Prüfungsprotokoll der mündlichen Hauptdiplomsprüfung zu Kurs 1802 - Betriebssysteme (Version 04/97)
Prüfer: Prof. Dr. Güting
Termin/Datum: 27.01.2003
Note: 1,0

Leute erst einmal: Schreibt bitte Prüfungsprotokolle und am besten noch mit kurzen Antworten!! Die Prüfungsprotokolle helfen einem sehr weiter, weil man ungefähr abschätzen kann, was in einer Prüfung gefragt wird und verkürzen somit erheblich die Vorbereitungszeit und die Antworten geben einem oft sehr gute Hinweise einen Themenbereich besser zu verstehen.

Welche Aufgaben hat ein Betriebssystem?

- Aufgaben aus KE1 aufgezählt + jeweils kurze Erläuterung dazu

- Bei der Betriebsmittelverwaltung antwortete ich: Das sind diejenigen Ressourcen die das Betriebssystem selbst und die Applikationen brauchen, um ablaufen zu können, z.B. entscheidet ein Scheduling-Algorithmus (= Scheduler) darüber, welcher Prozeß als nächster den Prozessor zum Rechnen zugeteilt bekommt.

Darauf hin die Frage: Sie erwähnten gerade den Scheduler - Was ist das?

Ein Scheduling-Algorithmus (= Scheduler) als Teil des Betriebssystems entscheidet darüber, welcher von mehreren rechenbereiten Prozessen als nächster den Prozessor zum Rechnen zugeteilt bekommt.

Was für Scheduling-Algorithmen gibt es und wann verwendet man sie?

- In Batchsystemen (= Stapelverarbeitungssystemen) nicht preemptive Scheduling-Algorithmen, wie bspw. SJF (Shortest Job First) oder FCFS (First Come First Served).

- In Dialogsystemen preemptive Scheduling-Algorithmen, wie z.B. Round Robin.

Was bedeutet nicht preemptiv? Und preemptiv?

- Nicht preemptiv bedeutet nicht verdrängend. Ein nicht preemptiver Scheduling-Algorithmus entzieht einem rechnenden Prozeß (= der Prozeß besitzt den Prozessor zum Rechnen) nicht den Prozessor. Der Prozeß kann nur selbst den Prozessor abgeben, z.B. wenn er freiwillig den Prozessor über einen Systemaufruf abgibt oder wenn er beendet ist oder wenn er bspw. eine E/A vornimmt, sich also im Zustand "blockiert" befindet.

- Preemptiv bedeutet verdrängend. Ein preemptiver Scheduling-Algorithmus entzieht einem rechnenden Prozeß nach Ablauf einer gewissen Zeitspanne, genannt: Zeitscheibe oder Quantum, den Prozessor, indem ein Hardwaretimer eine Unterbrechung erzeugt und somit die Kontrolle ans BS übergeht.

Prof. Güting fragte nicht genauer zu SJF, FCFS oder Round Robin nach.

Prof. Güting wollte nun das Prozeßmodell mit den Zustandsübergängen aufgeschrieben und erläutert bekommen:

Ich zeichnete also die Zustände "neu", "rechenbereit", "rechnend", "blockiert" und "beendet" hin + kurze Erläuterungen zu den Zustandsübergängen

Warum muß man parallele Prozesse synchronisieren?

- Für kritische Abschnitte muß gegenseitiger Ausschluß über Synchronisationsmechanismen gewährleistet werden, damit es zu keinen Wettbewerbsbedingungen kommt.

- Weiter muß man synchronisieren, wenn mehrere Prozesse gemeinsam an der Lösung einer Aufgabe arbeiten, d.h. miteinander Kooperieren, z.B. muß im Erzeuger-Verbraucher-Problem der Erzeuger dem Verbraucher mitteilen, daß er wieder eine Ware im gemeinsamen prozeßglobalen Pufferspeicher abgelegt hat.

- Weiter muß man synchronisieren um Deadlockzustände zu vermeiden.

Was sind Deadlocks?

Eine Menge von Prozessen befindet sich in einem Deadlockzustand, falls alle Prozesse aus der Menge auf das Eintreten eines Ereignisses warten, das aber nur von einem anderen Prozeß aus dieser Menge ausgelöst werden kann. Da aber alle Prozesse auf das Eintreten eines Ereignisses warten, kann keiner der Prozesse jemals ein Ereignis auslösen. Damit sind aber alle Prozesse für immer blockiert und können nicht in ihrer Bearbeitung fortschreiten.

