

Klausur zur Vorlesung Betriebssysteme
Universität Paderborn, WS 1999/2000
Prof. Dr. H.-U. Heiß
14. Februar 2000

Name, Vorname	Musterlösung
Matrikel-Nr.	
Fachbereich	
Studiengang	

Wichtige Hinweise!

- Schreiben Sie zunächst sofort alle persönlichen Angaben oben auf das Deckblatt in die vorgesehenen Felder!
- Lassen Sie die Klausur **zusammengeheftet!**
- Die Klausur dauert **75 Minuten** und umfasst **7 Aufgaben** auf 9 Seiten.
- Die Klausur ist bestanden, wenn mindestens 50 % der Punkte erreicht wurden.
- Es sind **keine Hilfsmittel** zugelassen!
- Abschreiben und abschreiben lassen führt zu Nichtbestehen der Klausur!
- Benutzen Sie kein eigenes Konzeptpapier bzw. Schmierpapier. Sie bekommen bei Bedarf Papier von der Klausuraufsicht!
- Wenn Sie Konzeptpapier abgeben wollen, dann **nur mit Namen und Matrikelnummer!**
- **Kennzeichnen Sie Ihre Lösungen eindeutig.** Es wird keine Lösung gewertet, wenn Sie zu einer Aufgabe mehr als eine Lösung abgeben.

Viel Erfolg!

Aufgabe	1	2	3	4	5	6	7	Σ
Punkte	10	10	16	10	12	8	14	80
Erreicht								
Bestanden (J/N)				Note				

Aufgabe 1**Verständnisfragen****10 Pkt.**

(a) Nennen Sie die vier Schichten eines Betriebssystemkerns aus der Vorlesung:

- (1) Kernoperationen.
- (2) Prozesswechseloperationen.
- (3) Datenstrukturoperationen.
- (4) Kern-Speicherverwaltung.

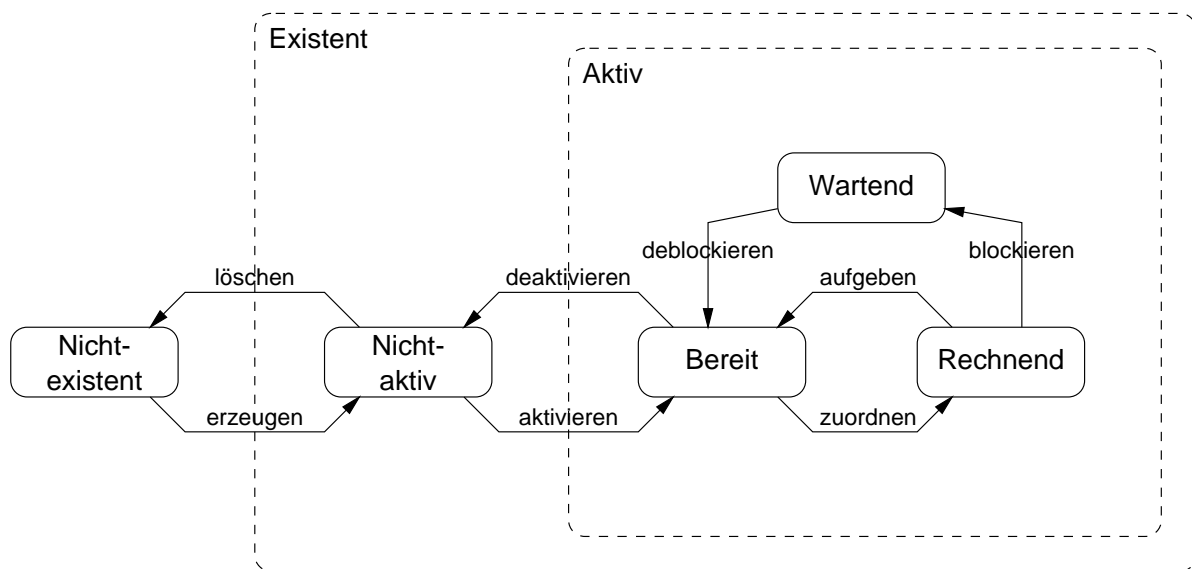
(b) Markieren Sie die folgenden Aussagen als **wahr** oder **falsch**:

- (1) Prozesse, die verzahnt auf dem Zentralprozessor eines Einprozessorsystems ausgeführt werden, sind nebenläufig. **wahr** **falsch**
- (2) Speicherplatz, der aus Sicht der Speicherverwaltung frei ist, aber im Rahmen einer Belegeoperation nicht belegt werden kann, zählt zum externen Verschnitt. **wahr** **falsch**
- (3) Um zu verhindern, dass Kernoperationen auf inkonsistenten Daten arbeiten, gibt es nur die Möglichkeit, Kernoperationen unter gegenseitig Ausschluss zu stellen. **wahr** **falsch**
- (4) In Multitasking-Betriebssystemen wird ein Stapelspeicher eingesetzt, dessen Größe sich dynamisch mit der Anzahl der Prozesse ändert. **wahr** **falsch**
- (5) Es wird immer dann eine Verwaltung von Betriebsmitteln benötigt, wenn mehrere Prozesse um ein BM konkurrieren. **wahr** **falsch**
- (6) Dienstgeber, die ihre Aufträge im Fließband (*Pipelining*) bearbeiten, benötigen zwischen den einzelnen Bearbeitungsphasen innere Kanäle, die als Zwischenlager dienen. **wahr** **falsch**

Aufgabe 2**Prozesszustände****10 Pkt.**

Ein Prozess befindet sich im Verlaufe seines Lebenszyklus in verschiedenen Zuständen.

- (a) Tragen Sie die Ihnen aus der Vorlesung bekannten Prozesszustände in die untere Abbildung ein. Achten Sie darauf, dass jeder Zustand in den korrekten Teilbereich einsortiert wird. Notieren Sie in jedem Zustand dessen Bezeichnung.
- (b) Ergänzen Sie das Diagramm um Zustandsübergänge, die Sie ebenfalls mit den Bezeichnern aus der Vorlesung beschriften.



Aufgabe 3**Prozessinteraktion****16 Pkt.**

Es sei eine spezielle Signalisierungsvariante gegeben, bei der durch jedes einkommende Signal *genau* zwei von beliebig vielen wartenden Prozessen fortgesetzt werden. Es handelt sich also um eine leichte Abwandlung der Form:

$n : 1$ - Einzel : UND_2 - Signalisierung mit Barrierensynchronisation.

