Aufgabe 1: **Gewichtete Buddy-Systeme** (3 + 3 + 3 Punkte)

Bei dem in dieser Aufgabe betrachteten gewichteten Buddy-System soll eine Aufteilung eines Speichers der Größe 1024 nach der im Skript angegebenen Wichtung in kleinere Blöcke erfolgen.

a) Zunächst sei der gesamte Speicher frei. Es folgt eine Anforderung der Größe 360, dann eine Anfrage der Größe 132. Skizzieren Sie die Speicherstruktur, die nach diesen beiden Anforderungen entsteht!

b) Geben Sie die Menge aller möglichen Größen an, die ein Block annehmen kann. Wie groß darf ein Block maximal sein, um in der unter a) erhaltenen Belegung noch gespeichert werden zu können?

c) Beweisen Sie, daß in gewichteten Buddy-Systemen bei der hier verwendeten Strategie immer zwei Buddies existieren, die die gleiche Größe haben! Geben Sie diese Größe im Verhältnis zum Gesamtspeicherplatz an!

Aufgabe 2: **Seitenersetzungsstrategien** (9 Punkte)

Gegeben sei ein physikalischer Speicher der Größe 3K. Über einem logischem Adreßraum von 5K soll die nachstehende Abfolge von Adreßzugriffen ausgeführt werden: (1024, 1023, 2047, 4097, 1026, 4095, 0, 4096, 5120, 1026, 5119, 1023).

Untersuchen Sie die Seitenersetzungsstrategien "LRU", "Second Chance" und "OPT" bei Seiten der Größe 1K (=1024 Byte). Geben Sie die Speicherzustände bei sequentieller Abarbeitung der Adreßanfragen an, und kennzeichnen Sie Seitenfehler. Zu Beginn sei der Speicher leer. Kann gemäß einer Strategie nicht eindeutig entschieden werden, wird nach FIFO verfahren.

Aufgabe 3: **Endliche Automaten** (1 + 3 + 5 Punkte)

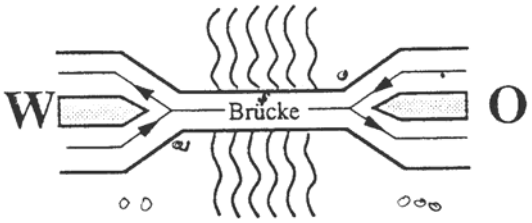
a) Um Seitenersetzungsstrategien mit endlichen Automaten darzustellen, werden die Speicherbelegungsmöglichkeiten als Zustände in einem Automaten aufgefaßt. Wieviele Zustände werden für einen Automaten unabhängig von der Seitenersetzungsstrategie höchstens benötigt, wenn man von einem gefüllten Speicher mit m Plätzen und n Seiten ($n \geq m$) ausgeht?

b) Geben Sie einen endlichen Automaten in graphischer Darstellung an, der die Strategie FIFO für 4 mögliche Seiten und 3 Speicherplätze ausgehend von der Speicherbelegung (1, 2, 3) realisiert, wobei 1 die zuletzt und 3 die zuerst eingetragene Seite darstellt.

c) Geben Sie für einen endlichen Automaten mit 3 Speicherplätzen die Übergangsfunktionen für die FIFO- und die LRU-Strategie für alle Speicherbelegungen an, bei denen die nächste Referenz r_t keinen Seitenaustausch verursacht.

Aufgabe 4: **Gegenseitiger Ausschluß mit Semaphoren** (8 + 5 + 2 Punkte)

Wagen, die aus Osten und Westen kommen, müssen eine Brücke über einen Fluß passieren (siehe nachfolgende Abbildung). Unglücklicherweise hat die Brücke nur eine Fahrspur. Sie kann also zu jedem Zeitpunkt nur von einem oder mehreren Wagen aus derselben Richtung benutzt werden.