Wollen Sie noch das Beispiel aus dem Skript hören, mit den zwei Studenten und den Büchern? Prof. Güting: Nein.

Wann können Deadlocks eintreten?

Dazu müssen die folgenden 4 Bedingungen gleichzeitig gegeben sein:

(1) **Bedingung des gegenseitigen (= wechselseitigen) Ausschlusses**: Ein Prozeß belegt alle seine Betriebsmittel exklusiv.

(2) **Bedingung der Nichtunterbrechbarkeit (no preemption)**: Die Betriebsmittel die ein Prozeß bereits exklusiv belegt, können ihm nicht von einem anderen Prozeß entzogen werden, wenn dieser die Betriebsmittel benötigt.

(3) **Belegungs- und Wartebedingung**: Ein Prozeß belegt bereits exklusiv Betriebsmittel, benötigt weitere Betriebsmittel und wartet nun darauf, daß diese freigegeben werden.

(4) **Bedingung der geschlossenen Kette (circular wait - zyklisches Warten)**: Es besteht eine geschlossene Kette aus Prozessen in der Weise, daß ein Prozeß bereits mindestens ein Betriebsmittel exklusiv belegt, daß vom nächsten Prozeß in der Kette benötigt wird.

Beispiel: (Student1) ----> [Buch A] ----> (Student 2) ----> [Buch B] (und ein Pfeil zurück zu Student1)

Welche Mechanismen zur Synchronisation gibt es?

- Synchronisationsvariablen
- Semaphore
- Monitore
- Nachrichtenaustausch (message passing)

Synchronisationsvariablen verwendet man normalerweise aber nicht, da ihr Hauptnachteil das aktive Warten (busy waiting) ist. Hier schaut ein Prozeß von Zeit zu Zeit in einer gemeinsamen prozeßglobalen Synchronisationsvariable nach, ob er den kritischen Abschnitt betreten darf, z.B. while turn <> 0 {do nothing}, das verschwendet aber unproduktiv Prozessorzeit.

Was sind Semaphore?

- Der Vorteil gegenüber Synchronisationsvariablen ist, daß hier kein busy waiting gebraucht wird um einen gegenseitigen Ausschluß für die kritischen Abschnitte zu gewährleisten. Hier wird ein Prozeß, wenn er nicht seinen kritischen Abschnitt betreten kann, in seinem Ablauf gestoppt, in den Wartezustand versetzt und in die vom Semaphor verwaltete Warteliste eingetragen. Damit kann aber ein anderer Prozeß den Prozessor zum Rechnen benutzen.

- Ein Nachteil von Semaphoren ist, daß es bei falscher Anwendung der P- und V-Operationen zu sporadisch auftretenden Fehlern kommt, die nur schwer lokalisiert werden können.

- Ein Semaphor ist eine gemeinsame prozeßglobale nicht-negative und ganzzahlige Synchronisationsvariable, die mit einer Warteliste verbunden ist (= jeder Semaphor besitzt genau eine Warteliste). Auf einen Semaphor kann nur über die atomaren Elementaroperationen P und V zugegriffen werden.

Erklären Sie bitte die P- und die V-Operation.

- Die P-Operation ist die Anfrageoperation. Ein die P-Operation ausführender Prozeß kann ermitteln, ob er seinen kritischen Abschnitt betreten kann. Die P-Operation funktioniert nun folgendermaßen:

  - Ist der Semaphorwert gleich 0, kann ein die P-Operation ausführender Prozeß nicht seinen kritischen Abschnitt betreten, da sich bereits ein anderer Prozeß in seinem kritischen Abschnitt befindet. Der Prozeß wird gestoppt, in den Wartezustand versetzt und in die vom Semaphor verwaltete Warteliste eingetragen. Er wird erst wieder aufgeweckt, wenn die korrespondierende V-Operation ausgeführt wird. Dann erniedrigt der die P-Operation ausführende Prozeß den Semaphorwert um 1 und tritt in seinen kritischen Abschnitt ein.
  - Ist der Semaphorwert echt größer 0, erniedrigt der die P-Operation ausführende Prozeß den Semaphorwert um 1 und betritt seinen kritischen Abschnitt.

