Vorlesungsprüfung aus Systemprogrammierung 2002-05-29

KNr. MNr. Zuname, Vorname

Ges.)(100)	1.)(35)	2.)(20)	3.)(25)	4.)(20)	Zusatzblätter:

Bitte verwenden Sie nur dokumentenechtes Schreibmaterial!

1 Synchronisation (35)

In der Produktionsanlage eines Zulieferbetriebs werden Komponenten aus Aluminiumzylindern und Kunststoffringen von Industrierobotern hergestellt. Die Aluminiumzylinder und Kunststoffringe stehen in einem Materiallager bereit. Fertige Teile werden in einem Endlager verstaut. Die Abfolge der Arbeitsschritte bei der Produktion ist in Abbildung 1 veranschaulicht.

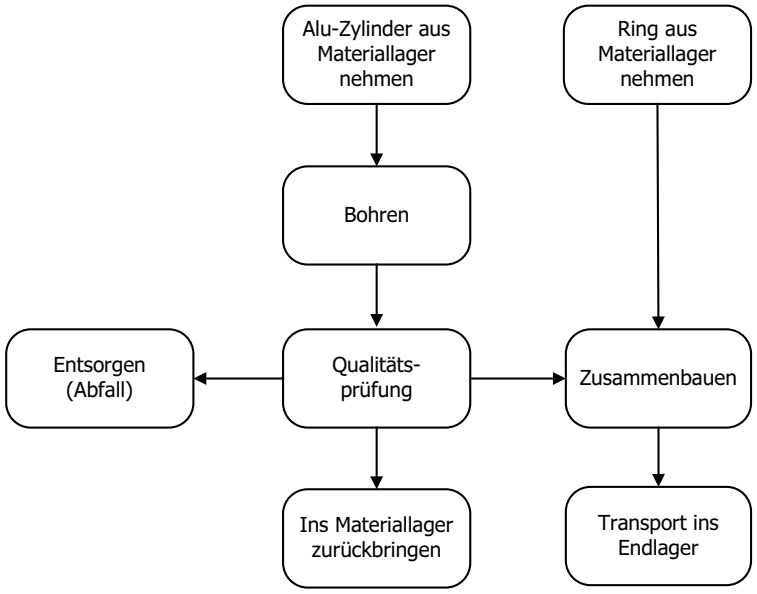


Abbildung 1: Abfolge der Arbeitsschritte

Für die Herstellung wird ein Kunststoffring an einem Aluminiumzylinder angebracht. Hierzu muss ein Aluminiumzylinder aus dem Materiallager geholt und gebohrt werden. Danach wird eine Materialprüfung durchgeführt, die den tatsächlich erzielten Bohrdurchmesser bestimmt. Ist der erzielte Bohrdurchmesser kleiner als 30 mm, so wird der Aluminiumzylinder

ins Materiallager retourniert, um später erneut gebohrt zu werden. Bei einem ermittelten Bohrdurchmesser über 31 mm muss der Aluminiumzylinder entsorgt werden. Nach erfolgreicher Materialprüfung, dh. der Bohrdurchmesser beträgt 30 mm oder 31 mm, kann ein Kunststoffring am Aluminiumzylinder angebracht und das fertige Teil ins Endlager abtransportiert werden.

Zur Durchführung dieser Verarbeitungsschritte stehen Industrieroboter bereit, die jeweils eine bestimmte Teilaufgabe ausführen können. Diese Roboter müssen außerdem Teile an andere Roboter weitergeben bzw. von diesen Teile übernehmen.

- Roboter 1 und Roboter 2 können Teile (Zylinder oder Ring) aus dem Materiallager holen.
- Roboter 3 führt Bohrungen am Zylinder durch und kann Teile ins Materiallager, sowie in die Entsorgungsstätte bringen.
- Roboter 4 ist mit der Qualitätsprüfung betraut und hat fehlerhafte Teile zu entsorgen.
- Roboter 5 kann zwei Teile zusammenbauen und fertige Teile ins Endlager transportieren.