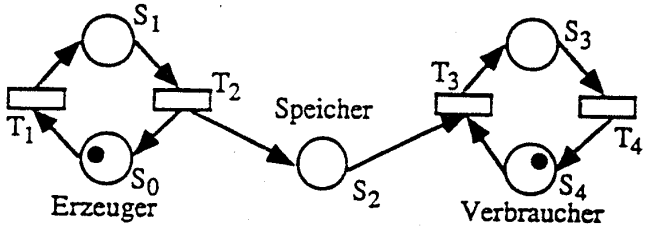
a) Schreiben Sie unter Verwendung von Semaphoren einen Algorithmus für die Koordination der Wagen aus den beiden Richtungen, wie sie an der Brücke ankommen, den Fluß überqueren und auf der anderen Seite wegfahren, ohne mit einem entgegenkommenden Wagen zu kollidieren.

b) Verfeinern Sie dieses Verfahren so, daß sich die Fahrtrichtung auf der Brücke zusätzlich auch dann ändert, wenn sie 10 Wagen aus einer Richtung passiert haben, und einer oder mehrere Wagen aus der anderen Richtung warten.

c) Wie kann man mit einer einfachen Modifikation des Verfahrens einer Fahrtrichtung Vorrang einräumen?

Aufgabe 5: **Modellierung mit Petri-Netzen** (3 + 4 + 2 Punkte)

Die folgende Abbildung stellt ein typisches Erzeuger-Verbraucher-Problem (EVP) dar, das mittels eines Petri-Netzes modelliert wurde.



a) Erweitern Sie das dargestellte Netz zu einem Petri-Netz, das ein Erzeuger-Zwei Verbraucher-Problem (E2VP) modelliert. Dabei soll der zweite Verbraucher so angeordnet sein, daß ein im Speicher befindliches Erzeugnis mit der nächsten Transition verbraucht werden kann!

b) Geben Sie einen Erreichbarkeitsgraphen für Ihr Petri-Netz, welches das E2VP modelliert, soweit an, daß dieser mindestens einen Zyklus enthält!

c) Kann es bei dem E2VP Petri-Netz zu einer teilweisen Verklemmung oder zu einem Deadlock kommen? Begründen Sie Ihre Aussage!

Aufgabe 6: **Sichere und codierte Übertragung** (4 + 5 Punkte)

Die Noten der Nachholklausur (1: bestanden, 0: nicht bestanden) sollen über einen unsicheren Boten vom Lehrstuhl für Informatik IV an das zentrale Prüfungsamt gemeldet werden. Um den Transfer abzusichern, wird das folgende Generatorpolynom P über dem booleschen Körper verwendet:

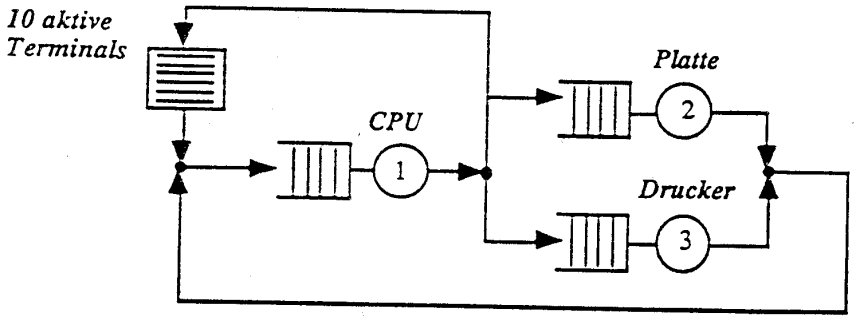
$$P(x) = x^5 + x^4 + 1$$

a) Berechnen Sie die tatsächlich übertragene Bitfolge, wenn der folgende "Noten"-Spiegel zugrundegelegt wird: 1, 1, 0, 0, 1, 1. Beschreiben Sie Ihr Vorgehen ausführlich!

b) Der Bote sei auf dem Weg zum ZPA durch ein paar Bierchen aufgehalten worden. Dadurch ist die Übertragung fehlerhaft verlaufen. Geben Sie eine Störfolge von Bits an, bei der das ZPA dennoch von einer korrekten Übertragung ausgehen muß, das Sicherungsverfahren also versagt hat.
(Tip: Überlegen Sie sich, welche Anforderungen an das Störpolynom gestellt werden, damit der Empfänger auf eine korrekte Übertragung schließt.)

