Grundlagen der Systemsoftware

SoSe 2018 · Übungsblatt 2 (VSIS 1) · Abgabe: Mittwoch 02.05.2018 (bis 23:59 Uhr)

Bitte beachten Sie, dass die optionalen Aufgaben zwar nicht bewertet werden, jedoch als Vorbereitung für den 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/gss18-2`

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`).

Da bei der Bearbeitung der Aufgaben die eigene Leistung und der Lerneffekt im Vordergrund stehen, sollen stets ausschließlich eigene Formulierungen verwendet werden. Daher sind hier keine Zitate (selbst bei korrekter Quellenangabe) gestattet.

Aufgabe 1: Betriebssysteme und Prozesse (12 Punkte)

a) Betriebssysteme kümmern sich um zwei an sich unabhängige Grundaufgaben. Um welche zwei Grundaufgaben von Betriebssystemen handelt es sich hierbei? Bitte erläutern Sie kurz die jeweils entstehende „Sicht“ auf Betriebssysteme (je max. 2 Sätze).

b) (optional) 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).

c) (optional) Beschreiben Sie in jeweils max. 2 Sätzen die Begriffe *Programm*, *Prozess* und *Thread* und setzen Sie dabei die Begriffe in Beziehung zueinander.

d) 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).

   In der Literatur wird ein entsprechendes Lebenszyklus-Modell u. a. verwendet in:
   Stallings: Betriebssysteme – Prinzipien und Umsetzung

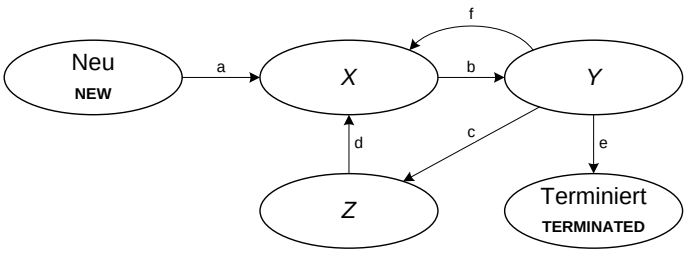


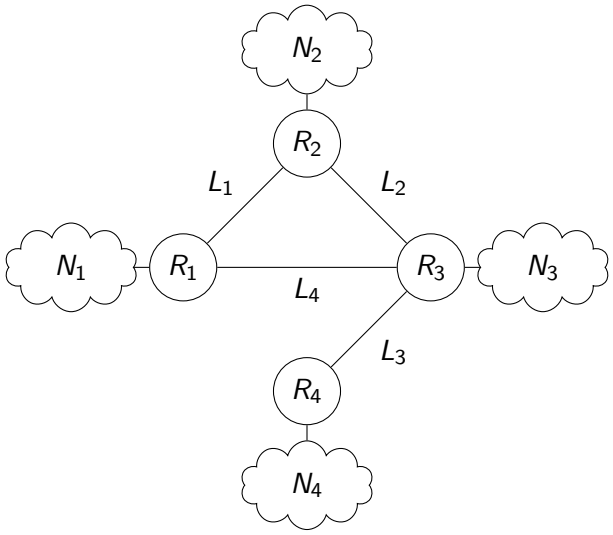
Abbildung 1: Lebenszyklus eines Prozesses

Grundlagen der Systemsoftware

SoSe 2018 · Übungsblatt 2 (VSIS 1) · Abgabe: Mittwoch 02.05.2018 (bis 23:59 Uhr)

Aufgabe 2: Routing in Rechnernetzen (12 Punkte)

Das *Routing Information Protocol* (RIP) ist ein Routing-Protokoll, das innerhalb eines autonomen Systems (z.B. LAN) eingesetzt wird, um die Routingtabellen von Routern automatisch zu erstellen. Die Routing-Entscheidungen werden mit Hilfe von Distanzvektoren (*Distance Vector Routing*) getroffen, wobei als Kostenmaß die Anzahl der Hops verwendet wird. Für ergänzende Erläuterungen verwenden Sie bitte die Literaturquelle: **Coulouris, Dollimore, Kindberg: Distributed Systems: Concepts and Design**. Betrachten Sie für diese Aufgabe das folgende Rechnernetz, bestehend aus den Routern R_1 bis R_4 mit ihren jeweiligen Teilnetzen N_1 bis N_4 sowie den die Router verbindenden Leitungen L_1 bis L_4.



