Aufgabe 1: Cacheverwaltung – Least Frequently Used

Für einen anfangs leeren, *Full Associative* Cache sei eine Sequenz von Adresszugriffen gegeben. Der Cache besitzt 4 Blöcke, die jeweils 16 Datenworte aufnehmen können. Der Cache verwendet eine *Least Frequently Used* Ersetzungsstrategie. Das bedeutet, dass im Konfliktfall jener Block ersetzt wird, der am seltensten angesprochen wurde. Falls es mehrere Blöcke mit gleich geringer Zugriffszahl gibt, wird davon jener entfernt, der sich bereits am längsten im Cache befindet (*First In, First Out*).

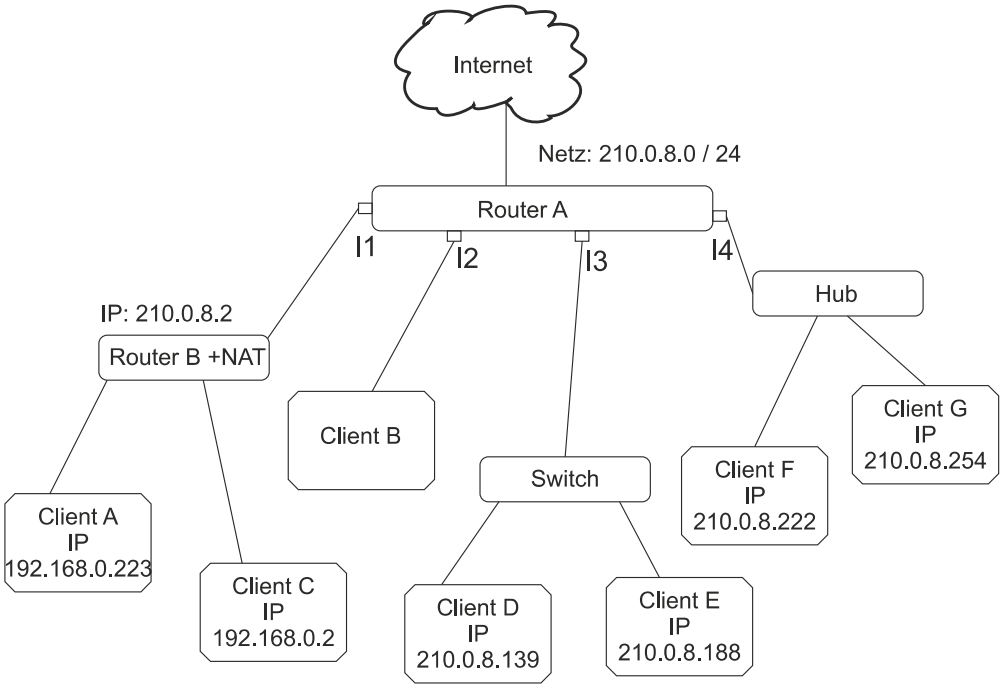
a) Tragen Sie in die nachfolgende Tabelle ein, welche Adressreferenz ein *hit* oder ein *miss* ist. Geben Sie an, welcher Cache-Block dabei angesprochen wird. Falls der Zugriff ein *miss* ist, tragen Sie bitte die Adressen ein, deren Inhalte in den Cache-Block geladen werden.

b) Was ist der Nachteil dieser Methode? Betrachten Sie dazu die Belegungen von Block 1 und Block 3.

Adresse	hit/miss	Cache-Block	# Zugriffe	Inhalt
70	miss	0	1	64 . . . 79
44				
141				
46				
78				
20				
75				
131				
177				
79				
132				
64				
23				
24				
56				

Aufgabe 2: Netzwerke

Gegeben ist die folgende Netzwerkstruktur:



Das vom NIC vergebene Netzwerk ist 210.0.8.0/24 und wird von Router A verwaltet. Beantworten Sie dazu folgende Fragen:

a) Welche Subnetzmaske muss Router A für die Interfaces I1 bis I4 vergeben, wenn das Netzwerk in vier gleich große Bereiche aufgeteilt werden soll?

b) Welche IP-Adressen bekommen die Interfaces I1 bis I4, wenn immer die erste verfügbare des jeweiligen Bereiches verwendet wird?

c) Wie groß ist das von Router B verwaltete Subnetz minimal? Geben Sie nur die Subnetzmaske an und betrachten Sie dazu die IP-Adressen von Client A und Client C.

d) Welche Netzwerkelemente sehen ein Datenpaket, das von Client D an die Adresse 210.0.8.222 versendet wird?

e) Welche Netzwerkelemente sehen ein Datenpaket, das von Client G an die Adresse 210.0.8.191 versendet wird?

Aufgabe 3: Netzwerk – ARP

Das *Address Resolution Protocol* (ARP) ist ein Netzwerkprotokoll, das zu einer Netzwerkadresse der Internetschicht die physikalische Adresse (Hardwareadresse) der Netzzugangsschicht ermittelt.

Ein ARP-Paket ist wie folgt aufgebaut:

<table>
<tr><td>Bit 0-7</td><td>Bit 8-15</td><td>Bit 16-23</td><td>Bit 24-31</td></tr>
<tr><td colspan="2">Hardwareadresstyp</td><td colspan="2">Protokolladresstyp</td></tr>
<tr><td>Hardwareadressgröße</td><td>Protokolladressgröße</td><td colspan="2">Operation (0=Reply, 1=Request)</td></tr>
<tr><td colspan="4">Quell-MAC-Adresse (Byte 1-4)</td></tr>
<tr><td colspan="2">Quell-MAC-Adresse (Byte 5-6)</td><td colspan="2">Quell-IP-Adresse (Byte 1-2)</td></tr>
<tr><td colspan="2">Quell-IP-Adresse (Byte 3-4)</td><td colspan="2">Ziel-MAC-Adresse (Byte 1-2)</td></tr>
<tr><td colspan="4">Ziel-MAC-Adresse (Byte 3-6)</td></tr>
<tr><td colspan="4">Ziel-IP-Adresse</td></tr>
</table>

Hinweis: Da die Daten im Big-Endian-Format (network order) vorliegen und die Bits von links nach rechts nummeriert sind, befindet sich das MSB links. Beispiel: "0F 23" = $(3875)_{10}$.

Gegeben ist folgendes ARP-Paket:

```
00 01 08 00 06 04 00 01 00 23 14 ca 52 b5 80 83
c4 20 00 00 00 00 00 00 80 83 c0 01
```

a) Ermitteln Sie daraus die Ziel- und Quell-IP-Adresse und geben Sie diese in Dezimalnotation an.

b) Wie lautet die Quell-MAC-Adresse des Pakets in Hexadezimalnotation?

c) Handelt es sich bei diesem Paket um einen ARP-*Request* oder ein ARP-*Reply*?

d) Welche Felder ändern sich wenn ARP für IPv6 eingesetzt wird?

Aufgabe 4: Prozesse – Prozesszustand

Gegeben ist der folgende Prozess in Pseudo-Code Notation:

```
    int main() {
 1:   r= get_input();
 2:   f();
 3:   u();
 4:   return 0;
    }

    p() {
 5:   b= a*3
    }

    f() {
 6:   a= r*r;
 7:   p();
 8:   F= b;
    }

    u() {
 9:   a= r*2;
10:   p();
11:   U= b;
    }
```