- Die V-Operation ist die Freigabeoperation. Ein die V-Operation ausführender Prozeß gibt damit den kritischen Abschnitt frei. Die V-Operation funktioniert folgendermaßen:

  - Der Wert des Semaphors wird um 1 erhöht. Zusätzlich wird, wenn sich mindestens ein Prozeß in der vom Semaphor verwalteten Warteliste befindet, ein Prozeß aus der Warteliste ausgewählt und aufgeweckt.
  - Danach setzt der die V-Operation ausführende Prozeß seinen Ablauf normal fort.

Was sind Monitore?

Monitore sind nur exklusiv nutzbare Objekte, bestehend aus einer Menge von Monitorprozeduren, innerhalb derer die kritischen Abschnitte realisiert werden und weiter bestehend aus Datenstrukturen, z.B. den condition-Variablen für die wait- und signal-Operationen oder im Erzeuger-Verbraucher-Problem der gemeinsame prozeßglobale Pufferspeicher, auf die nur über die Monitorprozeduren zugegriffen werden darf. Mit dem Monitor ist dann noch eine Warteliste verbunden.

Ein Monitor garantiert nun gegenseitigen Ausschluß, obwohl jederzeit Monitorprozeduren von mehreren Prozessen aufgerufen werden dürfen, indem aber zu jedem Zeitpunkt immer nur jeweils höchstens eine Monitorprozedur ausgeführt! werden darf. Dazu fügt der Compiler für den Programmierer unsichtbar zusätzlichen Programmcode ein, der überprüft, ob ein anderer Prozeß im Monitor aktiv ist.

Was ist der Vorteil von Monitoren gegenüber Semaphoren?

Der gegenseitige Ausschluß wird automatisch durch den Compiler gewährleistet (s.o.)

Was ist der Nachteil von Monitoren gegenüber Semaphoren?

Monitore müssen von einer Programmiersprache angeboten werden = der Compiler muß dem Sprachkonstrukt Monitor umgehen können, während Semaphore über Systemaufrufe realisiert werden können, wenn das Betriebssystem Semaphore anbietet.

Welche Verfahren der Hauptspeicherverwaltung kennen Sie?

- einfache zusammenhängende Speicherzuweisung

- mehrfache zusammenhängende Speicherzuweisung mit MFT und MVT

- die mehrfache nicht zusammenhängende Speicherzuweisung mit Paging und der Segmentierung

- virtuellen Hautpspeicher, realisiert über demand paging

Erzählen Sie bitte etwas zu MFT und MVT.

- bei beiden Verfahren können sich immer mehrere Prozesse im physischen Hauptspeicher befinden + jedem Prozeß wird ein zusammenhängender Abschnitt des physischen Hauptspeichers zugeteilt

- MFT: beim Starten des Betriebssystems wird der physische Hauptspeicher in mehrere Segmente unterschiedlicher Größe unterteilt, diese Unterteilung ist dann während des gesamten BS-Laufes unveränderlich, d.h. maximale Anzahl und Größe der einzulagernden Prozesse ist vorgegeben; Nachteil von MFT: interne Fragmentierung; die interne Fragmentierung kann dann bspw. durch best fit only abgemildert werden, aber Nachteile von best fit only

- MVT: hier bekommt jeder Prozeß während des BS-Laufes so viel physischen Hauptspeicherplatz zugewiesen, wie er braucht; Nachteil: externe Fragmentierung verringert die Anzahl parallel ausführbarer Prozesse; bei einer zu hohen externen Fragmentierung muß sehr zeitaufwendig kompaktifiziert werden

Welche Speicherplatzvergabestrategien gibt es bei MVT?

Best Fit, Worst Fit, First Fit, Rotating First Fit jeweils mit der Wirkung dieser Speicherplatzvergabestrategien + Buddy-Verfahren

Erläutern Sie bitte das Buddy-Verfahren.

Fazit der Prüfung: Prof. Güting ist ein sehr freundlicher Prüfer und seine Fragen waren sehr präzise gestellt - es gab keine Unklarheiten was er wissen wollte. Ich wünsche Euch viel Erfolg bei Eurer BS-Prüfung!!!

