

Bitte beachten Sie, dass die optionalen Aufgaben zwar nicht bewertet werden, jedoch als Vorbereitung für den nächsten Übungstermin und für die Klausur notwendig sind.

Die Abgabe erfolgt online als exakt eine PDF-Datei. Bitte verwenden Sie dabei den Zugriffscode einer vorherigen Abgabe, wenn die Zusammensetzung Ihrer Kleingruppe (**3–5 Studenten**) unverändert geblieben ist. Achten Sie in jedem Fall auf die gleiche Schreibweise der Namen!

<https://svs.informatik.uni-hamburg.de/submission/for/gss15-2>

Bitte benutzen Sie für die Lösung der Aufgaben ergänzend zur Vorlesung folgende Literaturquelle: **Tanenbaum: Moderne Betriebssysteme**. In der Informatik-Bibliothek sind zahlreiche Exemplare in Deutsch und Englisch sowie auch Präsenzexemplare vorhanden (Signatur: P TAN).

Aufgabe 1: Grundlagen von Betriebssystemen (12 Punkte)

Es ist recht schwierig, eine allgemeine und genaue Definition eines Betriebssystems zu finden. Dies liegt u. a. daran, dass Betriebssysteme zwei an sich unabhängige Grundaufgaben durchführen.

- Um welche zwei Grundaufgaben von Betriebssystemen handelt es sich hierbei? Bitte erläutern Sie kurz die jeweils entstehende „Sicht“ auf Betriebssysteme (je max. 2 Sätze).
- Welche konkreten Einzelaufgaben für ein Betriebssystem ergeben sich aus diesen zwei unterschiedlichen Sichten auf Betriebssysteme? Bitte nennen Sie mindestens zwei Einzelaufgaben für jede Sicht (max. 2 Sätze pro Einzelaufgabe).
- (optional) Vergleichen Sie monolithische und hierarchische Organisation von Betriebssystemen. Was sind die Vor- und Nachteile dieser Organisationen?
- (optional) Beschreiben Sie kurz die Unterschiede zwischen Netzwerk-Betriebssystemen und verteilten Betriebssystemen. Gehen Sie dabei auch auf die unterschiedliche Ausprägung der Verteilungstransparenz ein.

Aufgabe 2: Prozesse und Threads (12 Punkte)

- Beschreiben Sie in jeweils max. 2 Sätzen die Begriffe *Programm*, *Prozess* und *Thread* und setzen Sie dabei die Begriffe in Beziehung zueinander.
- (optional) Wie können sich verschiedene Threads gegenseitig negativ beeinflussen?
- (optional) Beschreiben Sie zwei Fälle, in denen es sinnvoll ist, nebenläufige Programmteile mit Hilfe von Threads anstatt von Prozessen zu implementieren
- Ein Prozess durchläuft in seinem Lebenszyklus (siehe Abbildung 1) verschiedene Zustände, beginnend mit der Erzeugung des Prozesses und endend mit dessen Terminierung. Vervollständigen Sie den Zustandsübergangsgraphen indem Sie die fehlenden Zustände X, Y und Z benennen und in jeweils max. 2 Sätzen deren Bedeutung beschreiben. Erläutern Sie außerdem die jeweilige Bedingung (bzw. Ereignis) *a-f*, die zu einem Zustandsübergang führt (max. 2 Sätze pro Bedingung).

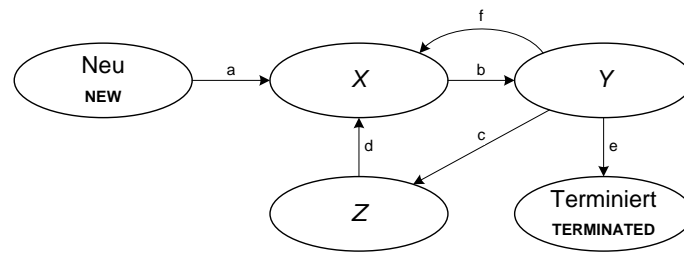


Abbildung 1: Lebenszyklus eines Prozesses

Aufgabe 3: n-Adressmaschine

(16 Punkte)

Der Ausdruck $R = \frac{a_1 + a_2}{a_3} + \frac{b_1 - b_2}{b_3}$ sei direkt (d.,h. ohne Umformung bzw. Optimierung) mit folgenden Maschinenarten zu berechnen:

- eine 2-Adressmaschine (mit Überdeckung des zweiten Operanden).
- (optional) eine 1-Adressmaschine (mit Überdeckung des Akkumulator-Inhaltes, AC).
- (optional) eine 0-Adressmaschine (Keller-Rechner).

Die Benutzung von Hilfsspeicherzellen (keine Register) sei bei der Berechnung erlaubt. Ermitteln Sie – in Anlehnung an die Beispiele für die Arbeitsweise einer n -Adressmaschine im Abschnitt B1b der GSS-Vorlesung – jeweils

- die Befehlsfolge mit zugeordneter Wirkung (bitte skizzieren),
- die Anzahl der Leseaufträge an den Speicher (bei jeweils direkter Adressierung der Hauptspeicherzellen),
- die Anzahl der Schreibaufträge an den Speicher (bei jeweils direkter Adressierung der Hauptspeicherzellen) sowie
- die Berechnungszeit für den Ausdruck (unter den vereinfachenden Annahmen, dass die Ausführungszeit eines Maschinenbefehls nur 5% eines Speicherzugriffs beträgt und die Befehle für Addition, Subtraktion und Division gleich viel Zeit kosten).