

Aufgabe 1:

25 + 10 Punkte

1. Welche Klassen von Unterbrechungen gibt es? Wodurch werden die Unterbrechungen jeweils ausgelöst?
2. Was ist der Hauptunterschied zwischen einer synchronen und einer asynchronen Unterbrechung?

Aufgabe 2:

15 Punkte

Um die Zugriffszeit auf eine Platte zu beschleunigen, werden die drei Techniken

1. SSTF und SCAN zur Verwaltung der Warteschlange der Übertragungsaufträge,
2. Interleaving,
3. Puffer z. B. für eine Spur

eingesetzt. Welche der Techniken kann

- die Suchzeit,
- die Latenzzeit,
- die Übertragungszeit

reduzieren? Warum?

Aufgabe 3:

10 + 10 Punkte

Ein bestimmter Diskettentyp (ED-Diskette) hat 2 Seiten mit je 80 Spuren mit je 18 Sektoren bei einer Sektorgröße (Blockgröße) von 1 kByte.

- a) Über wieviel Speicherplatz verfügt eine solche Diskette? Wieviel Bit benötigt man für eine Blockadresse?
- b) Die Diskette wird formatiert, d. h. ein Dateisystem mit Sektoradresstabellen (i-nodes) wird angelegt. Dabei wird ein Bootblock und ein Superblock erzeugt, außerdem werden 184 i-nodes mit je 128 Byte fest reserviert. Wie groß darf eine Datei maximal sein, die auf diesem Dateisystem abgespeichert werden kann?

Aufgabe 4:

34 + 10 Punkte

1. Geben Sie die im Kurs vorgestellten Prozesszustände und die Übergänge zwischen diesen Zuständen an. Zeichnen Sie das Prozesszustandübergangsdiagramm. Erklären Sie jeweils kurz (in einem Satz), warum ein bestimmter Zustandsübergang stattfindet.
2. Wir betrachten folgendes Problem: Es soll ein neuer Prozess erzeugt werden, oder der rechnende Prozess benötigt mehr Speicher, es ist aber nicht genügend Hauptspeicher verfügbar. Deshalb sollen existierende Prozesse als Ganzes auf die Festplatte ausgelagert werden können, vorrangig die zur Zeit blockierten Prozesse, ggfs. auch andere, damit der frei gewordene Platz für einen anderen Prozess genutzt werden kann.

Fügen Sie einen zusätzlichen Zustand *ausgelagert* und die sinnvollen Übergänge in das Prozesszustandübergangsdiagramm von oben ein, und erklären Sie wieder die Übergänge.

Aufgabe 5:

7 + 7 + 7 + 7 Punkte

Ein Computer hat einen 32-Bit-virtuellen Adressraum und einen 30-Bit physischen Adressraum. Eine Seitengröße beträgt 1 kByte und ein Seitentabelleneintrag 4 Byte. Wir betrachten ein System mit mehrstufigen Seitentabellen.

1. Wie viele Stufen von Seitentabellen werden mindestens benötigt, wenn eine Seitentabelle maximal so lang wie eine Seite sein darf?
2. Wie viele Einträge gibt es in einer Seitentabelle jeder Stufe?
3. Wie viele Seitentabellen gibt es insgesamt, wenn die Seitentabelle der ersten Stufe die minimale Anzahl von Einträgen hat?
4. Wie viele Seitentabellen gibt es, wenn die Seitentabelle der ersten Stufe nicht die minimale Anzahl von Einträgen hat?

Aufgabe 6:**12 + 10 Punkte**

Ein Prozess ist zum Zeitpunkt $t = 0$ gestartet und ihm sind vier Seitenrahmen zugeteilt. Die folgende Tabelle zeigt die virtuelle Seitennummer, die Seitenrahmennummer, den Zeitpunkt, zu dem eine Seite nach dem Start des Prozesses in die Seitenrahmen geladen worden ist, den Zeitpunkt des letzten Zugriffs, und das Zugriffsbit für jede Seite.

Virtuelle Seite	Seitenrahmen	Ladezeitpunkt	Zugriffszeitpunkt	Zugriffsbit
2	0	60	161	0
1	1	130	160	1
0	2	26	162	1
3	3	20	163	1

1. Zum Zeitpunkt $t = 164$ wird auf die Seite 4 zugegriffen und dabei ein Seitenfehler aufgelöst. Welche Seite wird nach den folgenden Seitenauslagerungsstrategien ausgelagert?

(a) Die optimale Seitenauslagerungsstrategie für die Seitenzugriffsfolge

4, 0, 0, 0, 2, 4, 2, 1, 0, 3, 2

(b) FIFO (first-in-first-out)

(c) LRU (least recently used)

(d) Clock-Algorithmus

2. Bestimmen sie die Zahl der Seitenfehler ab Zeitpunkt $t = 164$, wenn die Seiten nach LRU-Strategie ausgelagert werden und die Seitenzugriffsfolge

4, 0, 0, 0, 2, 4, 2, 1, 0, 3, 2

ist. Stellen Sie die belegten Seitenrahmen in Form einer Tabelle dar.

Aufgabe 7:**16 Punkte**

Das folgende Programm wurde vorgeschlagen, um das Problem des wechselseitigen Ausschlusses zu lösen. Kann ein Deadlock entstehen?

```
(* gemeinsame Variablen *)

blocked : array[0..1] of boolean;
turn : integer;

(* Initialisierung *)

turn := 0; (* wer ist gerade 'dran'? *)
blocked[0] := false;
blocked[1] := false;

Prozess i: (* wird parallel für i=0,1 ausgeführt *)

while (true) do begin
  blocked[i] := true; (* Prozess i bewirbt sich *)
  while (blocked[1-i]) do begin
    if (turn = 1-i) then (* ist der andere Prozess dran? *)
      begin
        blocked[i] := false;
        while (turn = 1-i) do
          blocked[i] := true; (* Prozess i bewirbt sich erneut *)
        end;
      end;
    end;

  kritischer Abschnitt i;

  turn := 1-i; (* lässt den anderen Prozess dran kommen *)
  blocked[i] := false;

  unkritischer Abschnitt;

end;
```

Aufgabe 8:

20 Punkte

Das Kommando `make` ist ein Beispiel für eine Kommandoprozedur unter UNIX. Was ist das Funktionsprinzip von `make`?