Gedächtnisprotokoll zur Diplomprüfung

Kurs: 01802 - Betriebssysteme

Prüfer: Prof. Dr. Güting
Datum: 29.1.2001
Dauer: 25 min
Note: 1,0

- **Was sind die Aufgaben eines Betriebssystems?**
  Betriebsmittelverwaltung, Gerätesteuerung, Fehlerbehandlung, Schutz, Mehrprogrammbetrieb, ...

- **Welche Probleme treten beim parallelen Ablauf von Programmen auf?**
  Verwaltung der gemeinsam benutzten Betriebsmittel, ... Prof. Güting wollte hier aber insbesondere 'Deadlock' hören.

- **Was muss bei parallelen Prozessen synchronisiert werden?**
  Der Zugriff auf gemeinsame Daten -> Determiniertheit der Prozesse.

- **Welche Mittel zur Synchronisation gibt es?**
  Synchronisationsvariablen, Semaphore, Nachrichten, Monitore.

- **Was sind Semaphore?**
  'gekapselte Synchronisationsvariable', Zugriff nur über die P- und V-Operation.

- **Welchen Wertebereich haben Semaphore?**
  Natürliche Zahlen inkl. 0.

- **Was ist das Erzeuger-Verbraucher-Problem?**
  2 Prozesse, Ware wird über einen gemeinsamen Puffer ausgetauscht.

- **Schreiben Sie eine Lösung des Erzeuger-Verbraucher-Problems auf!**
  Erst ganz einfache Lösung mit einem Semaphor, dann auf Nachfragen Verfeinerung mit begrenztem Puffer und drei Semaphoren.

- **Was ist ein Deadlock ?**
  Prosesse blockieren sich gegenseitig und dauerhaft benötigte Betriebsmittel, 'Wartet-auf-Graph' enthält Zyklus.

- **Wie löst man einen Deadlock auf?**
  Ein Prozess muss abgebrochen werden.

- **Wie kann man einen Deadlock verhindern?**
  Betriebsmittelhierarchie

- **Welche Möglichkeiten gibt es, den Hauptspeicher zu verwalten?**
  - zusammenhängende Speicherzuweisung: MFT und MVT, jeweils mit Vor- und Nachteilen sowie dazugehöriger Scheduling-Strategien.

- Paging mit LRU-Strategie

- **Wie wird der Haupspeicher auf die Prozesse verteilt?**
  Lokalität und Arbeitsmenge erklärt, Zuteilung von Seitenrahmen im Umfang der Arbeitsmenge

- **Was geschieht, wenn die Summe der Arbeitsmengen zu groß wird?**
  Einige Prozesse müssen mitsamt ihren Seiten ausgelagert werden.

- **Wie teilt man den Hauptspeicher bei der unzusammenhängenden Speicherzuteilung auf?**
  Jeweils eigene Segmente für Programm, Daten und Stack.

Die Prüfungsatmosphäre war erwartungsgemäß freundlich.
Prof. Güting begann ohne Umschweife mit der Prüfung. Der kurze Small-Talk zum Abbau der Nervosität, den ich von anderen Prüfern gewohnt bin, fand nicht statt.
Vielleicht auch deswegen waren meine Antworten zu Beginn etwas holprig. Dies hat sich aber nicht negativ auf die Note ausgewirkt.
Es kamen erneut keine Fragen zu Dateisystem, Sicherheit und Kommandosprachen, dafür geht es bei den anderen Themen bis ins Detail.

Viel Erfolg !

Protokoll Prüfung 1802 Betriebssysteme (Version 04/99)
Prüfer: Prof. Dr. Güting
Termin: 20.12.1999
Dauer: 25 Min.

- Welche Aufgaben hat das Betriebssystem?
- Welche Probleme treten bei parallel ablaufenden Prozessen auf?
- Mechanismen des wechselseitigen Ausschusses
- Was ist ein kritischer Abschnitt?
- Nachteile bzw. Schwierigkeiten von globalen Variablen für die Realisierung des wechselseitigen Ausschlusses
- Erklärung der Funktionsweise von Semaphoren mit ihren P- und V-Operationen
- Wozu werden Semaphoren eingesetzt?
  a) wechselseitiger Ausschluß, b) Synchronisation
  Am Beispiel des Erzeuger-Verbraucher-Problems erläutert.
- Was ist das Erzeuger-Verbraucher-Problem? In umgangssprachlicher Form erklären.
- Erzeuger-Verbraucher-Problem algorithmisch mittels Semaphoren für einen unendlich großen Speicher aufschreiben und erklären
- Was ist ein virtueller Speicher?
- Wie werden die Speicherseiten verwaltet?
- Formel für die Umrechnung von logischen Adressen in physische Adressen angeben
- Was wird gemacht, falls der Hauptspeicher vollständig mit Prozessen belegt ist? Auslagerung von Prozessen -> swapping
- Welche dynamischen Seitenaustauschverfahren gibt es? Arbeitsmengenstartegie, Seitenfehlerhäufigkeitsstrategie
- Was ist eine Arbeitsmenge?
- Wie wird die Menge der im Hauptspeicher befindlichen Seiten bei der Seitenfehlerhäufigkeitsstrategie festgelegt?