Verwenden Sie für diese Aufgabe den folgenden vereinfachten RIP-Algorithmus:

Send: Whenever local table *Tl* changes, send *Tl* on each nonfaulty outgoing link.
Receive: Whenever a routing table *Tr* is received on link *n*:

```
for all rows Rr in Tr {
  if (Rr.link ≠ n) {
    Rr.cost = Rr.cost + 1;
    Rr.link = n;
    if (Rr.destination is not in Tl) {
      add Rr to Tl; // add new destination to Tl
    } else {
      for all rows Rl in Tl {
        if (Rr.destination = Rl.destination and (Rr.cost < Rl.cost or Rl.link = n)) {
          Rl = Rr; // use remote row, remote has better route or is more authoritative
        }
      }
    }
  }
}
```

a) Nach einer Initialisierungsphase, in der die Router ihre Routing-Tabellen gegenseitig propagieren, bleiben die Routing-Tabellen unverändert, solange die Struktur des Netzes unverändert bleibt. Geben Sie die zu diesem Zeitpunkt stabilen Routing-Tabellen für jeden einzelnen Router an. Gehen Sie dabei von folgenden Initialwerten der Routing-Tabellen aus:

Tabelle von R_1

Ziel	Link	Kosten
N_1	—	0
N_2	?	?
N_3	L_1	2
N_4	?	?

Tabelle von R_2

Ziel	Link	Kosten
N_1	?	?
N_2	—	0
N_3	?	?
N_4	L_1	2

Tabelle von R_3

Ziel	Link	Kosten
N_1	L_2	2
N_2	?	?
N_3	—	0
N_4	?	?

Tabelle von R_4

Ziel	Link	Kosten
N_1	?	?
N_2	L_3	2
N_3	?	?
N_4	—	0

b) (optional) Nehmen Sie nun an, dass die Leitung L_2 zusammenbricht. Zeigen Sie Schritt für Schritt den zeitlichen Ablauf der Routing-Tabellen-Übermittlung, die durch den Ausfall der Leitung L_2 initiiert wurde, indem Sie die Änderungen an den Routing-Tabellen auflisten. Gehen Sie dabei gemäß der Vorlage (siehe unten) rundenbasiert vor, d. h. alle Router modifizieren in einer Runde gleichzeitig ihre jeweilige Routing-Tabelle (falls notwendig) und senden diese danach an die direkten Nachbarn (nur bei Änderungen). Anschließend beginnt die nächste Runde, wobei jeder Router auf Basis der nun von den Nachbarn erhaltenen Routing-Tabellen lokal Änderungen durchführt. Bitte achten Sie darauf, dass Sie zusätzlich zu den Änderungen der Routing-Tabellen auch den Zustand der Routing-Tabellen nach jeder abgeschlossenen Runde angeben.

Vorlage zur Lösung der Aufgabe:
R_2 und R_3 stellen den Ausfall von L_2 fest und ändern ihre Routing-Tabellen:

i) <Runde 1>
TI in R_2: setze Kosten von N_3 auf ∞ (anschl. senden der Tabelle an R_1)
TI in R_3: setze Kosten von ... auf ... (anschl. senden der Tabelle an R_4)

ii) <Runde 2>
R_1 bekommt Tr von R_2 über L_1: setze Kosten von ...
R_4 bekommt Tr von R_3 über ...: setze ...

iii) <Runde 3>...

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:

a) eine 2-Adressmaschine (mit Überdeckung des zweiten Operanden).

b) (optional) eine 1-Adressmaschine (mit Überdeckung des Akkumulator-Inhaltes, AC).

c) (optional) eine 0-Adressmaschine (Keller-Rechner).

Die Benutzung von zusätzlichen Speicherzellen (keine Register) sei bei der Berechnung erlaubt. Die Berechnungsparameter sollen erhalten bleiben. 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 jedes Maschinenbefehls gleich lange dauert und jeder Speicherzugriff die zwanzigfache Zeit wie ein Maschinenbefehl benötigt).