

9. Übungsblatt

Aufgabe 1 Virtuelle Maschine

Gegeben sei die nebenstehende Funktion `f` in der Sprache der virtuellen Maschine, die in der Vorlesung vorgestellt wurde. Skizzieren Sie den Ablauf des Programmes für den Fall, daß das einzige Argument der Funktion den Wert 5 hat.

- Wie verändert sich der Stack mit jeder Operation?
- Was berechnet das Programm, wenn der Wert n gegeben ist?

Hinweise:

Um sich den Ablauf zu verdeutlichen, können Sie den Emulator der virtuellen Maschine benutzen, der unter

<http://www.nand2tetris.org/software.php>

zur Verfügung steht. Schreiben Sie dazu die rechts gezeigte Funktion in eine Textdatei `Main.vm`. Die zum Ausführen in dieser Datei zusätzlich benötigte Funktion `main` ist unten links gezeigt, den Inhalt der ebenfalls benötigten Datei `Sys.vm` finden Sie unten rechts.

```
function Main.main 1      function Sys.init 0
  push constant 5        call Main.main 0
  call Main.f 1          label WHILE
  pop local 0            goto WHILE
  return
```

```
function Main.f 2
  push constant 1
  pop local 1
  label LOOP
  push local 0
  push argument 0
  push local 1
  lt
  if-goto END
  push local 1
  push local 1
  push constant 1
  and
  if-goto ADD
  neg
  label ADD
  add
  pop local 0
  push local 1
  push constant 1
  add
  pop local 1
  goto LOOP
  label END
  return
```

Aufgabe 2 Virtuelle Maschine: Fibonacci-Folge

- Nennen Sie jeweils zwei Vor- und zwei Nachteile von virtuellen Maschinen!
- Die Fibonacci-Folge ist eine unendliche Folge von Zahlen, bei der sich die jeweils nächste Zahl durch Addition der beiden vorangehenden Zahlen ergibt:
0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, ... (siehe auch <https://oeis.org/A000045>).
Gegeben sei die folgende, in Java implementierte Methode `fib(int n)` zur rekursiven Berechnung der n -ten Fibonacci-Zahl (wobei die 0-te Fibonacci-Zahl, also die 0 am Anfang der Folge, vernachlässigt wird):

```

public static int fib (int n) {
    if (n <= 2)
        return 1;
    else
        return fib(n-1) + fib(n-2);
}

```

Implementieren Sie eine Funktion `function Main.fib` zur rekursiven Berechnung der n -ten Fibonacci Zahl unter Verwendung der Sprache der virtuellen Maschine des Hack-Systems. Die `Main.fib` erwartet dabei den Parameter `n`, der nach Aufruf von `Main.fib` in `argument[0]` vorliegt.

Hinweis: Vervollständigen Sie die Funktion `Main.fib` im folgenden Programmtext (Inhalt einer Textdatei `Main.vm`; wofür steht eigentlich das k ?):

```

function Main.main 1
    push constant n
    call Main.fib 1
    pop local 0
    return

function Main.fib k
    ...
    Berechnung hier einfügen
    ...
    return

```

Zum Testen benötigen Sie außerdem die Datei `Sys.vm` aus Aufgabe 1.

Aufgabe 3 Virtuelle Maschine: Multiplikation

Schreiben Sie ein Programm in der Sprache der virtuellen Maschine des Hack-Systems, das zwei Zahlen als Parameter hat und deren Produkt ausgibt. Sie können Ihr Programm mit Hilfe des VM-Emulators testen (siehe auch Aufgabe 1).

Syntaxhilfe

Anweisung	Rückgabewert	Kommentar
<code>add</code>	$x + y$	Ganzzahladdition (Zweierkomplement)
<code>sub</code>	$x - y$	Ganzzahlsubtraktion (Zweierkomplement)
<code>neg</code>	$-y$	arithmetische Negation (Zweierkomplement)
<code>eq</code>	-1 falls $x = y$, sonst 0	Test auf Gleichheit
<code>gt</code>	-1 falls $x > y$, sonst 0	Test auf größer
<code>lt</code>	-1 falls $x < y$, sonst 0	Test auf kleiner
<code>and</code>	$x \& y$	bitweises Und
<code>or</code>	$x y$	bitweises Oder
<code>not</code>	$\sim x$	bitweise Negation

Das Diagramm zeigt einen vertikalen Stack-Frame. Von oben nach unten sind die Elemente `...`, `x` und `y` dargestellt. Ein Pfeil von links zeigt auf den unteren Rand des Frames und ist mit `sp` beschriftet, was den Stackpointer anzeigt.