Übungen zur Vorlesung Einführung in die Informatik III

Aufgabe 35 Synchronisation mit busy waiting auf der MI

Wir betrachten eine Erweiterung der fiktiven Rechenmaschine MI mit vier Rechnerkernen und einem gemeinsamen Speicher. (Einen MI-Simulator mit vier Rechnerkernen können Sie auf der sunhalle mit dem Befehl `Xmi4` starten)

Auf dieser soll ein Programmstück *P* als kritischer Bereich über busy waiting realisiert werden.

(a) Schreiben Sie dazu ein MI-Programm, das falls es auf allen vier Rechnerkernen gestartet wird, sicherstellt, dass der kritische Bereich *P* unter gegenseitigem Ausschluss ausgeführt wird.

(b) Ändern Sie das Programm so, dass falls das Programm auf allen vier Rechnerkernen gestartet wird, der kritische Bereich *P* zuerst auf dem Rechnerkern K3, dann K2, dann K1, dann K0 ausgeführt wird.

Hinweis: Sie benötigen den Befehl `SPRKMAP` (MI-Handbuch, S. 31), um die Nummer des Rechnerkerns zu erhalten, auf dem der Prozess gerade ausgeführt wird.

Aufgabe 36 Speichersegmente

Im folgenden sollen die Inhalte von *k* Feldern (*Segmenten*) in einem Speicher

[1:n]**array var m** sp

repräsentiert werden. Die Anzahl der Segmente (maximal *maxseg*) und die Länge der einzelnen Segmente kann sich dynamisch mit Hilfe folgender Operationen ändern:

sort snr = 1:maxseg, **sort adr** = 1:n	
func newSegment = **snr**:	Anlegen eines neuen Segments (Länge 0)
proc setSegmentSize = (**snr** s, **nat** sz):	Setzt die Größe von Segment s auf sz
proc removeSegment = (**snr** s):	Lösche das Segment s
proc setValue = (**snr** s, **adr** i, **m** v):	Setze den i-ten Wert des Segments s auf v
func getValue = (**snr** s, **adr** i)**m**:	Lese i-ten Wert aus dem Segment s

a) Skizzieren Sie eine Implementierung dieser Operationen, die durch Verschiebung von Segmenten den Speicherplatz optimal ausnutzt. Zeigen Sie, dass bei n = 20, maxseg = 3, die folgende Sequenz von Anforderungen ausgeführt werden kann:

1) Kreieren dreier Segmente
2) Setzen der Größe Segment 1 auf 5
3) Setzen der Größe Segment 2 auf 5

4) Setzen der Größe Segment 3 auf 5
5) Setzen der Größe Segment 1 auf 8
6) Setzen der Größe Segment 2 auf 7
7) Löschen von Segment 1
8) Setzen der Größe Segment 3 auf 8
9) Setzen der Größe Segment 1 auf 2

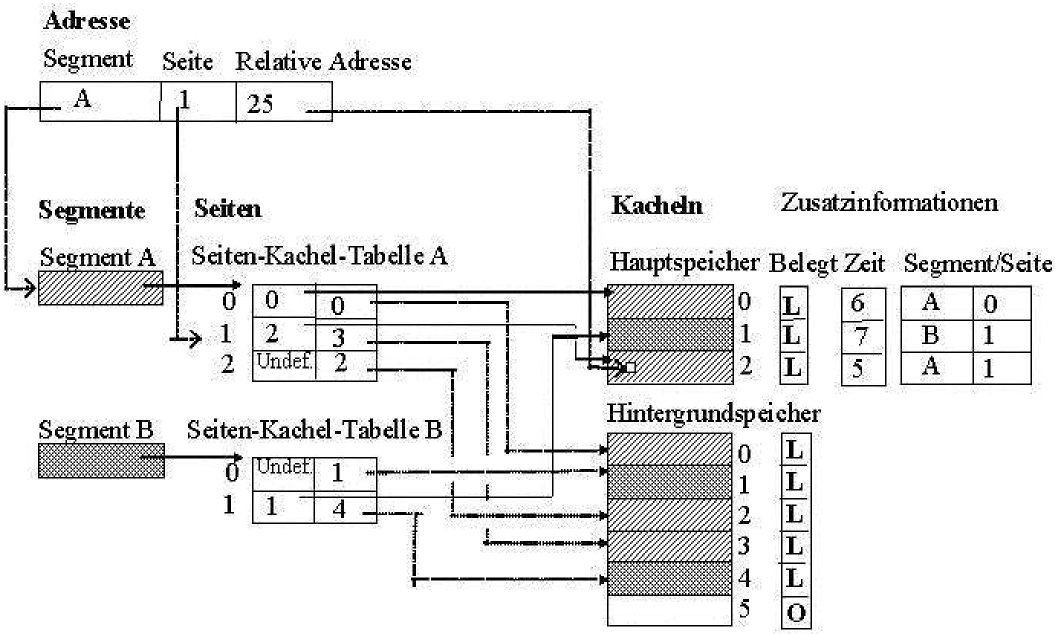
b) Geben Sie eine Implementierung der obigen Funktionen und Prozeduren an, die ohne Verschieben auskommt und dennoch diese Anforderungen erfüllen kann.

c) Wie groß ist bei maxseg = 3, n = 1000 der zusätzliche organisatorische Aufwand pro Speicherelement in sp,

(i) falls die Elemente **m** ein Byte lang sind;
(ii) falls die Elemente **m** jeweils 256 Byte lang sind?

