatfreie Ausgaben:

- a) Wie lang war die Tour de France letztes Jahr?
- b) Geben Sie die Namen der Fahrer der Equipe 'RENNSTALL' aus.
- c) Geben Sie die Namen der Fahrer an, die an allen Etappen teilgenommen haben.
- d) Geben Sie die Gesamtlaufzeit aller Läufer der Equipe 'RENNSTALL' in der 1. Etape (ETAPE.id=1) aus.

Fortsetzung nächste Seite!

3. Indexstrukturen

Gegeben ist der folgende B-Baum der Ordnung 3 (max. drei Kindknoten, max. zwei Schlüssel pro Knoten):



- Fügen Sie die Werte 9 und 45 ein. Löschen Sie anschließend die Werte 30 und 70. Zeichnen Sie den Baum nach jeder Einfüge- bzw. Lösch-Operation.
- Geben Sie das Vorgehen für exakte Suche und Bereichssuche im B+-Baum an.
- Geben Sie einen Vorteil und einen Nachteil von B+-Bäume im Vergleich zu Hash-Tabellen an.

Themenschwerpunkt B (Betriebssysteme)

Teilaufgabe 1

In allen gängigen Betriebssystemen wird heute die Parallelität durch Prozesse, teils auch zusätzlich durch Threads, umgesetzt. Zur Verwaltung der Prozesse sind die Zustände wichtig, die Prozesse während ihrer Lebenszeit annehmen können; zudem sind die Wechselwirkungen wichtig, die sich in der Synchronisation und Verklemmungen niederschlagen, sowie für die Zuordnung zu den Prozessoren das Scheduling und das Dispatching. Der Prozessadressraum basiert in der Regel auf einem virtuellen Speicher, so dass das Paging eine zentrale Rolle spielt. Für die langfristige, persistente Speicherung von Daten werden Dateisysteme herangezogen.

1. Prozesse und ihre Zustände

Prozesse durchlaufen während ihrer Lebenszeit einige Zustände. Benennen Sie die wichtigsten dieser Zustände, erläutern Sie diese kurz und skizzieren Sie ein Diagramm, das die Zustandsübergänge aufzeigt. Beschreiben Sie kurz die Bedeutung der jeweiligen Zustandsübergänge.

2. Synchronisation zwischen Prozessen

Gegeben sei die folgende Festlegung mit einer Klasse für Semaphore. Nutzen Sie diese für die Lösung der nachfolgenden Teilaufgaben a) und b).

```
public class Semaphor
{
    // Konstruktor
    public Semaphor (int init) {...}

    // Methoden
    public void prolog () {...}
    public void epilog () {...}
}
```

Fortsetzung nächste Seite!

- a) Gegeben seien zwei Prozesse P1 und P2. Der Einfachheit halber sollen beide Prozesse jeweils nur aus zwei Berechnungsschritten bestehen: P1 aus S11 gefolgt von S12 und P2 aus S21 gefolgt von S22. Ein einfaches Beispiel für die Synchronisation zwischen Prozessen besteht nun darin, dass der zweite Schritt des zweiten Prozesses (S22) erst dann ausgeführt werden darf, wenn der erste Schritt des ersten Prozesses erfolgreich abgeschlossen wurde. Geben Sie eine Lösung für diese Synchronisationsaufgabe unter Verwendung von Semaphoren an. Beschreiben Sie Ihre Lösung programmiersprachlich oder in Pseudocode und treffen Sie dazu die nötigen Annahmen.
- b) Das bei den Semaphoren benötigte Warten kann entweder mit aktivem Warten oder mit passivem Warten realisiert werden. Beschreiben Sie kurz die Unterschiede zwischen diesen Varianten.

3. Scheduling von Prozessen

Gegeben sei ein Ein-Prozessor-System mit sechs Prozessen bzw. Aufträgen A_1, \dots, A_6 und den dazugehörigen Bedienzeiten $b=(4,2,5,3,1,1)$, also den Zeiten, die die jeweiligen Prozesse zur Bearbeitung benötigen. Die Ankunftszeiten der Aufträge werden in den nachfolgenden Unteraufgaben angegeben. Vor dem Start des Scheduling liege kein Prozess bzw. Auftrag vor.

