Prof. Dr. Michael Jäger FB MNI

Klausur "Betriebssysteme I" – 9.3.2000

Bitte bearbeiten Sie die Aufgaben soweit wie möglich auf den Aufgabenblättern.

Nachname: Vorname:
Matrikelnummer: Semester:

Bitte beantworten Sie zunächst folgende Fragen durch Ankreuzen:

- O Ich mache mein Vordiplom nach der neuen Version der Prüfungsordnung und habe keine Betriebssystem-Prüfung nach alter Version bestanden (Prüfungsumfang = 4+2 SWS / Teil A + B der Klausur, 120 Min. Zeit)
- O Ich mache mein Vordiplom nach der alten Version der Prüfungsordnung (Prüfungsumfang = 3+1 SWS / nur Teil A der Klausur, 90 Min. Zeit)
- O Ich habe Betriebssysteme nach der alten Version der Prüfungsordnung schon bestanden mache mein Vordiplom nun aber nach der aktuellen Version der Prüfungsordnung (Prüfungsumfang = 1+1 SWS / nur Teil B der Klausur, 30 Min. Zeit)
- O Ich habe an der Blockveranstaltung „Betriebssysteme I" im Februar 2000 teilgenommen und mir die Praktikumsaufgabe testieren lassen

Ich bestätige hiermit die Korrektheit der oben gemachten Angaben.

Unterschrift:

Aufgabe	Punktzahl maximal	Punktzahl erreicht
Teil A - 1	4	
Teil A - 2	4	
Teil A - 3	10	
Teil A - 4	8	
Teil A - 5	6	
Teil A - 6	4	
Teil A - 7	4	
Teil A - 8	4	
Teil A - 9	3	
Teil A - 10	4	
Summe	51	

Aufgabe	Punktza4 maximal	Punktzahl erreicht
Teil B - 11	4	
Teil B - 12	3	
Teil B - 13	6	
Teil B - 14	4	
Summe	17	

Note:

Aufgabe 1 (Prozesse – 4 Punkte)

Punkte ☐ *von* **4**

Nennen Sie die wichtigsten Informationen, die ein Betriebssystem im Prozessdeskriptor eines Prozesses aufbewahrt.

Aufgabe 2 (Prozesse – 4 Punkte)

Punkte ☐ *von* **4**

Schildern Sie den Mechanismus, mit dem der Betriebssystemkern den ausführenden Prozess verdrängt, sobald ein Prozess mit höherer Priorität in den Zustand „bereit“ wechselt. Woran merkt das BS den Zustandswechsel? Wie kann das BS einen Prozess verdrängen, wenn dieser doch den Prozessor hat?

Aufgabe 3 (Prozesskontrolle, Pipes, E-/A-Umlenkung – 10 Punkte) *Punkte* ☐ *von* **10**

Schreiben Sie ein C-Programm, das die Anzahl der Zeilen in der Ausgabe des Kommandos "`ls -l`" ausgibt. Dazu soll das `ls`-Programm als Subprozess ausgeführt werden, der seine Ausgabe durch eine Pipe an den Vaterprozess leitet.

Aufgabe 4 (Synchronisation – Punkte 8)

Punkte ____ *von* **8**

Ein einfacher Pipe-Mechanismus soll implementiert werden. In die Pipe wird mit put(c) das nächste Zeichen geschrieben, get() liefert das nächste ungelesene Zeichen. Als Speicher dient ein Ringpuffer.

Ergänzen Sie in dem unten stehenden Lösungsansatz die Synchronisation:
- gegenseitiger Ausschluss
- get() bei leerem Puffer blockieren
- put() bei vollem Puffer blockieren

Verwenden Sie dazu Semaphore, deren Implementierung gemäß nachstehender Schnittstelle vorausgesetzt werden kann.

```
class semaphor {
public:
  semaphor(void);
  init(int anfangswert); // Initialisierung
  up(void);
  down(void);
};

class pipe {
  int puffergroesse;
  char *puffer;
  int frei, naechsteszeichen;
  semaphor  kritisch, frei, vorhanden;
public:
  pipe(int groesse);
  void put(char c);
  char get(void);
};

pipe::pipe(int groesse){
  puffergroesse=groesse;
  puffer=new char[puffergroesse];
  frei=0;
  naechsteszeichen=0;
}

void pipe::put(char c){
  puffer[frei]=c;
  frei = (frei+1) % groesse;
}

char pipe::get(){
  int c=puffer[naechsteszeichen];
  naechsteszeichen = (naechsteszeichen+1) % groesse;
  return c;
}
```

Aufgabe 5 (Shellscript – 6 Punkte)

Punkte ☐ *von* **6**

Das Kommando

`fgrep` *String file*

sucht nach der Zeichenkette *String* in der Datei *file*.

