

Anmerkungen zur Abgabe der Hausübung 2

- ¥ Generell wird der Rückgabewert 0 erwartet, falls der Aufruf von `nj vm` erfolgreich war.
- ¥ Alle Binärdateien der Tests haben die Versionsnummer 8. Achten Sie also darauf, dass die Abgabeversion Ihrer `nj vm` auch die Version 8 der Binärdateien akzeptiert.
- ¥ Einige Tests prüfen, ob der Garbage Collector (GC) nicht genügend Speicher bereitstellen kann. In diesem Fall muss der Returnwert 1 sein.
- ¥ Einige Tests prüfen, ob die VM mehr Speicher anfordert als erlaubt. In diesem Fall muss der Returnwert 1 sein.
- ¥ Das gegebene Programm `factor.nj` aus Aufgabe 8, ist einer der umfangreichsten Tests. Wenn dieses Programm bei Ihnen ohne Probleme läuft, dann werden Sie mit hoher Wahrscheinlichkeit auch die HU2 bestehen.
 - ! Bevor Sie den GC implementieren stellen Sie sicher, dass das Programm `factor.nj` bei Ihnen exakt die gleiche Ausgabe erzeugt wie in der Referenzimplementierung.
 - ! Treten nach dem Implementieren des GCs dann Fehler auf, wissen Sie es liegt an der Implementierung vom GC selbst.
- ¥ Der Returnwert 139 bedeutet eigentlich in allen Fällen, dass Ihr Programm einen sog. `segmentation fault` ausgelöst hat. In den Überwiegenden Fällen passiert dies, wenn man eine ungültige Speicheradresse dereferenziert (z.B. `NULL`).
 - ! Prüfen Sie immer darauf, ob `malloc` bzw. Ihre eigene Allokationsfunktion Ihnen eine gültige Speicheradresse zurückgibt (!=NULL)!
- ¥ Beachten Sie in jedem Fall die neuen Programmparameter `--stack` und `--heap` aus Aufgabe 8. Fast alle Testfälle verwenden diese Parameter mit unterschiedlichen Größenangaben. Ihre `nj vm` muss in jedem Fall die eigene Speicherreservierung aus Aufgabe 8 implementieren und darf auch nur so viel Speicher anfordern wie mittels den Parametern angegeben. Achten Sie hierauf ganz besonders, ansonsten schlagen diverse Testfälle fehl!
- ¥ Wenn Sie Ihrem `*ObjRef` zusätzliche Felder hinzugefügt haben, z.B. `forward_pointer`, `typ` und `brokenHeart`, dann achten Sie bei der Implementierung der `*Speicherverwaltung*` und des GCs unbedingt darauf, die Größe von `*ObjRef` korrekt zu berechnen. Je nach Objekttyp ist zusätzlich der im Feld `!size` angegebene Wert unterschiedlich zu interpretieren. Relevante Funktionen sind hier `newPrimitive()` und `copyObjectToFreeMem()` (bzw. je nachdem wie Ihre Funktion hier heißen.)
- ¥ Die BigInt-Library bringt eine eigene Definition für `*ObjRef` mit, die davon ausgeht, dass die Kodierungen (`typ`, `BrokenHeart`) und der `forward_pointer` ausschließlich in der `size`-Komponente erfolgt. Wenn Sie Ihrem `*ObjRef` also zusätzliche Felder hinzugefügt haben, z.B. `forward_pointer`, `typ` oder `brokenHeart`, dann müssen Sie auch der BigInt-Library Ihre Variante bekanntmachen. Ansonsten kann es sein, dass Sie Fehlermeldungen beim Kompilieren erhalten oder Fehler zur Laufzeit auftreten. In jedem Fall muss dafür gesorgt werden, dass sowohl Ihre `nj vm`, als auch die BigInt-Library, das gleiche `*ObjRef` verwendet.
- ¥ Denken Sie daran, dass Ihr Buildscript auch die `bigint`-Bibliothek kompilieren muss.

! Angenommen die **bigint**-Bibliothek befindet sich im Verzeichnis **src/bigint**, dann wäre der Aufruf

```
cd src/bigint; make; cd ../../
```

Anschließend befindet man sich wieder im Ausgangsverzeichnis.

Beispiel: Kompilation der *bigint*-Bibliothek

```
ar@vmr01: [~/KSP_public/hausuebung/njvm]$ ls
mknjvm src
ar@vmr01: [~/KSP_public/hausuebung/njvm]$ ls src/
bigint debug.c debug.h func.c func.h global.c global.h memory.c memory.h
njvm.c readNJBF.c readNJBF.h stackop.c stackop.h support.c
ar@vmr01: [~/KSP_public/hausuebung/njvm]$ cat mknjvm
#!/usr/bin/env bash

cd src/bigint/; make; cd ../../
gcc -g -Wall -pedantic -std=c99 -Isrc/bigint/build/include -Lsrc/bigint/build/lib -o
njvm src/debug.c src/func.c src/global.c src/memory.c src/njvm.c src/readNJBF.c
src/stackop.c src/support.c -lbignum
ar@vmr01: [~/KSP_public/hausuebung/njvm]$ ./mknjvm
for i in src tst ; do \
    make -C $i install ; \
done
make[1]: Entering directory '/home/ar/njvm/src/bigint/src'
mkdir -p ../build/include
cp support.h ../build/include
cp bigint.h ../build/include
mkdir -p ../build/lib
cp libbigint.a ../build/lib
make[1]: Leaving directory '/home/ar/njvm/src/bigint/src'
make[1]: Entering directory '/home/ar/njvm/src/bigint/tst'
mkdir -p ../build/bin
cp testbip ../build/bin
make[1]: Leaving directory '/home/ar/njvm/src/bigint/tst'
ar@vmr01: [~/KSP_public/hausuebung/njvm]$ ls
mknjvm njvm 'README(Submi t).txt' src submit.sh
```