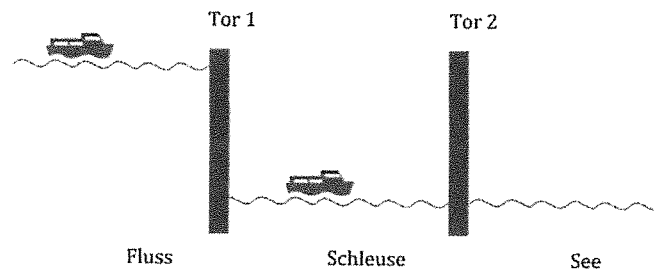
- a) Der Vektor der Ankunftszeiten für die sechs Prozesse sei durch $a=(0,1,2,3,4,5)$ gegeben. Berechnen Sie die mittlere Verweilzeit und die mittlere Wartezeit für die nichtunterbrechende Strategie Shortest Processing Time (SPT), also die Strategie, bei der in jedem Zeitintervall einer der Aufträge mit kürzester Bedienzeit ausgeführt wird. Notieren Sie den Ablauf entsprechend.
- b) Nun kommen die Prozesse (Aufträge) mit folgendem Vektor der Ankunftszeiten $a'=(5,4,3,2,1,0)$ an. Die Aufträge werden nun nach einer Zeitscheibenstrategie mit dem Quantum q bedient (Round-Robin Strategie). Die Reihenfolge wird durch die Zeitpunkte der Ankünfte festgelegt. Falls erforderlich werden Neuankünfte immer vor einem aktuellen Wiederholer eingereiht. Berechnen Sie die mittlere Verweilzeit und die mittlere Wartezeit für das Quantum $q=1/2$.

4. Virtueller Speicher, Seiteneinteilung und Paging

- a) Gegeben sei ein Prozess mit einem virtuellen Speicher von sechs Seiten, für dessen Realisierung drei Seitenrahmen (Kachelmenge = $\{K_1, K_2, K_3\}$) zur Verfügung stehen. Geben Sie für die Strategien LRU (Least Recently Used), LFU (Least Frequently Used) und FIFO (First in First out) die Entwicklung der Seitenrahmenbelegung bei der Zugriffsreferenzkette $a'=1\ 3\ 2\ 6\ 4\ 2\ 1\ 6\ 5\ 1\ 2\ 1\ 1\ 3\ 4\ 3\ 5\ 1$ an. Ist die Auswahlentscheidung nicht eindeutig, verwenden Sie zusätzlich FIFO. Die Seitenrahmen seien zu Beginn leer. Markieren Sie jedes Auftreten von Seitenfehlern und notieren Sie für die jeweilige Strategie die Gesamtzahl der Seitenfehler.
- b) Die Menge der Seitenrahmen (Kacheln), in die die Seiten des virtuellen Speichers eingelagert werden, muss durch das Betriebssystem verwaltet werden. Geben Sie eine Datenstruktur zur Verwaltung der Kacheln an und listen Sie die wichtigsten Elemente der Seitenrahmendesriptoren auf.

Teilaufgabe 2

1. Modellierung paralleler Systeme

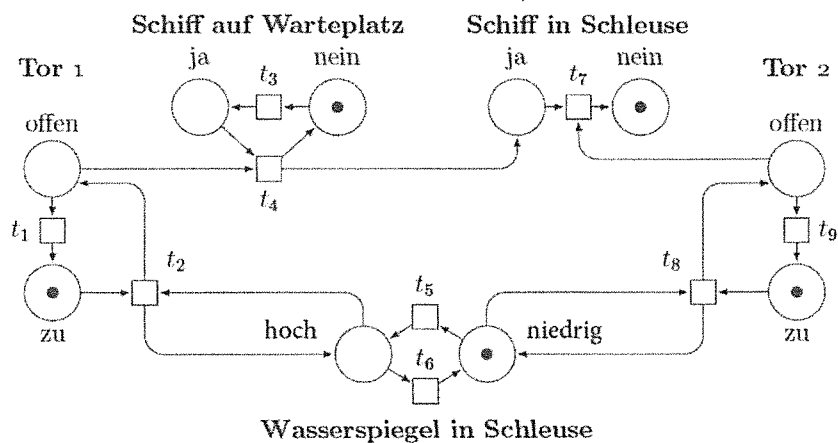


Wir betrachten eine Schleuse, die es Schiffen ermöglicht, von einem Fluss mit hohem Wasserspiegel in einen See mit niedrigem Wasserspiegel zu fahren (nur in diese Richtung). Die Schleuse hat zwei Tore. Der Wasserspiegel in der Schleuse kann entweder hoch oder niedrig sein. Im Fluss gibt es einen Wartepplatz für ein Schiff, das überfahren will, und natürlich kann sich auch in der Schleuse selbst ein Schiff befinden. Es gelten folgende zusätzliche Bedingungen:

- Tor 1 darf nur geöffnet sein, wenn der Wasserspiegel in der Schleuse hoch ist, und Tor 2 darf nur geöffnet werden, wenn der Wasserspiegel in der Schleuse niedrig ist.
- Die Schleuse muss nach Durchfahrt eines Schiffes immer in der Lage sein, ein weiteres Schiff durchfahren zu lassen.
- Zustände sind dichotom: Die Tore sind entweder offen oder zu, der Wasserspiegel entweder hoch oder niedrig. Auf dem Wartepplatz und in der Schleuse gibt es (unabhängig voneinander) ein Schiff oder nicht.

Fortsetzung nächste Seite!