Erstellen Sie die Programme der Industrieroboter und sorgen Sie für eine korrekte Abfolge der Arbeitsschritte durch Synchronisation mit Semaphoren!

Beachten Sie dabei folgende Hinweise:

- Die Verwendung von globalen Variablen bzw. Busy-Waiting zur Synchronisation ist verboten!
- Verhindern Sie das Auftreten von Deadlocks!
- Die Roboter 1, 2, 3 und 4 können zu einem bestimmten Zeitpunkt jeweils nur ein einziges Teil bearbeiten.
- Roboter 5 kann maximal zwei zum Zusammenbau bestimmte Teile aufnehmen.
- Achten Sie auf ein hohes Maß an Parallelismus bei der Verarbeitung der Teile durch die Roboter!
- Verwenden Sie möglichst wenige Semaphore!

a) Erstellen und initialisieren Sie alle benötigten Semaphore. (5)

Zum Anlegen eines Semaphores steht Ihnen das Statement `SemInit` zur Verfügung. Mit diesem kann man sowohl einfache Semaphore (`SemInit(SemaphorName,InitWert)`) als auch Semaphorarrays (`SemInit(SemaphorArrayName[Anzahl],InitWert1,InitWert2,...)`) erzeugen und initialisieren. Das Sperren eines Semaphors erfolgt durch den Aufruf von `P(SemaphorName)` bzw. `P(SemaphorArrayName[Index])`, mit `V(SemaphorName)` bzw. mit `V(SemaphorArrayName[Index])` gibt man es frei!

b) Vervollständigen Sie die Programme der Industrieroboter. (30)

Verwenden Sie hierzu die folgenden Routinen:

- *Nimm(roboter)* nimmt das Teil vom Industrieroboter *roboter* ∈ *{roboter1,...,roboter5}* entgegen.
- *HoleAusLager(obj)* besorgt das Teil *obj* aus dem Materiallager. Diese Aktivität ist Roboter 1 und Roboter 2 vorbehalten.

  (obj ∈ *{Aluzylinder,Kunststoffring}*).
- *Gib(ort)* reicht das Teil zum Standort *ort* weiter, wobei *ort* ein Industrieroboter *(roboter1,...,roboter5)*, das Endlager *(endlager)*, das Materiallager *(materiallager)*, oder die Entsorgungsstätte *(abfall)* ist. Wird das Teil an einen anderen Roboter übergeben, dann darf der Roboter nach dem Aufruf von *Gib(ort)* solange kein weiteres Teil aufnehmen, bis der andere Roboter das weitergereichte Teil mittels eines Aufrufs von *Nimm(roboter)* übernommen hat.
- *Bohren()* veranlasst einen Roboter zur Durchführung einer Bohrung am Teil. Diese Routine kann nur von Roboter 3 ausgeführt werden.
- Die Routine *int BestimmeDurchmesser()* ermittelt den erzielten Bohrdurchmesser in mm. Diese Routine kann nur von Roboter 4 ausgeführt werden.
- Zwei Teile werden mit der Routine *Zusammenbauen()* zu einem Teil verarbeitet. Diese Routine kann nur von Roboter 5 ausgeführt werden.

Roboter 1

Roboter 2

Roboter 3

Roboter 4

Roboter 5

2 Scheduling (20)

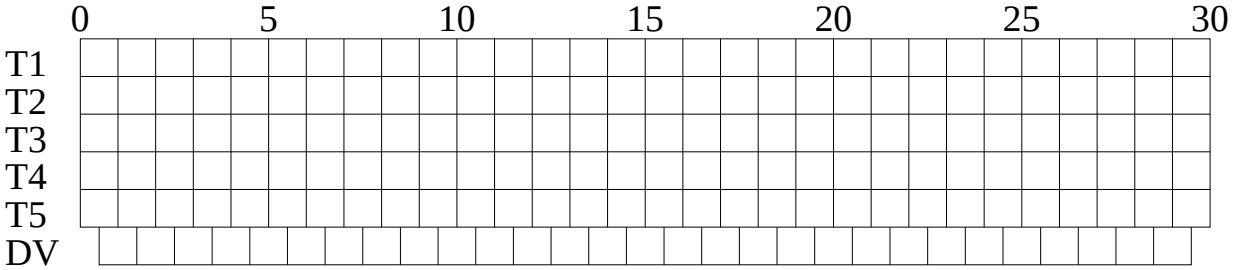
a) Rate-Monotonic Scheduling (10)