Fügen Sie in dem hier angegebenen Implementierungsrahmen die nötigen Anweisungen (im Stil der Vorlesung) ein:

```
kernel module n:1-Single:AND2-Barrier;
  export SIGNAL, WAIT;
  import DEBLOCK, BLOCK;
  var signal_object =
    record
      S : integer = 0      // Anzahl Signale
      WP: queue of process = empty
      WN: integer = 0     // Anzahl wartender Prozesse
    end;

  procedure SIGNAL(SO: signal_object);
  begin
    if SO.WN >= 2 then
      DEBLOCK(SO.WP);
      DEBLOCK(SO.WP);
      SO.WN = SO.WN - 2;
    else
      SO.S = SO.S + 1;
    endif
  end;

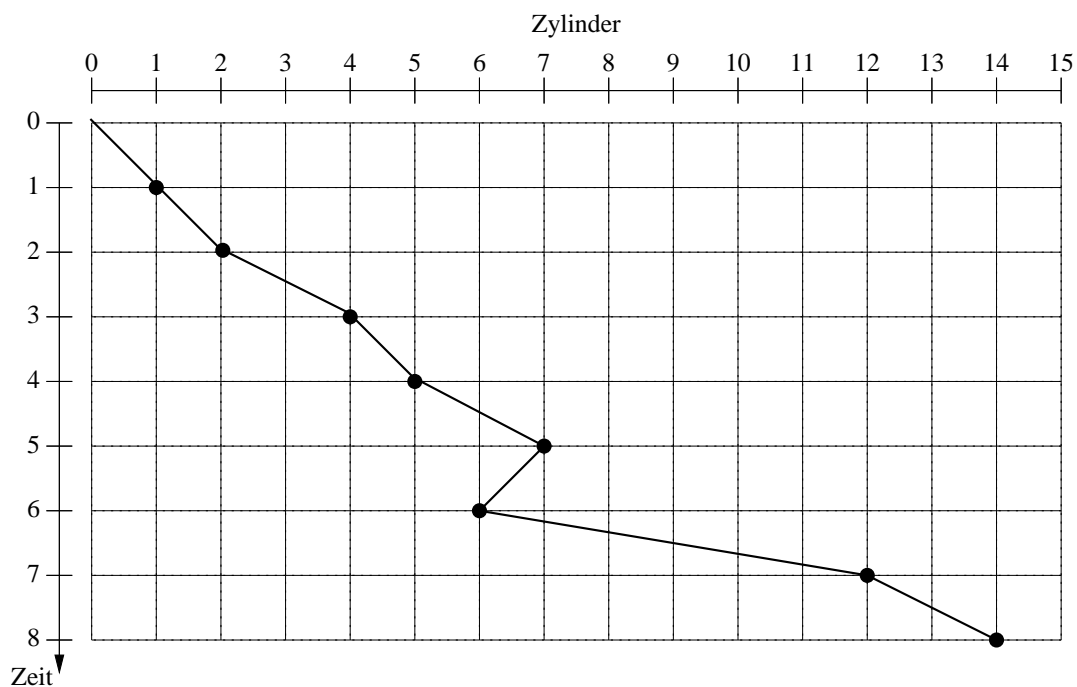
  procedure WAIT(SO: signal_object);
  begin
    if SO.S > 0 and SO.WN = 1 then
      DEBLOCK(SO.WP);
      SO.WN = 0;
      SO.S = SO.S - 1;
    else
      SO.WN = SO.WN + 1;
      BLOCK(SO.WP);
    endif
  end;
end module.
```

Aufgabe 4**Plattentreiber****10 Pkt.**

Im Plattentreiber können die Aufträge entweder in der Ankunftsreihenfolge oder nach ihrer Zylinder Nummer sortiert bearbeitet werden. Zum Zeitpunkt $t = 0$ liegen die folgenden Aufträge vor (von links nach rechts in Ankunftsreihenfolge, wobei die Zahlen die Zylinder Nummern angeben, auf die zugegriffen werden soll): **2, 5, 1, 12, 7, 14**. Zum Zeitpunkt $t = 2$ sei zusätzlich ein weiterer Auftrag für Zylinder Nummer **4** und zum Zeitpunkt $t = 5$ ein Auftrag für Zylinder Nummer **6** eingegangen. (Zu Beginn ($t = 0$) befinde sich der Plattenarm über Zylinder 0.)

