

Tutoriumsblatt 1

Betriebssysteme im WiSe 2020/2021

Zum Modul A

- Tutorium:** Im Rahmen eines einstündigen wöchentlichen Tutoriums werden Aufgaben zur vorangegangenen Vorlesung vorgestellt. Die Aufzeichnung zum Tutorium 1 wird am 4. November 2020 (17 Uhr) online zur Verfügung gestellt.
- Ankündigungen:** Um kurzfristige Ankündigungen nicht zu verpassen, bitten wir Sie regelmäßig die *Website zur Vorlesung* zu besuchen.

Aufgabe T1: Betriebssystem-Schichten

(– Pkt.)

In dieser Aufgabe sollen Sie sich klar machen, wie das Betriebssystem mit den restlichen Komponenten eines Rechners verbunden ist. Finden Sie für die folgenden Begriffe eine sinnvolle Kategorisierung, und stellen Sie die sich ergebenden Rechner-Schichten grafisch dar:

Web-Browser, CPU, Dateisystem, Office-Programme, Ein- und Ausgaberroutinen, Festplatte, Hauptspeicher, Gerätemanagement, Benutzer, Scheduler, Shell, Speicherverwaltung, Unix-Compiler, Drucker, Windows-Systemsteuerung.

Aufgabe T2: Unterprogramme und Stack

(– Pkt.)

In dieser Aufgabe soll das Zusammenspiel von Unterprogrammen und dem Stack untersucht werden. Ein Unterprogramm ist ein Programmstück, welches nur durch den Sprung an seine Anfangsadresse betreten und durch einen Rücksprung an die aufrufende Stelle beendet wird. Diese Art der Programmtechnik wird unter anderem zur Umsetzung rekursiver Aufrufe verwendet. Gegeben das folgende linear rekursive Unterprogramm `mult`:

```
1 int mult(int a, int b) {
2     if(a == 0) {
3         return 0;
4     } else {
5         a = a - 1;
6         return b + mult(a, b);
7     }
8 }
```

Damit man Unterprogramme korrekt ausführen kann muss das Unterprogramm sowie das aufrufende Hauptprogramm bestimmte organisatorische Bedingungen erfüllen. Dazu kann unter anderem ein Stack verwendet werden.

Bearbeiten Sie in diesem Zusammenhang die folgenden Aufgaben:

- Geben Sie eine Abfolge aller Unterprogrammaufrufe mit den entsprechenden Parametern an, die sich für den Aufruf von `mult(2, 3)` ergeben.

- b. Welche Zustandsinformationen müssen auf dem Stack gespeichert werden, damit das Hauptprogramm nach dem Unterprogrammaufruf korrekt fortgesetzt werden kann.
- c. Geben Sie die aus der Vorlesung *Rechnerarchitekturen* bekannte vierstufige Aufrufkonvention an, die ein korrektes Zusammenarbeiten von Hauptprogramm und Unterprogramm mit Hilfe des Stacks gewährleistet. Geben Sie zu jedem der vier Aufrufe an, welche der in Aufgabe b) genannten Zustandsinformationen jeweils verarbeitet werden.
- d. Gehen Sie im Folgenden davon aus, dass der CALL-Befehl für den Aufruf der Funktion `mult` an der Speicheradresse 1000 erfolgt. Die Prozedur `mult` beginnt an der Adresse 4000. Der rekursive Aufruf von `mult` erfolgt an Adresse $(4000+x)$. Skizzieren Sie nun den Stack mit den dazugehörigen Zustandsinformationen unmittelbar vor einem CALL, unmittelbar nach einem CALL und unmittelbar nach einem RET für den Aufruf von `mult(2, 3)`. Geben Sie zudem an, welchen Teil der in Aufgabe c) erläuterten Aufrufkonventionen der gegenwärtige Stackzustand repräsentiert. Sie können davon ausgehen, dass die kleinste adressierbare Einheit auf dem Stack einem Wort (4 Byte) entspricht. Dies entspricht auch der Breite eines Befehls samt seiner Parameter. Geben Sie in jedem Schritt stets den gesamten Inhalt des Stacks an.