Task	Laufzeit (ms)	Periode (ms)
T1	5	12
T2	2	23
T3	2	22
T4	2	11
T5	3	13

Folgendes wird angenommen:

- Alle Tasks sind periodisch.
- Deadline = Periode.
- Laufzeit ist bekannt.
- Der Scheduling Overhead wird vernachlässigt.
- Alle Tasks sind unabhängig.

Dieses Taskset soll mit einem Rate-Montonic Scheduling Algorithmus für den Worst Case[1] gescheduled werden. Bitte vervollständigen Sie die folgende Skizze zur Gänze bzw. bis zur ersten Deadlineverletzung. Kennzeichnen Sie eine solche, indem Sie in der Zeile DV den Task eintragen, für den die Deadline nicht eingehalten wurde.



Bitte beantworten Sie diese Frage unabhängig von Ihrer Lösung oberhalb: Nehmen Sie an, dass das obige Taskset gescheduled werden kann. Bis zu welchem Zeitpunkt müssen Sie das angegebene Taskset *mindestens* schedulen, um sicher gehen zu können, dass es wirklich schedulbar ist. Bitte richtige Antworten ankreuzen:

◯ 11 ms ◯ 12 ms ◯ 13 ms ◯ 22 ms ◯ 23 ms
◯ 24 ms ◯ 26 ms ◯ 44 ms ◯ 46 ms
◯ einen Zeitpunkt der in der Liste nicht aufscheint

[1] Im Worst Case beginnen alle Perioden bei 0.

b) Fragen zu Rate-Monotonic Scheduling (6)

Gegeben seien $n = 4$ Tasks; mit den Ausführungszeiten C_1, C_2, C_3, C_4 und den Perioden T_1, T_2, T_3, T_4; der Scheduling-Overhead wird vernachlässigt:

Welche Bedingung muss erfüllt sein, dass Sie sicher **kein** mögliches Scheduling finden können?

Welche Bedingung müssen die 4 Tasks erfüllen, dass es sicherlich ein mögliches Scheduling gibt?

Können Sie mit Hilfe dieser beiden Bedingungen alle Fälle klassifizieren, oder gibt es Fälle welche sich nicht eindeutig entscheiden lassen?

◯ alle Fälle sind eindeutig entscheidbar
◯ es gibt Fälle, die nicht eindeutig entscheidbar sind

In einem System das RMS verwendet, messen Sie eine Prozessorauslastung von 0,65. Die Taskanzahl ist Ihnen unbekannt; weiters wird der Scheduling-Overhead vernachlässigt. Können Sie sicher sein, dass alle Deadlines eingehalten werden?

◯ ja ◯ nein

Wenn Sie die vorherige Frage mit nein beantwortet haben, geben Sie bitte an, welche Informationen Sie zusätzlich benötigen, um eine sichere Aussage machen zu können. Im Fall, dass Sie mit ja geantwortet haben, erklären Sie bitte, warum diese Information ausreicht?

c) Fragen zu Scheduling allgemein (4)

Kreuzen Sie bitte an ob die Aussagen korrekt sind

Bei First-Come-First-Served (FCFS) Scheduling kann es niemals zu Starvation kommen:

◯ korrekt ◯ inkorrekt

Round Robin (RR) Scheduling erfordert nur geringen Scheduling-Overhead:

◯ korrekt ◯ inkorrekt

Shortest-Remaining-Time (SRT) Scheduling benachteiligt Prozesse mit langen Ausführungszeiten:

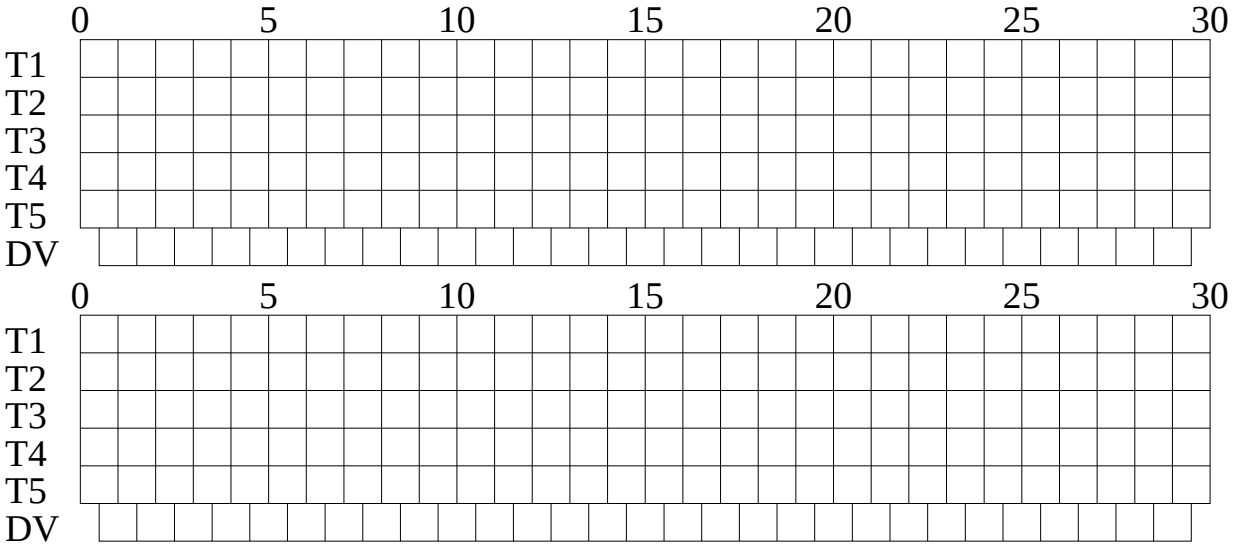
◯ korrekt ◯ inkorrekt

Bei Highest-Response-Ratio-Next (HRRN) Scheduling ist Starvation unmöglich:

◯ korrekt ◯ inkorrekt

Ersatzvorlagen zu a)

Streichen Sie ungültige Vorlagen deutlich durch, wenn Sie eine dieser Vorlagen verwenden!



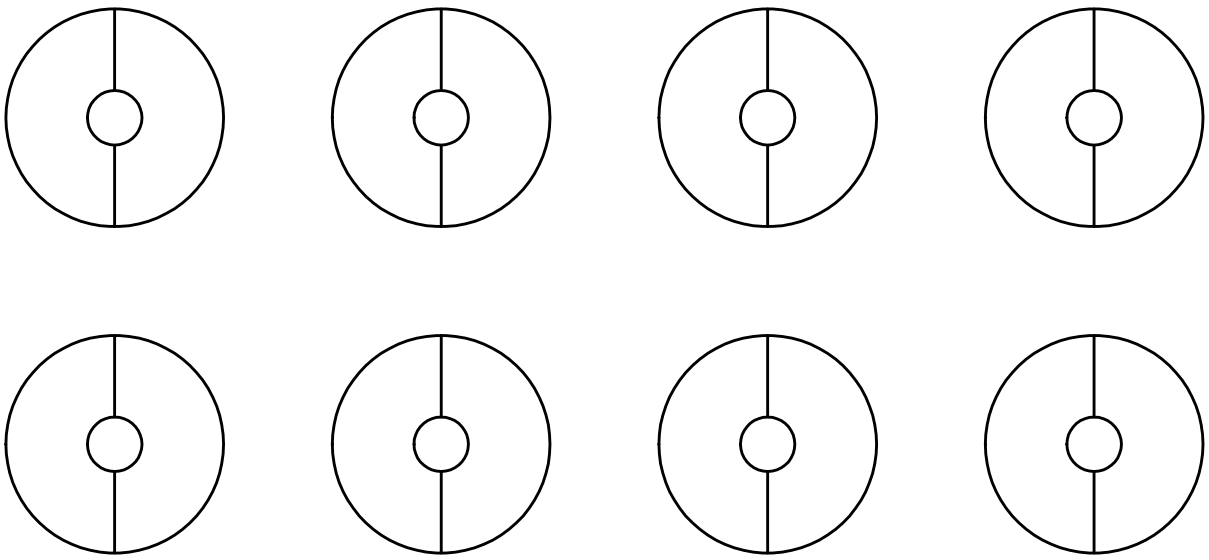
3 Speicherverwaltung - Replacement Policies(25)

