

# IT-Center Dortmund GmbH

**Sommersemester 2001**

## **Übungsklausur zur Vorlesung**

**Betriebssysteme**

Dortmund, den 28.06.2001

Name \_\_\_\_\_

Vorname \_\_\_\_\_

Dauer: 90 Minuten

Hilfsmittel: keine

Hinweise:

- Tragen Sie bitte zunächst auf jedem Blatt Ihre Daten ein!
- Verwenden Sie für Ihre Lösungen ausschließlich die vorhandenen Blätter.
- Lassen Sie alle Blätter im zusammengehefteten Zustand.

Aufgaben	1	2	3	4	5	Summe	Zensur
Punkte	20	20	20	20	20	100	
Ergebnis							

Name: \_\_\_\_\_ Vorname: \_\_\_\_\_

Aufgabe 1: (20 Punkte) Prozeßmanagement

- a) Welcher Unterschied besteht zwischen einem Programm und einem Prozeß ?
- b) Welche Informationen muß ein Process Control Block (PCB) enthalten und wofür werden diese Informationen benötigt?
- c) Welche Programme kommen bei der Erstellung eines Prozeßimages ausgehend von einer Datei mit Programmcode, z.B. C, C++ oder Pascal Code, zum Einsatz und welche Aufgaben erfüllen diese Programme ?

Lösung Aufgabe 1:

Aufgabe 2: (20 Punkte) Wechselseitiger Ausschluß

Sei ein paralleles Programm zur Implementierung eines Produzenten/Konsumenten Verhältnisses durch den folgenden Pseudocode gegeben. Produzent und Konsument kommunizieren über eine gemeinsame Puffervariable p: `append(p,v)` fügt ein Produkt v in den Puffer ein, `take(p)` entnimmt ein Produkt aus dem Puffer. Variable n gibt jeweils die Anzahl Produkte im Puffer an. Variable s und d sind binäre Semaphore. Alle Variable werden sowohl im Produzenten als auch im Konsumenten Thread benutzt, lediglich *lokal* ist eine Variable, die nur lokal im Konsumenten auftritt.

Das `parbegin` Statement in der Hauptfunktion läßt einen Produzenten und einen Konsumenten Thread parallel ablaufen.

```

Int n ;
Puffer p ;
binary_semaphore s = 1 ;
binary_semaphore d = 0 ;

void producer(){
  while (true) {
    product v = produce();
    wait(s);
    append(p,v);
    n = n + 1 ;
    if (n == 1)
      signal(d);
    signal(s) ;
  }
}

```

```

void consumer(){
  int lokal ;
  wait(d) ;
  while (true) {
    product v ;
    wait(s);
    v = take(p);
    n = n - 1 ;
    lokal = n ;
    signal(s) ;
    consume(v) ;
    if (lokal == 0)
      wait(d);
  }
}

void main(){
  n = 0; initialisiere(p) ;
  parbegin(producer,consumer);
}

```

- Die Implementierung nutzt binäre Semaphore. Erklären Sie zunächst deren Funktionsweise: Welchen Effekt hat ein Aufruf von `wait(s)` bei `s=0` und `s=1`, welchen Effekt hat ein Aufruf von `signal(s)` bei `s= 0` und bei `s= 1` ?
- Welche Variable müssen im wechselseitigen Ausschluß behandelt werden und warum ?
- Markieren Sie die kritischen Abschnitte in dem angegebenen Pseudocode.
- Sichert die Implementierung den wechselseitigen Ausschluß zu ? Begründen Sie Ihre Aussage.

Lösung Aufgabe 2:

Name: \_\_\_\_\_ Vorname: \_\_\_\_\_

Lösung Aufgabe 2:

Name: \_\_\_\_\_ Vorname: \_\_\_\_\_

Aufgabe 3: (20 Punkte) Deadlock

Deadlocks lassen sich durch eine vorsichtige Vergabe von Ressourcen vermeiden.

- a) Was ist ein sicherer Zustand in einer Vermeidungsstrategie für Deadlocks ?  
b) Ist der folgende Zustand sicher ? Wir betrachten ein Szenario aus 4 Prozessen, P1, P2, P3 und P4, und 3 Arten von Ressourcen, R1, R2, R3. R1 ist neunfach vorhanden, R2 dreifach und R3 sechsfach. Begründen Sie Ihre Entscheidung.

Matrix der maximalen Anforderungen				Matrix mit aktueller Ressourcenbelegung			
	R1	R2	R3		R1	R2	R3
P1	3	1	4	P1	2	1	1
P2	4	2	2	P2	0	0	2
P3	3	2	2	P3	1	0	0
P4	6	1	3	P4	6	1	2

- c) Wie würden Sie nach dem Bankier Algorithmus (Bankers Algorithm) entscheiden, ob in obiger Situation die Anforderung des Prozesses P3 nach 2 weiteren Ressourcen R2 erfüllt werden darf? Skizzieren Sie knapp das prinzipielle Vorgehen.

Lösung Aufgabe 3:

Name: \_\_\_\_\_ Vorname: \_\_\_\_\_

Aufgabe 4: (20 Punkte) Speichermanagement

- a) Wählen sie eine sinnvolle Seitengröße für Paging bei einer 32 bit Architektur.
- b) Wie groß ist die Seitentabelle eines Prozesses bei der von Ihnen gewählten Seitengröße ? Wie groß ist die Seitentabelle bei einer zweistufigen Tabelle und 1024 Einträgen in der Wurzeltabelle ?
- c) Wie groß ist die externe Fragmentierung und die interne Fragmentierung je Prozeß im schlimmsten Fall ?
- d) Bestimmen Sie in der logischen Adresse 01010101010100110101001010101010 die Seitennummer. Berechnen Sie die physikalische Adresse unter der Annahme, das die Seite sich im Rahmen mit der Nummer 12 (dezimal) befindet.

Lösung Aufgabe 4:

Name: \_\_\_\_\_ Vorname: \_\_\_\_\_

Aufgabe 5: (20 Punkte) Ersetzungstrategien von Speicherseiten bei Pagingverfahren

Wir betrachten die folgende Sequenz von Seitenzugriffen bei einer Speichergröße von 4 Seiten: 1 2 3 1 4 2 5 1 2 3 4 5 ausgehend von einem vollständig freien Speicher.

Skizzieren Sie die Auswirkungen bekannter Ersetzungsstrategien.

Lösung Aufgabe 5:

