## Aufgabe 1: Grundlagen von Betriebssystemen

(a) Das Betriebssystem als Betriebsvermittler:
Das Betriebssystem übernimmt die Allokation, also die Vermittlung/Buchung von Systemressourcen, dies schließt beispielsweise Speicher (z.B. RAM) oder Rechenzeit auf dem Hauptprozessor ebenso ein wie die Nutzung von Systemgeräten, Netzwerkschnittstellen, usw.

Betriebssystem als virtuelle Maschine:
Um eine zunehmend "komfortable"Nutzung der Systemeigenschaften zu ermöglichen muss von der Hardwareebene abstrahiert werden. Das Betriebssystem wird dabei als eine Art virtuelle Maschine tätig. Ohne diese Abstraktion müsste Software explizit für die aktuell verwendete Hardware kompiliert werden und der Benutzer zu jeder Zeit alle Hardwareschichten kennen und mitunter verwalten.

(b) Das Betriebssystem als Betriebsvermittler hat folgende Aufgaben:

1. Verwaltung der Betriebsmittel, also den Bedarf von Prozessen auf Anfrage hin abzudecken, sowie die Betriebsmittel möglichst effektiv zu verteilen.

2. Es können mehere Programme auf die selbe Systemressource zugreifen; daher muss das Betriebsystem mit Konflikten über die Betriebsmittlel umgehen können. Beispielsweise eine Deadlock Situation, wie in der Vorlesung beschrieben wurde.

Das Betriebssystem als virtuelle Maschine hat folgende Aufgaben:

1. Die Nutzung eines Rechners möglichst einfach zu gestalten, es sollten also spezifische Informationen über die vorhandene Hardware "versteckt"werden, um dem Benutzer die Arbeit/Verwaltung mit dem System zu erleichtern.

2. Ebenso zählt dazu die Vermittlung zwischen Hardware und Software, damit nicht jede Applikation spezifisch für die aktuelle Architektur neu kompiliert werden muss.

## Aufgabe 2: Prozesse und Threads

(a) Definitionen und Zusammenhänge

Programm: Ein Programm ist eine Sequenz von Anweisungen in einer bestimmten Programmiersprache. Die Anweisungen bestehen aus verschiedenen Instruktionen und Kontrollstrukturen sowie aus Deklarationen, diese Anweisungen ßteuernünd "benutzenßur Ausführung den Programprozess und, im Fall von Multrithreaded-Anwendungen, die von dem Programm gestarteten Threads.

Prozess: Ein Programm, welches gerade abläuft und durch das Betriebssystem verwaltet wird, nennt man Prozess. Ein Prozess besitzt statische und dynamische Daten während seiner Laufzeit, die nicht änderbar sind bzw. sich verändern können. Außerdem hat jeder Prozess einen Prozessadressraum.

Thread: Threads, eine Art vereinfachter Prozesse, sind Programmabläufe, die keinen eigenen Prozessadressraum verwenden, sondern den Raum eines ihnen übergeordneten Prozesses "mitbenutzen". Die Erzeugung eines Threads ist deutlich effizienter als die eines Prozesses und eignet sich daher eher für Aufgaben von kurzer Dauer.

(d) Lebenszyklus eines Prozesses

X: Ready/Wait: Warten auf Zuteilung von Prozessorzeit sowie weiteren Systemressourcen. Der Prozess wurde also ordnungsgemäß erstellt und wartet auf Ausführung.

Y: Run: Der Prozess wird gerade abgearbeitet/ausgeführt.

Z: Block: Der Prozess wurde pausiert, z.B. weil andere Prozesse gerade eine höhere Ausführungspriorität haben oder weil auf Benutzereingaben gewartet wird.

a: Prozess wurde in die Prozessliste eingetragen sowie die Daten und der Code geladen, damit kann er als nächstes vom Scheduler des Betriebssystems in die Queue übernommen werden, also auf "Wait"geschaltet werden.

b: Für den Prozess ist nun Prozessorzeit frei, ebenso sind weitere nötige Systemressourcen allokiert, der Prozess kann also arbeiten und wird in den State "Run"überführt. c: Der Prozess muss z.B. auf Benutzereingaben warten, dann ist die Ausführung anderer Prozesse prioritär. Der Prozess wird damit in den State "Block"überführt.

d: Auf die zu wartenden Eingaben/Ergebnisse wurde gewartet und die Daten sind nun verfügbar, damit könnte der Prozess wieder aktiviert werden. Da ihm allerdings erneut Prozessorzeit reserviert werden muss, muss er zunächst wieder in den State "Ready/Wait"überführt werden. Die nötigen Systemressourcen müssen also erst wieder vorhanden sein.

e: Sind alle Aufgaben und Threads eines Prozesses fertig und/oder bereits terminiert, so kann der Prozess insgesamt terminieren und damit aus der Prozessliste entfernt werden.

f: Sind z.B. alle aktuellen Threads eines Prozesses durch mit ihrer Arbeit, müssen aber selbst wiederum neue Threads starten, kehrt man von der aktuellen Ausführung ("Run") zur Reservierung der benötigten Systemressourcen ("Ready/Wait") zurück.

## Aufgabe 3: n-Adressmaschine

(a) Es gilt zu berechnen: $R = ((a1 + a2)/a3) + ((b1 - b2)/b3)$

```
ADD >a1<>a2<    // a1 wird mit a2 addiert und der Wert
                // wird in a2 gespeichert

DIV >a2<>a3>    // das neue a2 wird durch a3 geteilt
                // und der Wert wird in a3 gespeichert

SUB >b1<>b2<    // b2 wird von b1 subtrahiert und der
                // Wert wird in b2 gespeichert

DIV >b2<<b3<    // das neue b2 wird durch b3 geteilt
                // und der Wert wird in b3 gespeichert

ADD >a3<>b3<    // a3 wird mit b3 addiert und somit hat
                // man die Gleichung in b3 stehen
```

Gesamtaufwand für Lösung der Aufgabe:

Es sind 5 Befehle mit 9 Leseaufträgen an Hsp./Cache und 5 Schreibaufträgen.

n= Speicherzugriffzeit 0,05*n= Befehlsausführungszeit (DIV, SUB, ADD..) Berechnungszeit: 5*0,05*n