a)(9)

Ein Prozess referenziert 5 pages A,B,C,D und E. Wie groß ist die Anzahl der Page Faults, die während dieser Zugriffe auftreten? Nehmen Sie jeweils an, dass der Speicher vor dem Zugriff leer ist.

Access Order	Replacement Policies	Allocation Size in Number of Pages	Number of Page Faults
A;B;C;D;A;B;E;A	First In First Out	3	
A;B;A;B;C;A;D;E	Simple Clock	2	
E;B;A;B;C;A;C;B	Least Recently Used	3	

Für etwaige Skizzen zur Simple Clock Policy verwenden Sie bitte folgende Vorlagen:



b)(16)

Ein Prozess hat 4 Page Frames alloziert. Die Allocation Size in Number of Pages ist 4. Folgende Abkürzungen werden verwendet:

TL..time loaded: Der Zeitpunkt als die Page das letzte Mal geladen wurde

TR..time reference: Der Zeitpunkt als auf die Page das letze Mal zugegriffen wurde

Die angegebenen Zeiten sind die Ticks vom Zeitpunkt des Prozessstarts (0) weggezählt.

Virtual Page Number	Page Frame	TL	TR	Clock Use Bit
2	0	60	161	1
1	1	130	160	1
0	2	26	162	1
3	3	20	163	1

Ein Page Fault der virtuellen Page 4 ist aufgetreten. Welcher Page Frame wird ausgetauscht werden, wenn die folgende Memory Managment Policy verwendet wird?

Policy	Page Frame	Warum wird der Page Frame ausgetauscht werden?
First In First Out		
Least Recently Used		
Simple Clock*		
Optimal**		

*Der Frame Allocation Pointer wird den Page Frame Nummern entsprechend weitergesetzt, die Sequenz ist also: 0,1,2,3,0,....

**Zukünftige Access Sequence nach der aktuellen Virtual Page 4: 3,0,1

4 Prozessverwaltung und I/O (20)

a) (6)

Erklären Sie die beiden Begriffe *Mode Switch* und *Process Switch*. Was sind die Aufgaben jedes dieser Switches und wie stehen sie in Beziehung? Geben Sie für jeden der beiden Switches an, in welchem Execution Mode er abgearbeitet wird.

b) (4)

Nennen Sie die drei grundlegenden Mechanismen, mit denen die Kontrolle von einem laufenden Prozess an das Betriebssystem übergeht.

c) (10)

Geben Sie die Schritte an, die die Prozessverwaltung eines Multiprocessing-Betriebssystems im Rahmen einer blockierenden I/O-Operation (z.B. Lesen eines Zeichens von der Tastatur) ausführt. Nehmen Sie an, dass während dieser I/O Operation genau ein anderer Prozess zum Laufen kommt. Geben Sie für jeden Schritt an, in welchem Execution Mode des Betriebssystems er abläuft.