Aufgabe 7: **Operationelle Analyse von Rechensystemen** (6 + 4 + 2 Punkte)

Gegeben sei das folgende Modell eines Rechensystems mit zwei E/A-Einheiten:



Aus Messungen ergaben sich für die relativen Häufigkeiten für Jobübergänge zwischen CPU und E/A-Geräten $q_{12} = 0{,}6$ und $q_{13} = 0{,}2$. Der Systemdurchsatz beträgt 1,25 Jobs/Sekunde. Die Bedienzeiten von CPU und E/A-Geräten betragen:

$S_1 = 0{,}1$ Sekunden/Request, $S_2 = 0{,}2$ Sekunden/Request, $S_3 = 3{,}9$ Sekunden/Request

a) Welche Zeit verbraucht ein Job im Mittel durch Plattenzugriffe und Ausgaben am Drucker?

b) Berechnen Sie die mittlere Denkzeit an den Terminals.

c) Welcher Systemteil wird zum Engpaß, wenn die Anzahl der Terminals erhöht wird?

Aufgabe 1: **Bankers-Algorithmus bei MRU-Strategie** (8 Punkte)

Gegeben seien drei Prozesse, die in einem Rechensystem um vier gleichwertige Betriebsmittel (BM) konkurrieren. Die Prozesse haben folgenden maximalen BM-Bedarf:

Prozeß 1: 2 BM Prozeß 2: 3 BM Prozeß 3: 4 BM

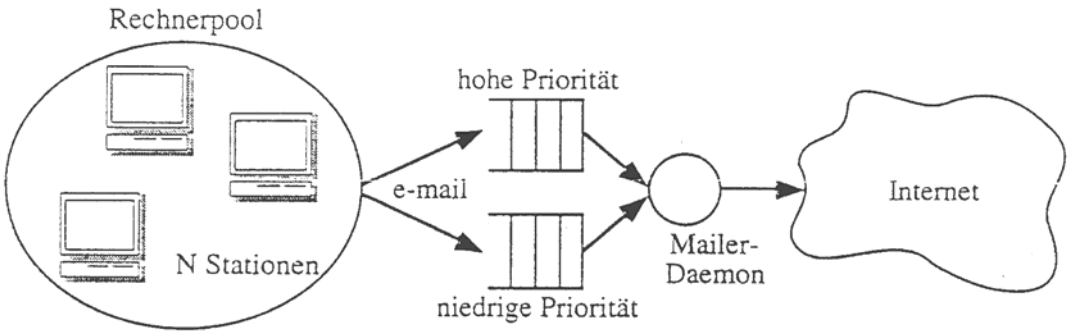
Zu Beginn seien drei BM verfügbar, eines sei bereits an Prozeß 1 vergeben. Folgender zeitlicher Ablauf sei vorgesehen:

t_1: Prozeß 3 fordert 3 BM
t_2: Prozeß 2 fordert 3 BM
t_3: Freigabe von 1 BM durch Prozeß 1
t_4: Freigabe von 2 BM, falls ein Prozeß sie zum Zeitpunkt t_1 oder t_2 erhalten hat
t_5: Prozeß 1 fordert 2 BM, Prozeß 2 fordert 1 BM

Führen Sie für alle Zeitpunkte t_1 bis t_6 den Banker´s Algorithmus unter Berücksichtigung der "Maximum Resource Usage"-Strategie durch, und geben Sie an, ob die geforderten BM zugeteilt werden oder nicht. Geben Sie im Falle der Zuteilung die Reihenfolge an, in der die Prozesse nach dem Banker´s Algorithmus lauffähig wären. Falls einem Prozeß die Zuteilung von BM verweigert wird, wird nach jeder Freigabe von BM geprüft, ob die angeforderten BM nun verfügbar sind.

Aufgabe 2: **Prozeß-Synchronisation mit Semaphoren** (9 Punkte)

Der CIP-Pool der Informatik sei mit folgendem Mail-System ausgestattet: Die Stationen legen ihre Post auf einem Mail-Server ab, der zwei Puffer für hohe und niedrige Priorität bereitstellt. Sind dort Nachrichten abgelegt, werden sie von einem Mailer-Daemon abgeholt, der sie ins Internet absetzt:



Der Mailer-Daemon arbeitet nach folgendem Prinzip:

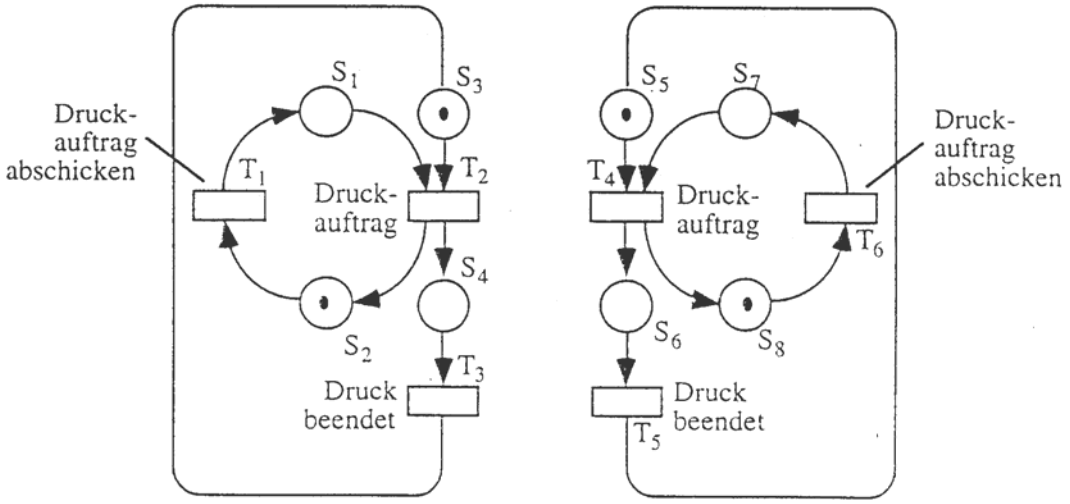
- Hat er eine Mail abgesetzt, prüft er, ob keine weiteren zur Bearbeitung anstehen. Ist dies der Fall, legt er sich schlafen.
- Ist mindestens eine Mail zu bearbeiten, prüft er zunächst den Puffer hoher Priorität auf Inhalt. Wenn dort ein Paket ansteht, wird dieses bedient, im anderen Fall wird eine Mail niedriger Priorität abgesetzt.

Synchronisieren Sie das obige Problem mittels binärer Semaphore. Benutzen Sie zwei Semaphore, einen, der den Zustand des Mailer-Daemons beschreibt, und einen für den Zugriff auf die Puffer. Gehen Sie davon aus, daß die Funktionen "e-mail generieren" (gibt die Priorität der erzeugten e-mail in einer Variable `prioritaet` zurück), "e-mail ablegen", "e-mail lesen" und "e-mail absetzen" zur Verfügung stehen.

Aufgabe 3: **Modellierung eines Druckauftrags mit Petri-Netzen** (4+3+2 Punkte)

Gegeben sei ein verteiltes System, das aus zwei Rechnern und einem Drucker besteht. Beide Rechner haben physikalischen Zugang zum Drucker, allerdings können die Druckaufträge nur nacheinander abgearbeitet werden. Dieser Sachverhalt soll mittels Petri-Netzen modelliert werden.

a) Gegeben seien die folgenden Petri-Netze, die zwei Rechenprozesse beschreiben. Das entsprechende Petri-Netz soll so ergänzt werden, daß immer nur höchstens einer der beiden Prozesse drucken kann und die beiden Teilnetze zu einem Netz zusammengefügt werden. Die Zuhilfenahme zusätzlicher Stellen und/oder Transitionen ist dabei erlaubt.



b) Geben Sie zu Ihrem entstandenen Petri-Netz den Erreichbarkeitsgraphen so weit an, daß dieser mindestens einen Zyklus enthält!

c) Kann es bei Ihrem ergänzten Petri-Netz zu einer teilweisen Verklemmung kommen? Kann ein Deadlock auftreten? Begründen Sie ihre Entscheidungen!

Aufgabe 4: **Seitenersetzungsstrategien** (2+4+2 Punkte)

Im folgenden sollen Speicherbelegungsmöglichkeiten als Zustände in einem Automaten aufgefaßt werden. Durch diese Vorgehensweise ist es möglich, Seitenersetzungsstrategien darzustellen. Es wird für alle Teilaufgaben von einem vollständig gefüllten Speicher mit m Plätzen ausgegangen. Zusätzlich stehen noch k Seiten zur Verfügung, die ebenfalls benötigt werden.

a) Geben Sie für den allgemeinen Fall, daß keine Seitenersetzungsstrategie bekannt ist, an, wieviele Zustände für den Automaten maximal benötigt werden!

b) Gehen Sie von m=3 Speicherplätzen aus, welche in der Form (A,B,C) dargestellt werden, wobei A die zuerst eingetragene Seite bezeichnet und C den letzten Eintrag. Zusätzlich existiert eine Seite, die mit D bezeichnet wird. Geben Sie die Zustandsübergänge zweier Automaten an, die die Seitenersetzungsstrategien LIFO und FIFO für das unter a) beschriebene Szenario modellieren. Gehen Sie von der Anfangsspeicherbelegung (A,B,C) aus!

c) Wieviel Prozent der von Ihnen unter a) insgesamt errechneten Zustände werden in diesen beiden Fällen jeweils genutzt?

Aufgabe 5: **Bestimmung von Prüfsummen** (4+4 Punkte)

In dieser Aufgabe sollen Sie zwei Funktionen zur Prüfsummenberechnung programmieren. Den Funktionen werden zwei Parameter übergeben: als erstes ein Zeiger auf ein `unsigned char` Feld, in dem die zu prüfende Bytefolge liegt, und als zweiter Parameter eine `int` Zahl, die die Länge der zu untersuchenden Folge angibt. Als Ergebnistyp sollen beide Funktionen die Prüfsumme als `int` Wert zurückliefern.

a) Die Parity-Funktion liefert als Ergebnis den Wert 1, falls die Anzahl der auf "1" gesetzten Bits im übergebenen Feld ungerade ist, sonst 0. Beispiel:

```
unsigned char b[8] = {1,2,3,4,5,6,7,8};
int i;
i = parity(b,8);
```

Die Funktion liefert den Wert 1, da in der Dualzahldarstellung die Anzahl auf 1 gesetzter Bits im Feld b ungerade ist.

Aufgabe 1: **Bankers-Algorithmus** (8 Punkte) Mid

Gegeben seien drei Prozesse P_1, P_2 und P_3, die auf einen in Seiten der Größe 1024 Byte aufgeteilten Speicherbereich der Größe 5K und auf einen zentralen Druckerpool mit drei Druckern zugreifen. Zum Zeitpunkt k ist von folgender Belegung auszugehen:

$H_1(k) = (2, 1)$, $H_2(k) = (1, 0)$, $H_3(k) = (1, 1)$
(P_1 belegt also zwei Seiten und einen Drucker usw.)

Führen Sie den Bankers-Algorithmus für folgende Anfragen durch:

i) $Q_1(k) = (3, 0)$, $Q_2(k) = (0, 3)$, $Q_3(k) = (0, 0)$
ii) $Q_1(k) = (3, 2)$, $Q_2(k) = (0, 0)$, $Q_3(k) = (2, 1)$

Geben Sie v(k) und jeweils die schrittweisen Änderungen von i, j, W und δ_j an, und kennzeichnen Sie etwaige Deadlockmengen.

Aufgabe 2: **Deadlocks** (12 Punkte) Easy

Zwei Prozesse A und B benötigen zu ihrer Ausführung je eine gewisse Zeitdauer t_A bzw. t_B und drei BM X, Y und Z. $t_{R,P}$ bezeichne die Zeitdauer, die das BM R vom Prozeß P während seiner Ausführungszeit benötigt wird. Eine Gruppe von Systemexperten versucht, Aussagen zu machen, wann ein Deadlock eintreten kann bzw. wann nicht.

EXPERTE 1: Ein Deadlock tritt auf jeden Fall dann auf,
wenn sowohl $t_{X,A} + t_{Y,A} + t_{Z,A} > t_A$ als auch $t_{X,B} + t_{Y,B} + t_{Z,B} > t_B$ ist.

EXPERTE 2: Ein Deadlock ist sicher, wenn $t_{X,A} \cdot t_{X,B} + t_{Y,A} \cdot t_{Y,B} + t_{Z,A} \cdot t_{Z,B} > t_A \cdot t_B$ ist.

EXPERTE 3: Ein Deadlock kann nicht auftreten, wenn gleichzeitig die folgenden beiden Bedingungen erfüllt sind: $t_{X,A} + t_{Y,A} + t_{Z,A} < t_A$ und $t_{X,B} + t_{Y,B} + t_{Z,B} < t_B$.