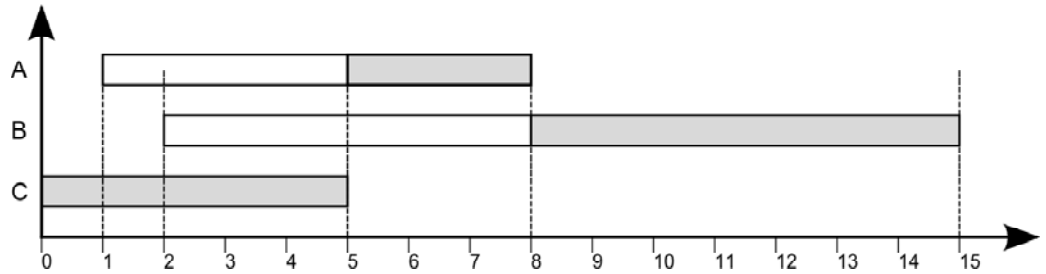
Nehmen Sie an, dass die Prozessausführung bei 1 beginnt und dass mit `get_input()` der Wert 5 eingelesen wird. Tragen Sie in die folgende Tabelle für jeden Schritt den Zustand des Prozesses *am Beginn* des jeweiligen Schrittes ein (siehe Foliensatz 20-21–Prozessmanagement, Folien 13ff).

Hinweis: Alle Variablen sind als global deklariert!

PC	r	a	b	F	U	Stack
1						
2						
6						
7						
5						
8						
3						
9						
10						
5						
11						
4						

Aufgabe 5: Scheduling – Analyse eines Scheduling-Algorithmus

Gegeben ist die folgende graphische Darstellung des Ablaufs der Prozesse A, B und C:



a) Welcher Scheduling-Algorithmus könnte dem dargestellten Ablauf entsprechen?

Berücksichtigt der Algorithmus Prioritäten der einzelnen Prozesse? Kann die Frage, ob er *preemptive* oder *non-preemptive* ist, eindeutig beantwortet werden?

b) Befüllen Sie die nachfolgende Tabelle, in der Sie für jeden Prozess den Startzeitpunkt, sowie die geplante und die tatsächliche Ausführungsdauer eintragen. Unter der geplanten Zeit wird hier jene Zeit verstanden, in der der Prozess tatsächlich am Prozessor abgearbeitet wird – mit anderen Worten die Zeit, die der Prozess im Zustand RUNNING verbringt.

Prozess	Startzeit	geplante Laufzeit	tatsächliche Laufzeit
A			
B			
C			

c) Geben Sie für jeden Prozess an, wann er sich im Zustand READY und wann er sich im Zustand RUNNING befindet.

Aufgabe 6: Scheduling – Non-Preemptive Priority Scheduling

Gegeben ist die folgende Prozesstabelle mit Priorität (hoher Wert = hohe Priorität), Startzeitpunkt und Ausführungsdauer des jeweiligen Prozesses.

Prozess	Startzeit	Laufzeit	Priorität
A	8	4	1
B	0	2	3
C	6	1	2
D	0	3	4
E	4	5	3
F	9	3	0

a) Stellen Sie graphisch den Ablauf der Prozesse dar, wenn der Scheduler einen *Non-Preemptive Priority Scheduling*-Algorithmus verwendet (siehe Foliensatz 20-21–Prozessmanagement, Folien 45ff).

b) Geben Sie für jeden Prozess den Zeitpunkt an, zu dem er in den Zustand RUNNING gelangt.

