

# Übungsblatt 6: Verklemmungen und Interprozess-Kommunikation

⇒ Abgabe der Lösungen bis Montag, 29. November 14:00 im AsSESS

## Aufgabe 1: Grundlagen (1+1+1 = 3 Punkte)

1. Beschreiben Sie den Unterschied zwischen einem Deadlock und einem Livelock.
2. Was sind Vor- und Nachteile von synchronem und asynchronem Nachrichtenaustausch zwischen Prozessen?
3. Erklären Sie den gewünschten und tatsächlichen Effekt des folgenden Systemaufrufs in einem C-Programm unter Linux: `signal(SIGKILL, SIG_IGN);`

## Aufgabe 2: Deadlock? (1+2 = 3 Punkte)

Betrachten Sie ein System mit vier gleichartigen Ressourcen, die von drei Prozessen gemeinsam genutzt werden. Jeder Prozess benötigt höchstens zwei Ressourcen.

1. Identifizieren Sie die mögliche kritische Situation in diesem System und stellen Sie diese in einem Betriebsmittelbelegungsgraphen dar. Geben Sie dazu eine Liste der Kanten des Graphen in folgender Notation an:  $A \rightarrow B$  für „es gibt eine gerichtete Kante von A nach B“.
2. Kann in diesem System ein Deadlock entstehen? Begründen Sie Ihre Antwort.

## Aufgabe 3: Bankier-Algorithmus (3+4 = 7 Punkte)

Betrachten Sie den folgenden Zustand eines Systems von fünf Prozessen  $P_0, \dots, P_4$ , in dem die Belegung der Ressourcen  $A, B, C, D$  mit dem Bankier-Algorithmus verwaltet wird. Von Ressource  $A$  sind im System 3 Instanzen vorhanden, von Ressource  $B$  gibt es 14 Instanzen und von den Ressourcen  $C$  und  $D$  jeweils 12 Instanzen.

	A	B	C	D
$P_0$	0	0	1	2
$P_1$	1	0	0	0
$P_2$	1	3	5	4
$P_3$	0	6	3	2
$P_4$	0	0	1	4

Aktuell belegt

	A	B	C	D
$P_0$	0	0	0	0
$P_1$	0	7	5	0
$P_2$	1	0	0	3
$P_3$	0	0	2	0
$P_4$	0	6	4	2

Maximale Nachforderung

1. Untersuchen Sie mit dem Bankier-Algorithmus, ob dieser Zustand sicher ist. Geben Sie die mögliche Ablaufreihenfolge und für jeden Schritt die freien Ressourcen an.

2. Nehmen Sie an, dass nun Prozess  $P_1$  weitere Ressourcen anfordert, nämlich 5 Instanzen von  $B$  und 2 Instanzen von  $C$ . Kann diese Anforderung sofort erfüllt werden? Geben Sie die Startkonfiguration des Algorithmus und die Schritte wie in Aufgabenteil 1 an.

Die Graphen-Notation aus Aufgabe 2 lässt sich mit wenig Aufwand in eine Eingabe für das Visualisierungs-Tool Graphviz umwandeln, indem sie am Anfang um `digraph {` und am Ende um `}` ergänzt wird. Solche Graphviz-Dateien (Dateiendung: `.dot`) lassen sich z.B. mit dem Programm `xdot` anzeigen oder mit `dot -Tpdf -o ausgabe.pdf eingabe.dot` in ein PDF zeichnen. Die Form der Knoten lässt sich ebenfalls festlegen:

```
digraph {
    node [shape=circle] A, C
    node [shape=box] B, D
    A -> B
    B -> C
    C -> D
}
```