Gegeben sei nun ein fehlerhafter Entwurf eines natürlichzahligen Petrinetzes zur Schaltung der Schleuse:



- | | | |
|--------------------------------|--------------------------------------|--------------------------------|
| t_1 : Tor 1 wird geschlossen | t_4 : Schiff fährt in die Schleuse | t_7 : Schiff fährt weg |
| t_2 : Tor 1 wird geöffnet | t_5 : Wasser wird hoch | t_8 : Tor 2 wird geöffnet |
| t_3 : Schiff kommt an | t_6 : Wasser wird niedrig | t_9 : Tor 2 wird geschlossen |

Korrigieren Sie das obige Petrinetz durch Einzeichnen der sieben fehlenden Pfeile im gegebenen Diagramm.

Hinweis: Gehen Sie das Netz anhand eines naheliegenden Szenarios durch. Gibt es Schaltfolgen, nach denen eine der genannten Bedingungen nicht mehr erfüllt ist?

2. Prozess- und Threadsynchronisation

Das erste Readers-Writers-Problem beschreibt ein Szenario, bei dem mehrere Prozesse – manche lesend, manche schreibend – um den Zugriff auf eine Ressource konkurrieren. Mehrere lesende Prozesse können zeitgleich auf die Ressource zugreifen, solange kein schreibender Zugriff stattfindet. Will ein Prozess schreiben, wartet er, bis andere Zugriffe abgeschlossen sind. Greift ein Prozess schreibend auf die Ressource zu, ist anderen Prozessen der Zugriff so lange verwehrt. Der nachfolgende Pseudocode modelliert beide Prozesstypen und ist so zu vervollständigen, dass Prozesse schreiben \rightarrow `doActualWriting()` respektive lesen \rightarrow `doActualReading()` können, ohne die genannten Voraussetzungen zu verletzen. Verwenden Sie hierfür ausschließlich die bereits vorgegebenen Konstrukte. Gehen Sie davon aus, dass die untenstehenden Funktionen wiederholt aufgerufen werden. Ein Aufruf löst einen Schreib- bzw. Lesevorgang aus. Verzichteten Sie daher auf Schleifen oder rekursive Funktionsaufrufe.

Fortsetzung nächste Seite!

```
Semaphore write=1;
Semaphore lock=1;

int readcount=0; //zähle Lesezugriffe

writer()
{

    //schreibe
    doActualWriting();

}

reader()
{

    //lese
    doActualReading();

}

}
```

3. Scheduling

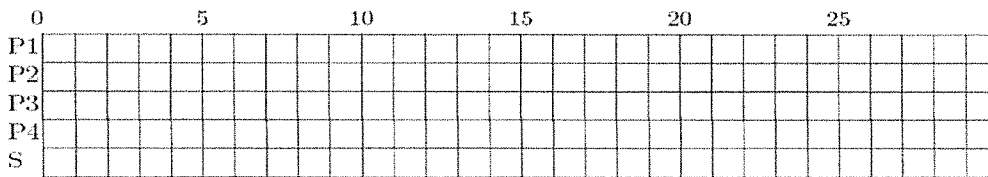
- a) Nennen Sie außer dem Prozessnamen vier Inhalte des Prozesskontrollblocks eines Prozesses.
- b) Welche Abwägung gilt es bei der Wahl des (fixen) Zeitquantums bei der Implementierung eines Round-Robin-Schedulers zu treffen?

Fortsetzung nächste Seite!

c) Gegeben seien vier Prozesse mit ihrer jeweiligen Ankunftszeit und Bedienzeit:

Prozess	Ankunftszeit	Bedienzeit in ms
P1	0	13
P2	0	4
P3	16	3
P4	17	4

Nehmen Sie an, dass jeder Kontextwechsel 1ms kostet. Bestimmen Sie die durchschnittliche Prozesswartzeit inklusive Kontextwechsel für einen Round-Robin-Scheduler mit einem Quantum von 4ms. Ist nach einem vollen Quantum ein Prozess nicht **sofort** verfügbar, so wird er nicht in Betracht gezogen – selbst dann nicht, wenn er nach dem Kontextwechsel verfügbar wäre. Bei nicht eindeutiger Entscheidbarkeit haben kleinere Prozess-IDs Vorrang. Das nachfolgende leere Gantt-Diagramm kann Sie bei der Berechnung unterstützen. Dokumentieren Sie Ihr Vorgehen.



4. Speicherverwaltung

- a) Erläutern Sie worin der Unterschied zwischen externer und interner Fragmentierung von Speicherbereichen besteht!
- b) Gegeben seien eine Menge von Seiten (pages) $N = \{0, 1, 2, 3, 4, 5, 6, 7\}$ und eine Menge von Seitenrahmen (frames) $F = \{f_1, f_2, f_3\}$. Nun wird in folgender Reihenfolge auf die Seiten zugegriffen:

$$w = 7 \ 2 \ 5 \ 0 \ 4 \ 0 \ 7 \ 5 \ 1 \ 3$$

Vollziehen Sie schrittweise die Seitenersetzungsstrategie LRU (Least Recently Used), indem Sie die leere Tabelle befüllen.

Anforderung	f_1	f_2	f_3	Page-Fault j/n
7				
2				
5				
0				
4				
0				
7				
5				
1				
3				