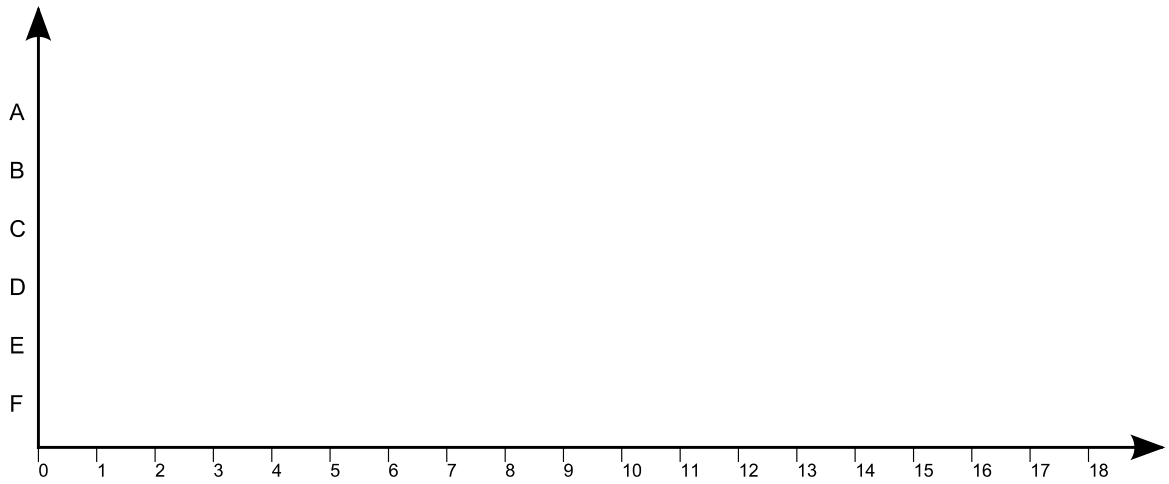
Aufgabe 7: Scheduling – Preemptive Priority Scheduling

Gegeben ist die folgende Prozesstabelle mit Priorität (hoher Wert = hohe Priorität), Startzeitpunkt und Ausführungsdauer des jeweiligen Prozesses.

Prozess	Startzeit	Laufzeit	Priorität
A	9	4	0
B	1	2	2
C	5	2	3
D	0	4	1
E	6	4	4
F	13	2	2

a) Stellen Sie graphisch den Ablauf der Prozesse dar, wenn der Scheduler einen *Preemptive Priority Scheduling*-Algorithmus verwendet.

b) Geben Sie für jeden Prozess an, wie lange es dauert, bis der Prozess vollständig ausgeführt ist. Vergleichen Sie diese Zeit mit der ursprünglich angegebenen Laufzeit.



Aufgabe 8: Interprozesskommunikation – Race Conditions

Gegeben sind zwei Prozesse, die Textdaten über ein gemeinsames Array B austauschen. Die Prozesse benutzen dazu zwei Befehle:

read(B[i]) liest das Zeichen an der i-ten Position von B.

write(B[i],c) schreibt das Zeichen c an die i-te Position von B.

Die beiden Prozesse sind wie folgt programmiert:

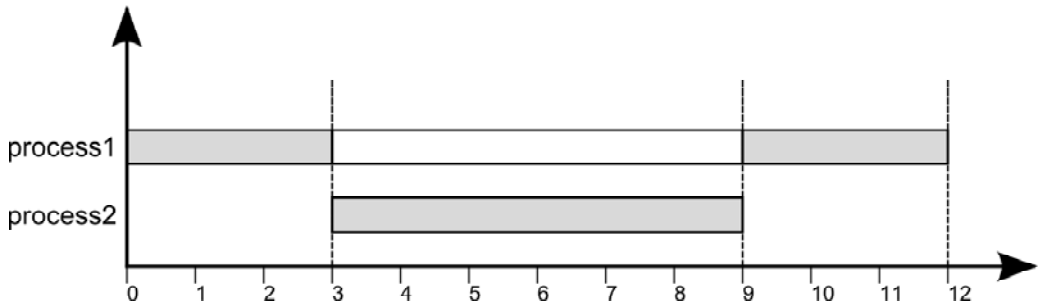
```
    process1() {
1:    read(B[1]);
2:    read(B[2]);
3:    read(B[3]);
4:    read(B[4]);
5:    read(B[5]);
6:    read(B[6]);
    }
```

```
    process2() {
1:    write(B[1],'B');
2:    write(B[2],'a');
3:    write(B[3],'r');
4:    write(B[4],'n');
5:    write(B[5],'e');
6:    write(B[6],'y');
    }
```

Das Array B ist zu Beginn folgendermaßen belegt:

Index	B[1]	B[2]	B[3]	B[4]	B[5]	B[6]
Initialbelgegung	'F'	'r'	'e'	'd'	' '	' '

