

Parallelprogrammierung - Komplex 1

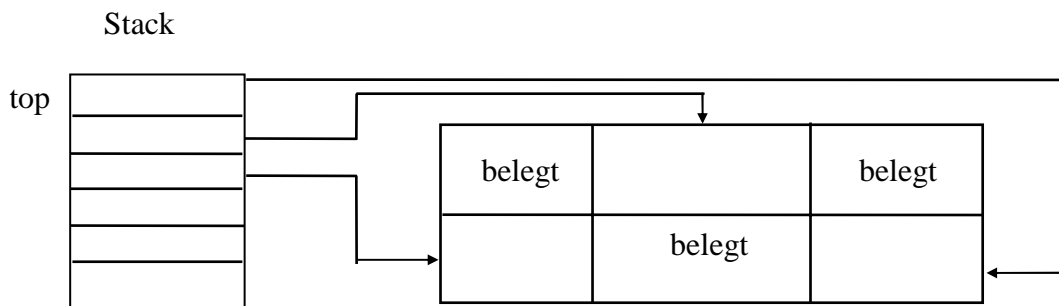
1. Parallele Verarbeitung existiert auf unterschiedlichen Abstraktionsebenen. Nennen Sie diese! Auf welchen Abstraktionsebenen ist der Programmierer gefordert?
2. Wodurch unterscheiden sich asynchrone und synchrone Parallelität?
3. Wodurch sind nebenläufige Prozesse gekennzeichnet? Was sind parallele Prozesse?
4. Was versteht man im Zusammenhang mit paralleler Verarbeitung unter den Begriffen Kooperation, Konkurrenz, Synchronisation und Kommunikation?
5. Nennen Sie Konzepte für die ablauforientierte Prozeßerzeugung und geben Sie eine Kurzerklärung!
6. Wie werden in PEARL, ADA und Concurrent PASCAL parallele Prozesse erzeugt und was verwenden diese Sprachen als Synchronisationskonzept?
7. Wozu dienen Prozesskontrollblöcke und welche Art von Informationen enthalten sie?
8. Nennen und erläutern Sie Ihnen bekannte Problemabstraktionen der Parallelprogrammierung!
9. Was verstehen Sie unter dem „kritischen Abschnitt“ eines nebenläufigen Prozesses und wodurch ist dieser charakterisiert?
10. Was verstehen Sie unter Sicherheit und Lebendigkeit von parallelen Programmen?
11. Wann ist ein paralleles Programmsystem verklemmungsgefährdet und welche Gegenmaßnahmen können getroffen werden?
12. In einem Programmsystem existieren k Erzeuger desselben Typs, die ein Datenkontingent erzeugen. Diese Daten können beliebig von n Verbrauchern übernommen werden. Stellen Sie dieses Erzeuger-Verbraucher-Problem in einem Petri-Netz dar, wenn zum Betrachtungszeitpunkt $t = 0$ eine Menge von p Datenkontingenten erzeugt worden ist!
13. In einem Programmsystem sind n nebenläufige Prozesse. Diese können lesend oder schreibend auf einen Speicherbereich zugreifen. Folgende Bedingungen sollen gelten:

Wenn kein Schreiber aktiv ist, können k Prozesse lesen. Ein Schreibprozeß darf nur aktiviert werden, wenn kein anderer Schreib- oder Lese-Prozess aktiv ist.

Beschreiben Sie das Problem als Petri-Netz!

Parallelprogrammierung - Komplex 2

1. Wie wird beim Algorithmus nach Dekker der gegenseitige Ausschluß realisiert? Wann findet der Algorithmus nach Dekker Anwendung?
2. Erläutern Sie den Begriff Semaphor! Skizzieren Sie die Eigenschaften von Semaphoren und die Arbeitsweise mit Semaphoren! Welche Arten von Semaphoren gibt es?
3. Geben Sie das Prinzip für die algorithmische Implementierung der P- und V-Operation nach Arbeitsblatt 5.1. für Integer- Semaphoren an!
4. Untersuchen Sie die Teilbarkeit der P- und V-Operationen! Welche Nachteile birgt das Semaphorkonzept?
5. Gegeben ist ein Pufferblock mit n Blöcken gleicher Länge. Ein Prozess kann mit der Operation/Prozedur **getblock** (gibt Blockadresse als Pointer zurück) einen Block verwenden. Mit der Operation/Prozedur **release(adr)** (Adresse Pointer) wird der Block wieder freigegeben:



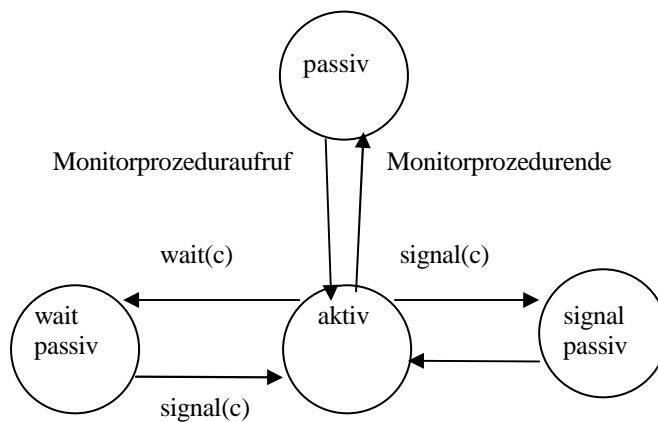
Realisieren Sie die Operationen **getblock** und **release** als unabhängige Prozesse in PASCAL-ähnlicher Notation. Achten Sie darauf, dass bei Belegung aller Blöcke eine weitere Blockanforderung nicht zu einem Fehler führt!

6. Synchronisieren Sie die im Bild gezeigte Erzeuger-Verbraucher-Problematik: Achten Sie auf sehr kurze kritische Abschnitte!



Parallelprogrammierung - Komplex 3

1. Welche Zielstellung wird mit der Abstraktion von Semaphoren verfolgt? Nennen Sie Ihnen bekannte Abstraktionen von Semaphoren!
2. Erklären Sie die Wirkungsweise additiver Semaphoren und ihren Anwendungsbereich!
3. Erklären Sie die Arbeitsweise von Mehrfachsemaphoroperationen und ihren Anwendungsbereich!
4. Welche Möglichkeiten bietet die Anwendung von Pfadausdrücken für die Parallelprogrammierung? Was kann durch Pfadausdrücke und Pfadoperatoren notiert werden?
5. Erläutern Sie die Grundidee des Monitorkonzepts und den Aufbau eines Monitors!
6. Innerhalb einer Monitorprozedur können s.g. Signalvariable (Bedingungsvariable, *condition*) und die darauf arbeitenden Operationen *signal(condition)* und *wait(condition)* verwendet werden. Erläutern Sie Anwendungsbereich und Wirkungsweise der Signalvariablen!
7. Erklären Sie das nachfolgende Zustandsmodell eines Monitors. Wie viele Warteschlangen sind für die Implementierung eines Monitors notwendig?



8. Das Monitorkonzept besitze eine weitere Stammfunktion, die *noempty(condition)*. Erläutern Sie ihre Anwendung!
9. Modifizieren Sie das Leser-Schreiber-Beispiel auf den Bild 7.6.,7.7 so, das Schreiber die höhere Priorität zur Anmeldung eines Schreibwunsches und zum Schreiben besitzen.
10. Implementieren Sie ein Ereignisflag unter Verwendung eines Monitors!

Parallelprogrammierung - Komplex 4

1. Welche Merkmale besitzen Operationen zum Senden und Empfangen?
2. Erläutern Sie Anwendungsbereiche und Merkmale des selektiven Senden und Empfangen!
3. Was versteht man unter einem Prozedurfernaufruf (RPC) und welche Möglichkeiten zur Kommunikation von parallelen Prozesse werden durch einen RPC ermöglicht?
Wie erfolgt ihre prinzipielle Implementierung?
4. Erläutern Sie die Semantik der ACCEPT- und der SELECT-Anweisung von ADA zur Anwendung des Rendezvouskonzeptes auf Hochsprachebene!
5. Entwerfen Sie eine Rendezvous-Task, durch die ein beliebiges Betriebsmittel belegt bzw. freigegeben wird. Verwenden Sie eine ADA-ähnliche Notation!
6. Erstellen Sie ein Programm (ADA) zur Lautstärkeregelung in Analogie zum Algorithmus in der Sprache OCCAM (Bild 8.7.)! Der direkte Zugriff auf den Verstärkereingang erfolgt als Dienst durch ein anderes Programm im System.
7. Entwerfen Sie eine Rendezvous-Task zur Ansteuerung eines Transportantriebes unter Verwendung der delay-Anweisung (ADA). Der Antrieb soll solange aktiv bleiben, bis keine Transportaufträge mehr vorliegen. Maximal 10 sec später ist der Transport auszuschalten