Fazit der Prüfung:
Die Prüfung fand in einer sehr ruhigen, entspannten und angenehmen Atmosphäre statt. Bei Mißverständnissen oder Verständnisschwierigkeiten hat Prof. Güting die Fragen nochmals in anderer Art wiederholt und weitere Tips abgegeben, so dass eine Beantwortung der Fragen leichet möglich war. Die Benotung ist fair. Prof. Güting ist als Prüfer sehr zu empfehlen.

Gedächtnisprotokoll Diplomprüfung
Kurs 1802 Betriebssysteme

Prüfer: Prof. Dr. Güting

Zeit: 20.12.1999

Dauer: 25 Minuten

Prüfungsinhalt

1. Welche Aufgaben hat ein Betriebssystem?

   Alle Punkte aus KE 1 mit kurzer Erläuterung

2. Prozesskommunikation

- Was für Prozesse gibt es?

  Konkurrente, disjunkte, überlappende Prozesse

- Warum müssen Prozesse synchronisiert werden

  Erläuterung von kritischer Abschnitt, gegenseitiger Ausschluß

- Möglichkeiten der Synchronisation

  Synchronisationsvariable, Semaphore, Monitore, Nachrichtenaustausch mit kurzer Erklärung der Verfahren

- Erzeuger-Verbraucher-Problem mit Semaphoren

  Algorithmus aufschreiben und erklären

- Deadlock

  Nur kurz angerissen

3. Speicherverwaltung

   Alle Verfahren der Speicherverwaltung genannt und kurz erläutert, Schutzmechanismen (Grenzregister), MFT, MVT, Interne/externe Fragmentierung, Speicherzuweisungsstartegien bei MFT und MVT, Unterschied von First Fit und Rotating First Fit, Warum sind viele kleine Segmente bei First Fit am Anfang des Speichers?

Prüfungsverlauf

Prof. Dr. Güting ist ein sehr angenehmer Prüfer, die Prüfung findet im Dialog statt, wobei auch ein bißchen diskutiert werden kann, wenn die Diskussion nicht zu sehr ausschweift. Zur Prüfungsvorbereitung habe ich auch das Buch "Moderne Betriebssysteme" von Andrew Tannenbaum (dem Betriebssystempapst) herangezogen, weil der Kurstext nicht sehr ausführlich ist. Das Wissen aus dem Buch hat mir sehr viel mehr gebracht als der nackte Kurstext. Aufgrund der Note von 1,7 fühle ich mich in der Prüfungsvorbereitung mit dem Buch bestätigt. Prof. Güting sagte, dass es aufgrund von zwei Unsicherheiten "nur" eine gute Zwei und keine Eins geworden ist.

Und denke daran dein Prüfungsprotokoll hier veröffentlichen zu lassen!