EXPERTE 4: Seien t_1 bis t_6 aufeinanderfolgende Zeitpunkte, die in der Ausführungszeit der Prozesse liegen. Beide Prozesse starten zum Zeitpunkt 0.

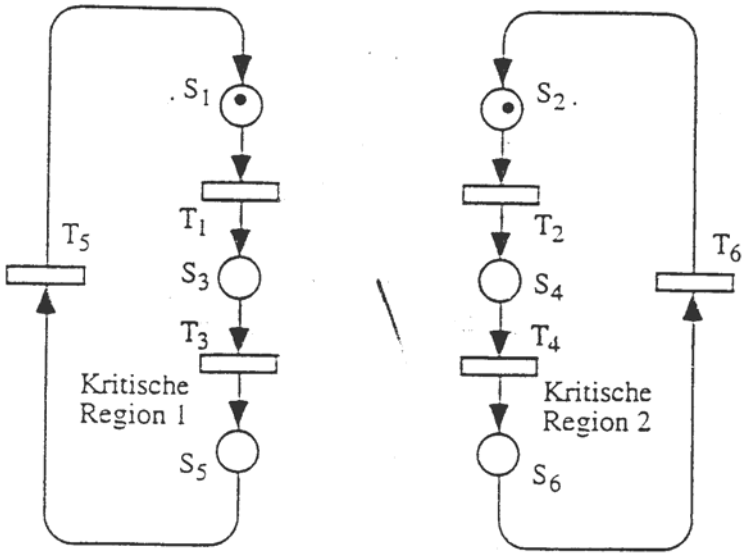
Zum Zeitpunkt t_1 benötigt Prozeß A das BM X,
zum Zeitpunkt t_2 benötigt Prozeß B das BM Y,
zum Zeitpunkt t_3 benötigt B das BM X,
zum Zeitpunkt t_4 benötigt A das BM Y,
zum Zeitpunkt t_5 gibt A beide Betriebsmittel frei
zum Zeitpunkt t_6 gibt B seine BM frei,

dann gibt es keinen zulässigen Ausführungsschritt, der unsichere Bereiche und Deadlockzustände vermeidet.

Wieviele Experten haben Recht? Begründen Sie Ihre Entscheidung im Falle einer wahren Expertenaussage kurz, bzw. widerlegen Sie die Aussagen der sich irrenden Experten mit einem (ggf. graphischen) Gegenbeispiel!

Aufgabe 3: **Kritische Regionen** (4 + 6 Punkte)

In zwei Prozessen gebe es je eine kritische Region. Diese umfaßt die Transitionen T_1 und T_3 bzw. T_2 und T_4 sowie das jeweils eingeschlossene und als Stelle modellierte Betriebsmittel. In der unten dargestellten Modellierung der beiden Prozesse als Petri-Netze ist noch nicht ausgeschlossen, daß nur höchstens einer der beiden Prozesse Zugriff auf die kritische Region hat.



a) Ergänzen Sie das dargestellte Petri-Netz durch Hinzufügung einer beliebigen Anzahl von Stellen und/oder Transitionen so, daß sich nicht mehr beide Prozesse gleichzeitig in ihren kritischen Regionen befinden können! Beachten Sie, daß Ihre Lösung unabhängig sein muß von der Reihenfolge, in der die Prozesse in den kritischen Bereich eintreten wollen.

b) Geben Sie den Erreichbarkeitsgraphen Ihres erweiterten Petri-Netzes an, und entscheiden Sie, ob es zu einem Deadlock oder einer teilweisen Verklemmung kommen kann!

Aufgabe 4: **Buddy-Systeme** (5 + 3 Punkte) OK

Gegeben sei ein Speicherbereich der Größe 64. In diesem seien aktuell drei Speicherbereiche belegt, von denen jeder die Größe 4 besitzt. Die Listen der freien Speicherplätze sind wie folgt angegeben:

$L_6 = L_5 = \emptyset$
$L_4 = \{0010000, 0110000\}$
$L_3 = \{0000000, 0100000\}$
$L_2 = \{0001000\}$
$L_1 = L_0 = \emptyset$

a) Geben Sie die entsprechende baumartige Struktur des ungewichteten Buddy-Systems an!

b) Das belegte Speichersegment mit der kleinsten Adresse wird freigegeben. Geben Sie die danach entstehenden neuen Listen der freien Speicherplätze an!

