

14. Praktikum

Während der Praktikumstermine am 26. und 28.1.2016 findet die Abnahme der APL (Teil 2) statt.

Für die Wartezeit, oder nach erfolgter Abnahme kann die folgende Aufgabe gelöst werden.

Aufgabe:

Erweitern Sie das vorgegebene Programm `p14_vorlage.c` um die Berechnung von arithmetischen Ausdrücken in der Postfix-Notation! Der Postfix-Ausdruck mit Integer-Zahlenwerten und Operatoren (+, -, *, /) wird in der Kommandozeile beim Aufruf des Programms angegeben. Die Kommandozeile kann dann als Parameter `argc` und `argv[]` der `main`-Funktion ausgelesen werden. Der Aufruf

*`meinprog.exe 12 4 * 16 /`*

transferiert 6 Parameter (`argc` ist 6) als einzelne Zeichenketten ("`meinprog.exe`", "`12`", "`4`", "`*`", "`16`" und "`/`") in die `main()`-Funktion. Dabei ist der Name des aufgerufenen Programms die durch 0 indizierte Zeichenkette.

Der oben dargestellte Ausdruck soll den Wert $(12 * 4) / 16$ ausdrücken und 3 ergeben.

Realisieren Sie Ihr Programm mit Hilfe eines Integer-Stacks!

Die Zeichenketten werden mit ansteigendem Index von 1 bis `argc-1` verarbeitet. Operanden (d.h. Zahlenwerte) werden per `PUSH`-Operation auf den Stack abgelegt. Wird ein Operator gelesen, so werden zwei Werte mit der `POP`-Operation aus dem Stack entnommen, entsprechend verknüpft und das Ergebnis wieder mit `PUSH` auf dem Stack abgelegt. Sind alle Zeichenketten verarbeitet, dann kann das Ergebnis mit der `POP`-Operation vom Stack entnommen werden.

Als Reihenfolge der Schritte würde sich für o.g. Beispiel ergeben:

Lese "`12`" -> `PUSH(12)`;

Lese "`4`" -> `PUSH(4)`;

Lese "`*`" -> `POP(&b)`, `POP(&a)`; `erg = a*b`; `PUSH(erg)`;

Lese "`16`" -> `PUSH(16)`;

Lese "`/`" -> `POP(&b)`, `POP(&a)`; `erg = a/b`; `PUSH(erg)`;

`POP(&endwert)`; // `endwert` ist hier 3