Protokoll Prüfung 1802 Betriebssystem Variante 4/98

Prüfer: Prof. Dr. Güting
Zeit: 13. 01.99,
Dauer: 35 min.,
Note: 1,7

Güting: Warum muß man Prozesse synchronisieren?
Ich: Erstens um sie zeitlich aufeinander abzustimmen, um ein Beispiel zu nennen, beim Erzeuger/Verbraucher-Problem, beim Messen durch einen Prozeß und Auswerten durch einen anderen. Dann, wenn exklusive Betriebsmittel gemeinsam genutzt werden sollen, beim Nutzen gemeinsamer Programmabschnitte usw.
Güting: Ein Punkt fehlt noch?
Ich: ?
Güting: Um Deadlocks zu vermeiden! Wie kann man das machen?
Ich: (nannte die bekannten Methoden: Synchronisationsvariablen, Semaphore, Monitore, Mail-Box-Systeme)
Güting: Wie kommt es überhaupt zu Deadlocks?
Ich: (nannte die 4 Gründe aus den KE)
Güting: So, jaja. Gehen wir noch mal zu der Deadlockvermeidung. Sie haben ja ganz richtig vier Verfahren genannt. Können Sie die mal erläutern?
Ich: (Beginn mit Synchronisationsvariablen)
Güting: Warum nimmt man Synchronisationsvariablen eher weniger oft?
Ich: Es geht um die Unteilbarkeit eines Prozesses, der den Eintritt in einen kritischen Abschnitt regelt. Das ist bei Synchronisationsvariablen nicht so einfach. (Jetzt wollte ich ein Beispiel mit den Synchronisations-Variablen S_1 und S_2 aus dem Kurs bringen, wurde aber gestoppt).
Güting: Schon gut. Was sind Semaphore?
Ich: (begann mit Erklärung von $P(S_1, ..., S_n)$ und $V(S_1, ..., S_n)$)
Güting: (stoppte mich) Sagen Sie nur das Wesentliche!
Ich: Beim Eintritt muß für alle Si (i = 1, ... n) Si >=1 gelten
Güting: Und wenn nicht?
Ich: Wird der Prozeß gestoppt und kommt in eine Warteschlange.
Güting: Ja. Und was macht V?
Ich: V(s) = s + 1
Güting: Was kommt nach der V-Operation?
Ich: Der Prozeß fährt ganz normal fort. Während die korrespondierende P-Operation und der nachfolgende Prozeß aus der Warteschlange aufgeweckt wird.
Güting: So, naja. Geben Sie doch mal ein Beispiel für die Wirkung von Semaphoren.
Ich: (schrieb Erzeuger-Verbraucher-Prozeß mit Semaphoren Zustand" und „Inhalt", mit 1 bzw. 0 initialisiert, auf.)
Güting: Jetzt erläutern Sie die Monitore.
Ich: (brachte Beispiel mit Raum und Schlüssel usw.)
Güting: Das ist mir doch ein bißchen allgemein!?
Ich: Nun gut, die praktische Realisierung erfolgt über ATP's.
Güting: Was ist ein ATP?
Ich: Ein abstrakter Datentyp.
Güting: Und der heißt ATP? (lachte)
Ich: (die Nerven) Nein, natürlich ADT!
Güting: Und wie sieht nun die Realisierung aus?
Ich: (Kritischer Abschnitt realisiert sich durch die Nutzung des ADT, diese Nutzung ist exklusiv, so daß immer nur ein kritischer Abschnitt aktiv ist; brachte dann noch ein Anwendungsbeispiel.)
Güting: Gut, kommen wir jetzt zu der Speicherverwaltung. Was wissen sie da so alles?
Ich: (Nannte als Stichworte einfach, mehrfach, nicht zusammenhängend, weiter MFT, MVT, first fit, rotating fit, worst fit, best fit, buddy, Kompaktifizierung, feste Segmente, paging. Zählte Zuweisungs-und Auslagerungsstrategien auf.)
Güting: Erklären Sie die Begriffe.
Ich: (Erklärungen)
Güting: (Wollte jetzt zu first fit, rotating fit usw. Vor- und Nachteile wissen.)
Ich: (nannte Vor- und Nachteile)