Das Kommando

`zcat` *file* `|` `fgrep` *String*

dekomprimiert *file* vorher noch.

Das folgende Pipeline-Kommando terminiert erfolgreich, wenn es sich bei *File* um eine *gzip*-komprimierte Datei handelt.

```
file Dateipfad | grep -q "gzip compressed"
```

Setzen Sie aus diesen Bausteinen ein `suchestring`-Shellscript zusammen, dem man die Namen beliebig vieler, ggf. komprimierter Textdateien als Argumente übergibt. `suchestring` soll interaktiv einen Suchstring einlesen und diesen in allen angegebenen Textdateien suchen. Komprimierte Textdateien sollen vorher dekomprimiert werden.

Aufgabe 6 (Verklemmungen – 4 Punkte)

Punkte ☐ *von* **4**

Um durch eine sichere Ressourcenvergabe beim Problem der 5 dinierenden Philosophen eine Verklemmung zu vermeiden, kann man Ressourcenzuordnungsdiagramme mit solchen Kanten erweitern, die später benötigte Ressourcen repräsentieren.

Erläutern Sie an nachstehendem Beispiel die Verwendung dieser Kanten.

Aufgabe 7 (Hauptspeicher – 2+2 Punkte)

Punkte ☐ *von* **4**

a) Welche Möglichkeiten der Verwaltung freier Speicherblöcke (Platte oder Hauptspeicher) kennen Sie ?

b) Erläutern Sie, wie das Betriebssystem in einem System mit Paging und Segmentierung die Seiten-Seitenrahmenzuordnung eines *segmentierten* Prozesses verwaltet.

Aufgabe 8 (Hauptspeicher – 4 Punkte)

Punkte ☐ *von* **4**

In einem System mit Paging steht ein nur 3 Rahmen großer Hauptpeicher zur Verfügung. Geben Sie für die Auslagerungsstrategien FIFO und OPT (optimale Strategie) die Speicherbelegung und die Anzahl der Seitenfehler nach folgenden Seitenzugriffen an:

5 7 1 2 7 6 7 4 2 7 1 6 4 5

FIFO

5	5	5	2	2				
	7	7	7	7				
		1	1	1				

Seitenfehler:

OPT

5	5	5	2	2				
	7	7	7	7				
		1	1	1				

Seitenfehler:

Aufgabe 9 (CPU-Scheduling – 3 Punkte)

Punkte ☐ *von* **3**

Zur Berücksichtigung von Prioritäten verwalten viele Scheduler separate Prozesswarteschlangen für jede Prioritätsstufe.

Wie könnte man mit nur einer Warteschlange für alle nicht blockierten Prozesse und einer einfachen „round robin“-Strategie dafür sorgen, dass Prozesse mit hoher Priorität begünstigt werden ?

Aufgabe 10 (Dateisystem – 4 Punkte)

Punkte ☐ *von* **4**

Wozu enthält bei einem UNIX-Dateisystem jeder Inode einen Referenzzähler ?

Wenn ein Prozess in einem Mehrbenutzersystem, z.B. UNIX oder Windows NT, eine Datei öffnet, führt das Betriebssystem eine Zugriffsrechtsprüfung durch. Beschreiben Sie kurz, wie diese Prüfung typischerweise stattfindet, insbesondere auch, welche Datenstrukturen das Betriebssystem zur Prüfung der Rechte braucht.

Aufgabe 11 (Netzwerkprotokolle – 4 Punkte)

Punkte ☐ *von* **4**

Erläutern Sie kurz die Funktion des IP-Protokolls in folgender Netzwerkkonfiguration: Zwei lokale Netze N_1 und N_2 (Ethernet) sind durch einen Router R verbunden. Der Rechner H_1 in N_1 sendet dem Rechner H_2 in N_2 Daten.

Aufgabe 12 (Server-Architekturen – 3 Punkte)

Punkte ☐ *von* **3**

Was ist ein nebenlaüfiger Server (concurrent server)?

Aufgabe 13 (Socket-Schnittstelle – 2+1+3 Punkte)

Punkte ☐ *von* **6**

a) Erläutern Sie die Begriffe *verbindungsorientierte* und *paketorientierte* Kommunikation.

b) Welches der beiden Kommunikationsschemata wird durch die Socket-Schnittstelle unterstützt?

c) Nennen Sie einige Socket-Systemaufrufe mit kurzer Erläuterung ihrer Funktionalität.

Aufgabe 14 (Internet-Adressen – 2+2 Punkte)

Punkte ☐ *von* **4**

a) Wie ist eine Internet-Adresse (`struct inet_addr`) aufgebaut?

b) Wozu dient eine TCP-Port-Nummer ?