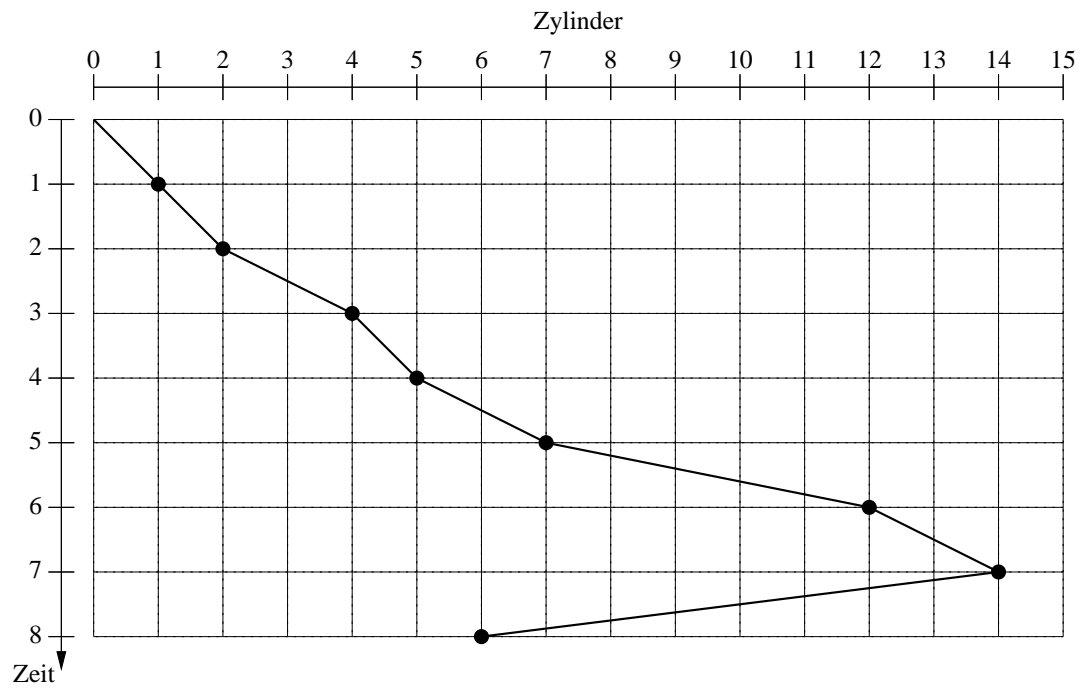
Zeichnen Sie nun in die nachfolgenden Weg-Zeit-Diagramme ein, welcher Auftrag zu welchem Zeitpunkt unter der jeweiligen Strategie ausgeführt wird. Zu jedem Zeitpunkt, beginnend mit $t = 1$, werde ein Auftrag ausgeführt. Berücksichtigen Sie, daß ein Auftrag spätestens zum Zeitpunkt $t = i$ vorliegen muß, um zum Zeitpunkt $t = i + 1$ abgearbeitet werden zu können. Das Diagramm heißt Weg-Zeit-Diagramm, weil es angibt, welchen Weg der Plattenarm zurücklegt.

(a) Beim Shortest-seek-first-Algorithmus (SSF):



Aufträge: $t = 0 : \{2, 5, 1, 12, 7, 14\} / t = 2 : \{4\} / t = 5 : \{6\}$

(b) Beim Aufzugalgorithmus (SCAN):