Güting: Was ist der wesentliche Unterschied von Arbeitsmengenstrategien und Seitenfehlerhäufigkeitsstrategie?
Ich: (erklärte Verhalten bei Fehlgriffen)
Güting: Mit anderen Worten, bei der Arbeitsmengenstrategie wird eine Seite also beim ersten Fehlgriff eingelagert, bei der Seitenfehlerhäufigkeitsstrategie erst nach mehreren Fehlgriffen?
Ich: (tappte in die Falle.) Ja.
Güting: Was macht dann ein Prozeß bei der Seitenfehlerhäufigkeitsstrategie, wenn die gesuchte Seite nicht eingelagert wird. Meinen Sie nicht, wenn ein Prozeß eine Seite braucht, muß sei eingelagert werden?
Ich: (Auwei!) Doch! Die Seite wird natürlich gebraucht. Gegebenenfalls erhöht man die Anzahl der bereitgestellten Seitenrahmen.
Güting: (Fragte nach Vor- und Nachteilen der Strategien)
Ich: (siehe Kurs)
Güting: Was ist ein virtueller Hauptspeicher?
Ich: (siehe Kurs)
Güting: Wird beim Einlagern der gesamte Hauptspeicher zur Verfügung gestellt?
Ich: Nein, der Betriebssystemkern ist schon drin. Und wird die Platte mit einer FAT (ich sagte „fatt") verwaltet, ist die auch üblicherweise im Hauptspeicher, weil die nach dem Rechnerstart dort eingelagert wird.
Güting: Was is 'ne FAT (er sagte auch „fatt")?
Ich: Die „File Allocation Table"
Güting: Und da sagen sie „fatt"? Hab' ich so noch nie gehört. (Pause) Speicherhierarchie, was ist das?
Ich: Register, Cache, Haupt- oder Arbeitsspeicher, Hintergrundspeicher.
Güting: Sie sprachen vom virtuellen Hauptspeicher. Wie groß ist der?
Ich: Der ist jeweils so groß wie der zur Verfügung stehende Arbeitsspeicher, d. h., der physikalische 'Adreßraum ist genauso groß wie der logische bzw. virtuelle.
Güting: Sind Sie sich da sicher?
Ich: Ah, nein, ich glaube, die Größe des virtuellen Hauptspeichers wird doch eher durch die Bus- bzw. Adressbreite bestimmt.
Güting: Genau. Wie wird nun eine physikalische Adresse bestimmt?
Ich: (Begann die Umwandlung einer „3teiligen" virtuellen 32-bit-Adresse aus Segment-Nr./Seiten-Nr./Zeilen-Nr. in eine physikalische, ausgehend vom CR3 Register des 80 386-Prozessors, aufzuzeichnen.)
Güting: (Stoppte mich.) Ich meine eine andere Art der Umwandlung.
Ich: (Begann mit virtueller Adresse r, Seiten-Nr. = r div p und Zeilen-Nr.- in_ der- Seite = r mod p usw...)
Güting: (unterbrach mich mitten in meinen Erklärungen) Was ist 'ne Zeile? Habe ich dem Zusammenhang nie gehört.
Ich: Aber so wird das im Kurs genannt!
Güting: Ich sollte den Kurs vielleicht doch mal lesen! Ja, woll'n mal langsam Schluß machen. Was ist 'ne Arbeitsmenge?
Ich: Die Menge der Seiten, auf die ein Prozeß in letzter Zeit zugegriffen hat.
Güting: Was heißt in letzter Zeit?
Ich: Habe ich mich auch gefragt. Aber in den Kurseinheiten habe ich nix gefunden.
Güting: Was wäre denn möglich?
Ich: Vielleicht im Rahmen von 10.000 oder nur 1 .OOO Speicherzugriffen.
Güting: Ja, das wär'ne Möglichkeit.

Und damit war ich entlassen.
In Anbetracht zweier Schnitzer, auch wenn ich sie selber bemerkte und korrigierte, war die Benotung fair. Mir persönlich sind lange Prüfungen, wie diese, lieber. Man hat mehr Gelegenheit, Fehler wieder wettzumachen. Prof. Güting ist mit Sicherheit kein Prüfer, der lustvoll lächelnd bei der ersten Wissenslücke, die er findet, bis zum Prüfungsende genüßlich verharrt, um dann den am Boden zerstörten Prüfling mit einen „gerade noch so bestanden" demotiviert nach Hause zu schicken.

Prüfer-Testurteil: Empfehlenswert!

Gedächtnisprotokoll: Diplomprüfung praktische Informatik

Kurs : 1802 Betriebssysteme (Version SS 98)
Prüfer : Prof. Dr. Güting
Termin : Dezember 98
Dauer : 30 Minuten

Themen:

1. Prozesse:

Was läuft denn so in einem Rechner ? - Prozesse

Wann und wieso laufen Prozesse parallel ? - Parallelität innerhalb von Programmen durch mehrere Prozesse erläutert; erklärt, warum Prozesse von unterschiedlichen Programmen parallel laufen.

Was gibt es dabei für Probleme ? - Den Begriff des kritischen Abschnitts erläutert.

Was gibt es hier für Mechanismen ? - Die Handhabung von Semaphoren erläutert.