Aufgabe 5: **Working-Set-Größe** (10 + 3 Punkte) ?

Gegeben sei der folgende Referenzstring: $\omega = (1, 1, 2, 1, 1, 3, 2, 1, 1, 1)$.

a) Bestimmen Sie die Working-Sets für die Zeitpunkte $t = t_1, ..., t_{10}$ in Abhängigkeit von der Fensterbreite $h = 1,2,...$. Gehen Sie von einem leeren Working-Set zum Zeitpunkt t_0 aus. Stellen Sie die durchschnittliche Working-Set-Größe als Funktion der Fensterbreite h graphisch dar.

b) Bestimmen Sie die optimale Fensterbreite h_{opt}: Wenden Sie dazu als Kriterium die 50%-Regel von Knuth an, hier bezogen auf den jeweils maximal zur Verfügung stehenden Speicher.

Aufgabe 1: Bankers-Algorithmus (4+2 Punkte) Mist

Gegeben seien drei Prozesse P_1, P_2 und P_3, die sich die Betriebsmittel BM_1, BM_2 und BM_3 teilen. Es stehen 10 BM vom Typ1 und jeweils 2 BM von Typ 2 und 3 zur Verfügung, die nur exclusiv nutzbar sind. Von folgender Habenbelegung zum Zeitpunkt k ist für die Teilaufgaben auszugehen:

$H_1(k) := (3, 0, 0)$, $H_2(k) := (4, 1, 1)$, $H_3(k) := (1, 0, 1)$

a) Führen Sie den in Übungsaufgabe 25 angegebenen Bankers-Algorithmus für folgende Anfragen durch:

i) $Q_1(k) := (2, 1, 0)$, $Q_2(k) := (2, 0, 0)$, $Q_3(k) := (0, 1, 1)$
ii) $Q_1(k) := (1, 0, 0)$, $Q_2(k) := (0, 1, 1)$, $Q_3(k) := (6, 1, 0)$
iii) $Q_1(k) := (3, 2, 0)$, $Q_2(k) := (2, 1, 0)$, $Q_3(k) := (7, 0, 1)$

Geben Sie v(k) und jeweils die schrittweisen Änderungen von i, j, W und δ_i an.

b) Falls sich in den Anfragen i) - iii) eine Teilmenge D von Prozessen in einem Deadlock befindet, prüfen Sie dieses formal nach.

Aufgabe 2: Segmentierung/Paging (4+2 Punkte) geht so

a) Für die Speicherverwaltung nach dem Segmentierungsverfahren ist für ein Programm folgende Segmenttabelle gegeben (Länge in Speicherworten):

Segment-Nr.	Länge	Basis
0	600	2019
1	14	200
2	100	230
3	750	507
4	96	1257

i) Wieviele Speicherworte stehen dem Programm im physikalischen Speicher zur Verfügung?
ii) Welches ist die kleinste und welches die größte verfügbare physikalische Adresse?
iii) Berechnen Sie zu den folgenden physikalischen Adressen jeweils die logischen Adressen:

1) 527 2) 1256 3) 1257 4) 225

b) Betrachten Sie ein Programm mit 3 logischen Modulen der Größe von 700, 200 und 500 Worten. Wie hoch ist der Speicherbedarf bei Verwendung des Segmentierungsverfahren? Wie hoch ist der Speicherbedarf bei Verwendung von Pagingverfahren mit einer jeweiligen Seitengröße von 200, 500, 600 oder 700 Wörtern? Bei allen Überlegungen können Sie davon ausgehen, daß genügend Speicherplatz im Hauptspeicher zur Verfügung steht. Diskutieren Sie die Ergebnisse.

Aufgabe 3: Readers-Writers-Problem (4+5 Punkte) geil

a) Stellen Sie sowohl das 1. Courtois-Problem (keine Prioritäten) als auch das 2. Courtois-Problem (Priorität für den Schreiber) in einem Petri-Netz dar. Die Zahl n potentieller Leser/Schreiber sei beliebig, aber fest. Geben Sie jeweils die Anfangsbelegungen der Petri-Netze mit den Marken an.

b) Schreiben Sie in der informellen Notation der Vorlesung einen Algorithmus, der das 2. Courtois-Problem mit Hilfe von Semaphoren löst.