Aufträge: $t = 0 : \{2, 5, 1, 12, 7, 14\} / t = 2 : \{4\} / t = 5 : \{6\}$

Aufgabe 5

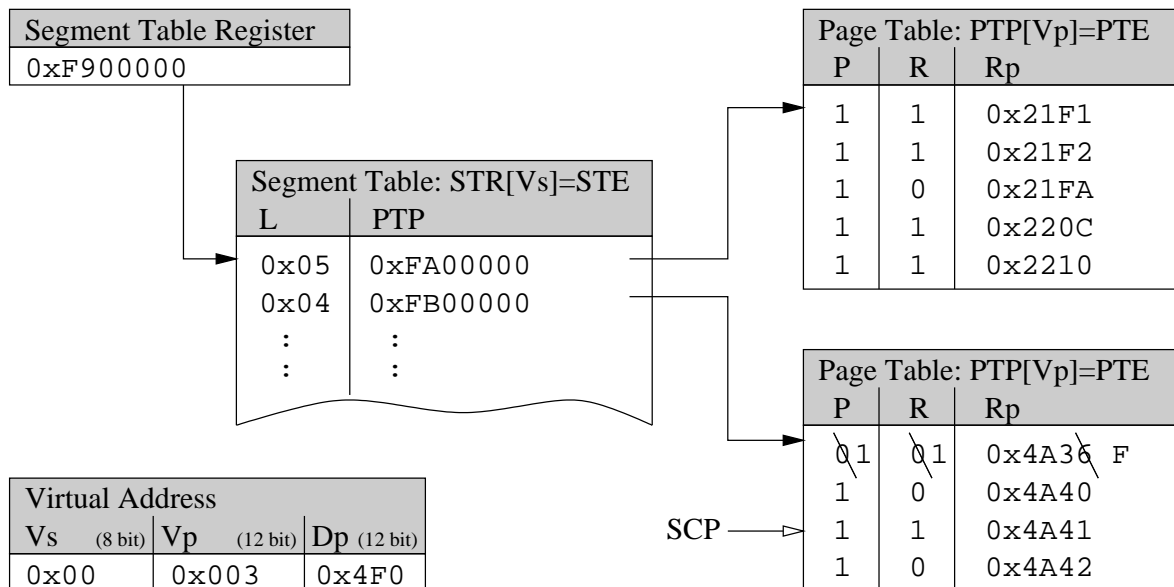
Virtueller Speicher

12 Pkt.

Die Abbildung zeigt einen Ausschnitt aus der aktuellen Konfiguration eines zweistufigen virtuellen Speichers mit den folgenden Kenngrößen:

Seitengröße = 4 kByte, max. Segmentgröße = 16 MByte, max. physikalischer Speicher = 256 MByte, max. virtueller Speicher = 4 GByte.

- (a) Wie lang ist das Feld D_p der virtuellen Adresse in Bits? 12 Bit..
- (b) Welche reale Speicherstelle gehört zur virtuellen Adresse $0x34F0$? $0x220C$ | $4F0$.
- (c) Die Daten der Seite $0x4A36$ werden ausgelagert. Wie lautet die virtuelle Adresse, an der die Seite beginnt? $0x1$ | 000 | 000 .
- (d) Die ausgelagerten Daten werden benötigt und auf der Seite $0x4A3F$ eingelagert. Ändern Sie in der Abbildung die Seiten- und Segmenttabellen entsprechend.
- (e) Anschliessend tritt ein Seitenfehler auf. Nehmen Sie an, dass der *Second-Chance-Algorithmus* eingesetzt wird. Welche Seite wird für die nächste Auslagerung gewählt? $0x4A42$.



Legende:

Vs: Virtual Segment	STR: Segment Table Register	PTE: Page Table Entry
Vp: Virtual Page	STE: Segment Table Entry	P: Presence Bit
Dp: Displacement in Page	L: Length (# pages)	R: Reference/Read Bit
SCP: Second Chance Pointer	PTP: Page Table Pointer	Rp: Real Page

Aufgabe 6**Working Set****8 Pkt.**

Stimmt die folgende Aussage über die Größe des Workingsets eines beliebigen Prozesses?

$$0 < \tau_1 < \tau_2 < t \implies w(t, \tau_1) \leq w(t, \tau_2)$$

Finden Sie einen Referenzstring als Gegenbeispiel oder skizzieren Sie einen Beweis.

Lösung:

Die Aussage ist richtig.

Der Working-Set $W(t, \tau)$ ist definiert als die Menge aller Seiten, auf die der Prozess im Intervall $[t - \tau, t)$ zugegriffen hat. Wird also das Zeitfenster τ größer gemacht, so kann der Working-Set nur größer werden.

Seien T und T' zwei nichtleere Zeitintervalle mit $T' \subset T$. Dann gilt $W(T') \subseteq W(T)$, denn jede in T' referenzierte Seite muss auch in T referenziert worden sein.

Damit gilt aber auch für die Working-Set-Größen: $w(T') = |W(T')| \leq |W(T)| = w(T)$.

Mit $T' := [t - \tau_1)$ und $T := [t - \tau_2)$ erhalten wir den genannten Zusammenhang.

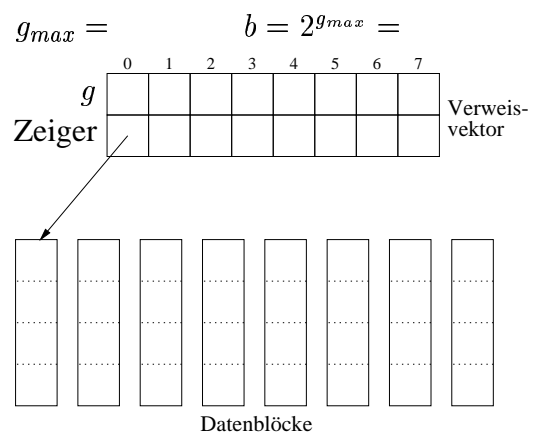
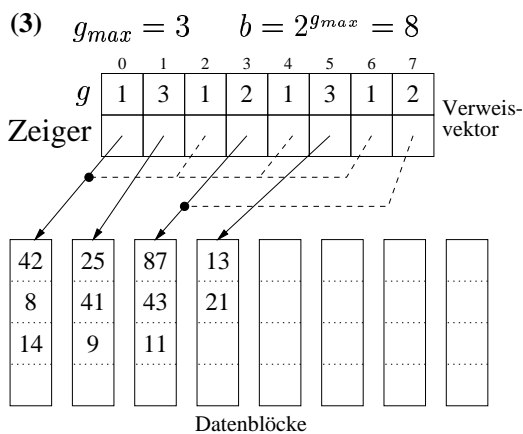
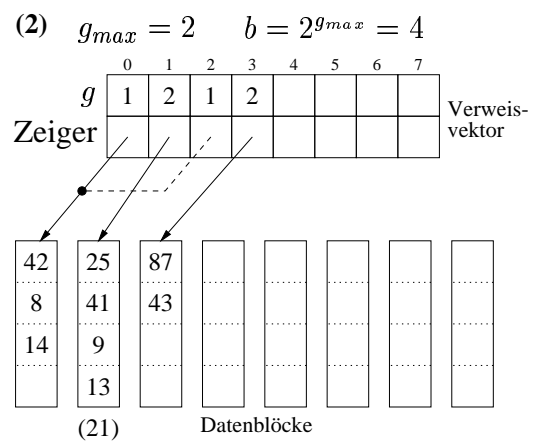
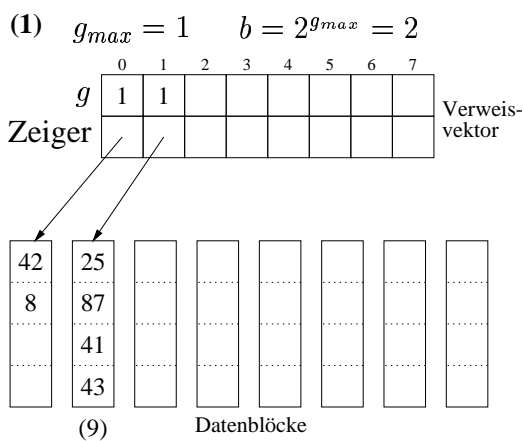
Aufgabe 7

Dateiverwaltung

14 Pkt.

Ein Dateisystem unterliege einer direkten Dateiorganisation mit einem erweiterten Hashing. Beim Anlegen einer neuen Datei werden nacheinander die Sätze mit den Schlüsseln 25, 42, 87, 41, 43, 8, 9, 13, 14, 21, 11 der Datei zugeordnet. Tragen Sie in der folgenden Darstellung jeweils die Werte für g_{max} und b , die Anzahl der g Stellen, nach denen diskriminiert wurde, die Verzeigerung zu den Datenblöcken, sowie die Schlüssel in die entsprechenden Datenblöcke ein.

Zu Beginn sei $g_{max} = 1$. Ein Datenblock kann maximal 4 Sätze aufnehmen. Verwenden Sie jeweils ein neues Bild wenn eine Erweiterung des Hashing notwendig wird!



Schlüsselfolge: [25, 42, 87, 41, 43, 8], 9, 13, 14, 21, 11.