Es wird angenommen, dass jeder *read*- oder *write*-Befehl genau einen Takt zur Ausführung benötigt. Die beiden Prozesse werden auf einem System mit einem *Preemptive Priority Scheduling*-Algorithmus in folgender Weise ausgeführt, wobei `process1()` eine niedrigere Priorität als `process2()` hat:



a) Welche (falsche) Zeichenkette wird von `process1()` gelesen?

b) Welche (falsche) Zeichenkette wird von `process1()` jeweils gelesen, wenn `process2()` einen Schritt früher oder später startet?

c) Wäre es eine Lösung, die Prioritäten der beiden Prozesse zu vertauschen oder beiden Prozessen die gleiche Priorität zu geben? Geben Sie dabei auch an, ob Ihre Antwort vom eingesetzten Scheduling-Algorithmus abhängt.

Aufgabe 9: Interprozesskommunikation – Semaphore

Gegeben sind zwei Prozesse, die Textdaten über ein gemeinsames Array B austauschen. Die Prozesse benutzen dazu zwei Befehle *read* und *write* (vgl. Aufgabe 8). Um Probleme beim Zugriff auf den gemeinsamen Datenbereich zu umgehen, verwenden die beiden Prozesse eine Synchronisation mittels einer Semaphor-Variable S:

```
    process1() {
1:    P(S);
2:    read(B[1]);
3:    read(B[2]);
4:    read(B[3]);
5:    read(B[4]);
6:    read(B[5]);
7:    read(B[6]);
8:    V(S);
    }
```

```
    process2() {
1:    P(S);
2:    write(B[1],'B');
3:    write(B[2],'a');
4:    write(B[3],'r');
5:    write(B[4],'n');
6:    write(B[5],'e');
7:    write(B[6],'y');
8:    V(S);
    }
```

Das Array B ist zu Beginn folgendermaßen belegt:

Index	B[1]	B[2]	B[3]	B[4]	B[5]	B[6]
Initialbelgegung	'F'	'r'	'e'	'd'	' '	' '

Die beiden Prozesse werden auf dem System gemäß der folgenden Tabelle ausgeführt:

Prozess	Erste Startzeit	Zweite Startzeit	Laufzeit
process1	0	11	8
process2	2		8

a) Zeichnen Sie graphisch den Ablauf der drei Prozessausführungen (2x `process1()`, 1x `process2()`). Nehmen Sie ein *Round Robin Scheduling*-Verfahren ohne Prioritäten mit einer Zeitscheibenlänge von 3 an (siehe Buch *Einführung in die Technische Informatik*, Seite 269f).

b) Geben Sie für jeden Schritt den Wert der Variablen S an.

Aufgabe 10: Interprozesskommunikation – Priority Inversion

Gegeben seien drei Prozesse: H mit höchster Priorität, M mit mittlerer Priorität und L mit geringster Priorität. H und L verwenden eine Semaphore S, um den Zugriff auf eine gemeinsame Ressource zu koordinieren. Die Prozesse laufen auf einem System mit einem *Preemptive Scheduling*-Verfahren, das den Prozessor grundsätzlich immer dem Prozess mit der höchsten Priorität zuteilt, sofern dieser Prozess nicht blockiert ist.
Die folgende Tabelle gibt an, zu welchem Zeitpunkt der jeweilige Task gestartet wird und wie lange der Task läuft. Weiters gibt die Tabelle an, wann die Operationen P(S) und V(S) ausgeführt werden. Die Zeiten werden immer relativ zum Startzeitpunkt des jeweiligen Prozesses gemessen und zwar unter der Annahme, dass der Prozess nicht unterbrochen wird.

Prozess	Startzeit	P(S)	V(S)	Dauer
H	5	3	5	7
M	7	–	–	7
L	0	4	8	9

a) Stellen Sie den Ablauf der Prozesse graphisch dar.

b) Wenn Sie den Ablauf betrachten – was fällt Ihnen dabei auf?