

GSS-Übungsblatt 2

Alexander Timmermann, Jannis Krämer

1 Grundlagen von Betriebssystemen

- a)
 - 1. Verwaltung von Betriebsmitteln, d.h. OS als “**Betriebsmittelverwalter**”. Das OS übernimmt die Ermittlung, Zuteilung und Verwaltung der nötigen Betriebsmittel (Platz im Speicher, CPU-Zeit, ...)
 - 2. Abstraktion der Hardware, d.h. OS als “**virtuelle Maschine**”. Das OS macht durch die Abstraktion die Hardware für den Endnutzer bedienbar und bietet ein (im besten Falle) universelles Interface für Programme.
- b)
 - 1.
 - ★ Zuordnung von Betriebsmitteln an Prozesse, d.h. RAM, CPU-Zeit, ...
 - ★ Behandlung von Ressourcen-Konflikten und -Engpässen, insbesondere z.B. race conditions oder Programme in einer Endlosschleife.
 - 2.
 - ★ Rechnerarchitektur vor dem Nutzer verbergen, da sie in den meisten Fällen irrelevant ist.
 - ★ Hardwareunabhängigkeit durch Abstraktion herstellen

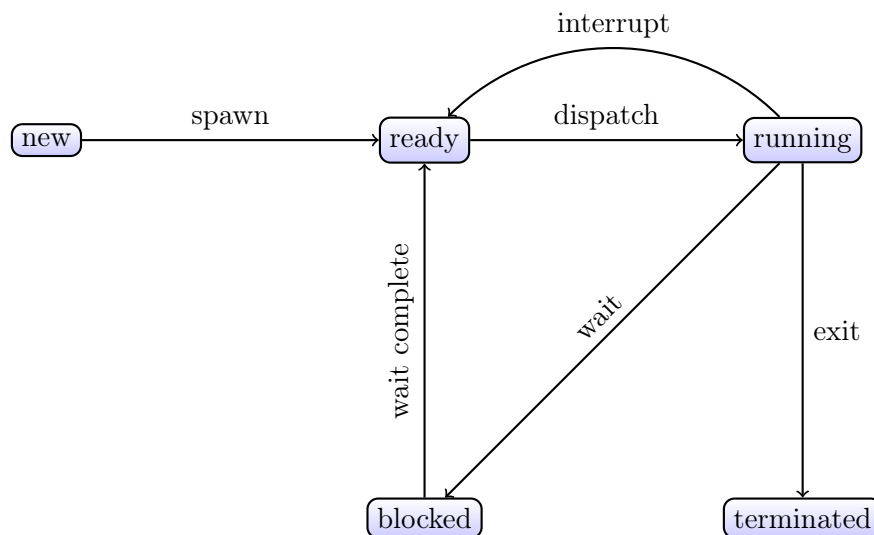
2 Prozesse und Threads

- a) **Programm:** Ein Programm ist eine vordefinierte Anzahl von Anweisungen an den Computer bzw. das Betriebssystem als Abstraktionslayer. Programme werden meist in höheren Programmiersprachen geschrieben und dann in für den Computer verständlichen Bytecode übersetzt.

Prozess: Ein Programm wird in einem sog. Prozess ausgeführt, dem vom Betriebssystem ein eigener Bereich im Speicher des Computers zugewiesen wird.

Thread: Ein Thread ist eine Art “Miniprozess”, der einem Prozess zugeordnet ist und sich mit den anderen Threads eines Prozesses den gleichen Bereich im Arbeitsspeicher teilt.

d)



ready: Ausführung möglich, jedoch momentan angehalten

running: Benutzung der CPU zur Berechnung

blocked: Ausführung hält und wartet auf externen Input

spawn: Prozess wird erstellt und wechselt in Zustand *ready*

dispatch: Dispatcher weist Aufgabe zu, die ausgeführt wird

interrupt: laufender Prozess wird vom Betriebssystem unterbrochen

wait: laufender Prozess wechselt in *blocked*-Zustand, d.h. wartet auf externen Input

wait complete: Input ist erfolgt, Prozess bereit zur weiteren Ausführung

exit: Der Prozess terminiert

3 n-Adressmaschine

a) 2-Adressmaschine, 8 Befehle, 13 Lese- und 8 Schreiboperationen

MOVE	>a ₁ <	>H ₁ <	$H_1 := a_1$
ADD	>a ₂ <	>H ₁ <	$H_1 := a_1 + a_2$
DIV	>H ₁ <	>a ₃ <	$H_1 := \frac{a_1 + a_2}{a_3}$
MOVE	>b ₂ <	>H ₂ <	$H_2 := b_2$
SUB	>b ₁ <	>H ₂ <	$H_2 := b_1 - H_2 = b_1 - b_2$
DIV	>H ₂ <	>b ₃ <	$H_2 := \frac{b_1 - b_2}{b_3}$
MOVE	>H ₁ <	>R<	$R := H_1 = \frac{a_1 + a_2}{a_3}$
ADD	>H ₂ <	>R<	$R := R + H_2 = \frac{a_1 + a_2}{a_3} + \frac{b_1 - b_2}{b_3}$

b) 1-Adressmaschine, 8 Befehle, 13 Lese- und 8 Schreiboperationen

LOAD	>b ₁ <	$AC := b_1$
SUB	>b ₂ <	$AC := b_1 - b_2$
DIV	>b ₃ <	$AC := \frac{b_1 - b_2}{b_3}$
SAVE	>H ₁ <	$H_1 := AC$
LOAD	>a ₁ <	$AC := a_1$
ADD	>a ₂ <	$AC := a_1 + a_2$
DIV	>a ₃ <	$AC := \frac{a_1 + a_2}{a_3}$
ADD	>H ₁ <	$AC := \frac{a_1 + a_2}{a_3} + \frac{b_1 - b_2}{b_3}$
SAVE	>R<	

c) **Stack-Rechner**, 12 Befehle, 6 Leseoperationen

PUSH	>a ₁ <	a_1
PUSH	>a ₂ <	a_1 a_2
ADD		$a_1 + a_2$
PUSH	>a ₃ <	$a_1 + a_2$ a_3
DIV		$\frac{a_1 + a_2}{a_3}$
PUSH	>b ₁ <	$\frac{a_1 + a_2}{a_3}$ b_1
PUSH	>b ₂ <	$\frac{a_1 + a_2}{a_3}$ b_1 b_2
SUB		$\frac{a_1 + a_2}{a_3}$ $b_1 - b_2$
PUSH	>b ₃ <	$\frac{a_1 + a_2}{a_3}$ $b_1 - b_2$ b_3
DIV		$\frac{a_1 + a_2}{a_3}$ $\frac{b_1 - b_2}{b_3}$
ADD		$\frac{a_1 + a_2}{a_3} + \frac{b_1 - b_2}{b_3}$
POP	>R<	