Können Sie das Erzeuger-Verbraucher-Problem beschreiben? - Hier sollte dann der Algorithmus erläutert werden.

Was ist das Problem der gegenseitigen Blockade ? - Deadlockbedingungen erläutert und Erkennungsmöglichkeiten beschrieben.

Wie kann man einen Deadlock verhindern ? - Verhinderung über hierarchische Ordnung der Betriebsmittel erläutert.

2. Hauptspeicher

Was bedeutet die Hauptspeicherhierarchie ? - Achtung hier sind nicht nur die Teile aus dem Skript zu nennen, sondern auch andere Komponenten wie Cache oder tertiäre Speicher.

Welche Verfahren für die Hauptspeicherverwaltung gibt es ? - Kurzer Überblick über die einfache zusammenhängende Speicherzuweisung, die mehrfache zusammenhängende Speicherzuweisung, die mehrfache getrennte Speicherzuweisung und das Paging genannt.

Sie haben gerade MFT und MVT genannt. Was ist das ? - Unterschiede erklärt,

Was gibt es für Strategien zur Einlagerung von Segmenten beim MVT ? - First fit, rotating first fit, best fit, worst fit und buddy kurz erläutert.

Was ist der Unterschied zwischen first fit und rotating first fit ? - Warum unterscheidet man das ? Unterschiedliche Auswirkungen auf Lücken erläutert.

Sie haben vorher die Seitenverwaltung genannt. Was ist das ? - Näher erläutert. Die wichtigen Begriffe waren hier wohl Seiten, Seitenrahmen, virtueller Hauptspeicher, Seitenadresstabelle, virtuelle Adresse.

Was sind Lokalität und Arbeitsmenge ? - Genauer erläutert.

Wenn der Hauptspeicher voll ist, muß eine Seite ausgelagert werden. Welche Strategien gibt es ? - Die optimale Strategie, LRU, FIFO und Zugriffbits erläutert.

Allgemeiner Eindruck:

Professor Güting ist ein ruhiger und sehr fairer Prüfer. Er stellte eine lange Reihe von Fragen auf die er jeweils eine kurze Erläuterung wollte. Langschweifige Ausführungen sind hier nicht gewünscht.
Zwischendurch wußte ich nicht worauf er eigentlich hinaus wollte. Dann hat er die Frage noch einmal anders formuliert. Hier sind Nachfragen unbedingt zu empfehlen. Negativ wurde weder die Nachfrage noch das anfängliche nicht richtige Verstehen der Frage bewertet.
Die Benotung ist ebenfalls fair und wohlgesonnen. Professor Güting ist als Prüfer unbedingt zu empfehlen.

Gedächtnisprotokoll zur Diplomprüfung Praktische Informatik

Kurs: 1802 Betriebssysteme (Version 199 1)

Pnifer: Prof. Güting

Termin: Januar 1995

Dauer: 20 min

Fragen:

- Weiche Aufgaben erfüllt ein Betriebssystem?
- Wie können Prozesse miteinander kommunizieren?
- Was bedeutet gegenseitiger Ausschlug?
- Wie wird die Synchronisation durch Semaphore realisiert (Erläuterung der P- und V-Operation)?
- Was heißt 'deadlock' , wann liegt ein deadlock vor?
- Welche Prozesse sind an einem deadlock beteiligt?
- Was kann unternommen werden, um einen deadlock zu verhindern bzw. was wird getan, wenn ein deadlock eingetreten ist?
- Erläuterung des Bankers Algorithmus
- Wie kann der Hauptspeicher vom Betriebssystem verwaltet werden?
- Welche Strategien gibt es bei zusammenhangender Speicherzuweisung?
- Was versteht man unter einem virtuellen Hauptspeicher?
- Wie werden die Seitenrahmen verwaltet?
- Was passiert. wenn alle Seitenrahmen belegt sind?
- Wie sollten die Seitenrahmen von den einzelnen Prozessen zugeordnet werden (feste oder variable Anzahl je Prozess, wonach sollte sich die Zuteilung richten)?
- Welche Problematik tritt beim Arbeitsmengenmodell auf, welche beim Beobachten der Seitenfehlerrate?

Bemerkungen:

angenehmes Prüfungsklima, keine Fragen zu den Themen Dateisysteme, Sicherheit und Kommandosprachen.