Aufgabe 37 Seitenaustauschverfahren

In dieser Aufgabe wird die Speichersegmentierung aus Aufgabe 36 mit einem Seitenaustauschverfahren kombiniert. Gegeben sei ein Hauptspeicher (auch Arbeitsspeicher – AS) bestehend aus 3 Kacheln und ein Hintergrundspeicher (HGS) mit 6 Kacheln. Die Kacheln sind nach der Kachelnummer sortiert und haben eine Größe von 100 Zellen.



Die Seiten-Kachel-Tabelle enthält die Kachelnummer des HGS und ggf. die Kachelnummer des AS. Wird in „setSegmentSize" eine neue Kachel benötigt, dann wird die erste freie Kachel des AS genommen. Wenn die AS-Kachel-Nr. nicht definiert ist, wird bei „setValue" und „getValue" die Kachel vom HGS in den AS geholt. Dabei wird die erste freie AS-Kachel verwendet. Ist keine freie Kachel vorhanden, dann muß eine Kachel des AS verdrängt werden. Z.B. kann die am längsten unbenutzte Kachel in den HGS ausgelagert werden.

a) Skizzieren Sie den Zustand von AS und HS nach folgenden Aktionen

1) Segment 0 anlegen, um eine Seite erweitern und auf Seite 0 schreiben.

2) Segment 1 anlegen, um zwei Seiten erweitern und auf Seite 1 schreiben.
3) Segment 0 anlegen, um eine Seite erweitern und auf Seite 1 schreiben.
4) Segment 1 um eine Seite verkürzen
5) Segment 0 um eine Seite erweitern, auf Seite 2 schreiben und von Seite 1 und 0 lesen
6) Segment 1 um eine Seite erweitern und auf Seite 1 schreiben.

b) Skizzieren Sie die Implementierung der Speicher-Operationen aus Aufgabe 36.

c) Nennen Sie Vor- und Nachteile des Seitenaustauschverfahrens.

Aufgabe 38 (P) Segmentierung und Seitenaustauschverfahren — Java (2 Wochen)

Implementieren Sie in Java eine Klasse `VirtualMemory`, die die in Aufgabe 37 beschriebene Funktionalität realisiert und Thread-safe ist:

- Stellen Sie den Hauptspeicher durch ein Array von `byte`-Arrays dar. Ein `byte`-Array entspricht dabei einer Kachel.
- Verwenden Sie für den Hintergrundspeicher die Klasse `java.io.RandomAccessFile`. Fangen Sie Ausnahmen der Klasse `java.io.IOException` gegebenenfalls ab und erzeugen Sie statt dessen Ausnahmen der Klasse `java.lang.RuntimeException`.
- Verwenden Sie für die Segment-Tabelle und die Seiten-Kachel-Tabellen die Klasse `java.util.Vector`.
- Verwenden Sie für einen Eintrag in der Seiten-Kachel-Tabelle ein `int[2]` Array. Das Feld `[0]` enthält die Kachelnummer im Hauptspeicher oder `-1` (undefiniert) und `[1]` enthält die Kachel-Nr. im File.
- Erstellen Sie einen Konstruktor mit folgender Schnittstelle:

  ```
  public VirtualMemory( int pageSize, int memoryPageCount,
  int filePageCount, String pageFileName )throws IOException
  ```
- Erzeugen Sie beim Laden und Auslagern einer Kachel folgende Ausgaben:

  `Load file page` »File-Kachel-Nr.« `to memory page` »Speicher-Kachel-Nr.« `Write memory page` » Speicher-Kachel-Nr.« `to file page` »File-Kachel-Nr.«
- Erzeugen Sie nach dem Anlegen einer Seite folgende Ausgabe:

  `Added page` » Seiten-Nr.« `in file page` »File-Kachel-Nr.«
- Erzeugen Sie beim Lesen und Schreiben eines Wertes in eine Seite folgende Ausgabe:

  `Segment` » Segment-Nr.« `uses page` » Seiten-Nr.« `in memory page` » Speicher-Kachel-Nr.« `at` » Zeitpunkt «
- Schreiben Sie ein Programm, das die Aktionen aus Aufgabe 37a) ausführt und am Ende als Aktion (7) das in Aktion (5) im Segment 0 auf der Seite 2 geschriebene `byte` noch einmal liest. Erzeugen Sie beim Schreiben und Lesen dieses Wertes die Ausgabe:

  `Value:` » Wert der Zelle «

  Erzeugen Sie für jede Aktion die Ausgabe:

  `\nAction: (` » Nr. der Aktion « `)`

  Die Seitengröße und die Anzahl der File- und Speicherkacheln sind wie in Aufgabe 1 vorgegeben.

- Schicken Sie ein Ausgabeprotokoll und die Sourcen per Email an Ihren Tutor.