

# KSP Probeklausur

---

Claude Stephane M. Kouame

February 22, 2022

## 1 AUFGABE 1

### 1.1 Part A

```
1 #include <stdio.h>
2
3 int main(int argc, char *argv[]) {
4     return 0;
5 }
```

Was bedeuten hier in dem o.g C-Code **int argc** und **char \*argv[]** ?

### 1.2 Part B

Benennen Sie folgende Deklarationen in deutscher Umgangssprache.

```
1 int argc;
```

argc ist ...

```
1 char str[100];
```

str ist ...

```
1 int *w;
```

w ist ...

```
1 unsigned int y;
```

y ist ...

```
1 char *argv[];
```

argv ist ...

```
1 char **argv2;
```

argv2 ist ...

```
1 void *pt;
```

pt ist ....

### 1.3 Part C

- Wieso benutzt man die Funktion **strcmp()**, um zwei Strings zu vergleichen statt die booleschen vergleichsoperation **==** ?
- Was bedeutet die Signale **0** und **!=0** oder **1** beim Terminieren eines C-programm?
- Beschreiben Sie anhand eines Beispiels, wie man Mehrfachinklusion von Header-Dateien verhindert. Beschreiben Sie außerdem ein Szenario, in dem eine Mehrfachinklusion ohne Schutz davor zu einem Fehler führen würde

## 2 AUFGABE 2

1. In wie viel Bits wird jede Ninja-Instruktion repräsentiert ?, in Wie viel Bereiche werden die aufgeteilt? Geben Sie auch die Namen dieser Bereiche und wie viele Bits werden für jeweils reserviert.
2. Geben Sie die Ninja Eingabe/Ausgabe Instruktionen.
3. Geben Sie für den folgende Ninja-Codeabschnitt eine mögliche Ninja-Assembler code, zeichnen Sie auch dazu den Stackablauf. Hier ist der 'sp=0'. Es gibt kein 'fp'

```
1   writeInteger (6-2*4) ;  
2
```

4. Wo werden in der NinjaVM die ganzen Instruktionen aus einer Ninja bytecode gespeichert ? kreuzen Sie die richtige Antwort.
  - stack
  - heap
  - programm-speicher
  - sda
5. Welche Struktur in der NinjaVM ,die die zentrale Speichereinheit die von allen Instruktionen verwendet wird um Operationen auszuführen ? kreuzen Sie die richtige Antwort.
  - stack
  - heap
  - programm-speicher
  - sda
6. Wofür dienen Stackframes ?

### 3 AUFGABE 3

1. Welche Funktion dient zur Eröffnung einer Datei in C? was sind die mögliche Modi(mode) ?
2. Schreiben Sie einen kleinen C-Programm , um eine Datei 'hello.bin' aufzumachen und zu lesen. Die Datei ist 128 bytes groß, die ersten 80 bytes müssen in einem array von Typ int gespeichert werden und der Rest in einem String. Vergessen nicht am Ende Ihres Programms, die Datei wieder zuzumachen.
3. Welche Funktionen dienen in C zur Speichieranforderung und Speicherfreigabe ?

### 4 AUFGABE 4

1. Definieren Sie ein Makro namens ZWEIHOCH(x), welches eine Zahl darstellt, die den Wert  $2^x$  hat.
2. Folgender Code ist gegeben:

```
1  int i[] = {1, 2, 3, 4, 5};
2  int *p = &i[3];
3  int k = p[-1] + 4;
4  int d = i[*p-2]
5  char c = 'c' + *p;
6
```

Welchen Wert haben

```
1  i[0] =
2  *(p+1) =
3  *p =
4  k=
5  d=
6  c=
7
```

## 5 AUFGABE 5

SDA (Links sind die Index und recht die Werte!)

0	5
1	12
2	6

Schreiben Sie eine mögliche Ninja Abschnitt-code, der die alle Werte aus SDA addiert und das Ergebnis in die Standartausgabe ausgibt.

Schreiben Sie demnach den Ninja Abschnitt-code in Ninja Assembler Code.

## 6 AUFGABE 6

Geben Sie für den folgenden Code einen möglichen Ninja-Abschnitt code.

```
1 _main:
2   asf 2
3   pushc 0
4   popl 0
5   rdint
6   popg 0
7   rdint
8   popl 1
9   jmp _L1
10 _L1:
11   pushl 0
12   pushl 0
13   lt
14   brt _L2
15   rsf
16   halt
17 _L2:
18   pushl 0
19   wrint
20   pushc 10
21   wrchr
22   jmp _L1
```

## 7 AUFGABE 7

1. Definieren eine C-Struktur für einen Baum und eine Linkedlist
2. Wir wollen für unseren C-Code ein Type '**nj\_bool**' benutzen, der die int-Werte 1 und 0 einnehmen kann. Wie kann man dieser Type in C definieren ?

## 8 AUFGABE 8

1. Sie haben zur Verfügung eine Bibliothek names 'bigstring', die zwei wichtige Ordner besitzt : /bigstring/build/include und /bigstring/build/lib. Man möchte aus einer C-Datei 'string.c' eine ausführbare Datei 'string' haben, die mit der 'bigstring' lib kompiliert werden muss.

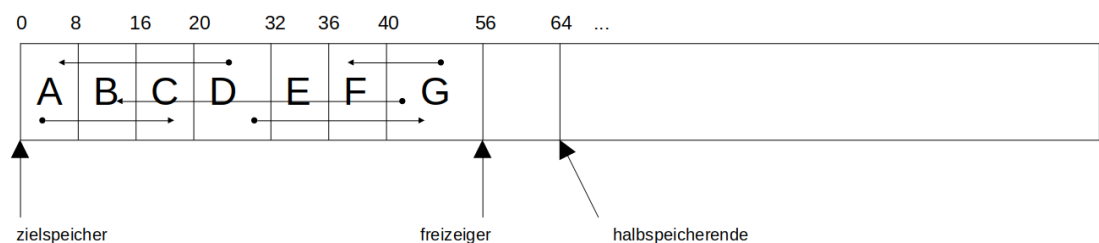
Geben Sie einen Kompilierbefehl, der die ausführbare Datei 'string' erzeugt, dabei muss betrachtet werden, dass alle mögliche Warnungen betreten werden müssen.

2. Wie kann man in Ninja die Objekttypen unterscheiden ? bzw. ein Primitives Objekt von zusammengesetztes Objekt?
3. Definieren Sie ein Makro, zur Erkennung des Objekttype in der NinjaVM.

## 9 AUFGABE 9

### 9.1 PART A

1. Was ist die Aufgabe eines Garbage Collector ? Was bedeutet implizit und explizit Speicherfreigabe ?
2. Geben Sie 2 Verfahren, die für den Garbage Collector verwendet werden.
3. Stop & Copy Verfahren.



In der NinjaVM wurde einen Bin-Datei ausgeführt, mit einer Heap größe von 128 Bytes. Nun ist jetzt der Freizeiger an der Adresse 56, ein neues Objekt von größe 16 bytes muss allokiert werden.

- a) Gibt es noch verfügbaren Plätze für das neue Objekt ? Begründen Sie ihre Antwort.
- b) Wie sieht der Heap nachdem der Garbage Collector ausgelöst wurde , Wenn die Primär Objekte ( root object ) A und D sind Zeichnen Sie.  
Geben Sie den forward\_pointer von A und F nach ihrer Relocate-Phase.
- c) Wie viel bytes hat der Heap frei nachdem der GC ausgelöst wurde ?

## 9.2 PART B

1. Geben sei ein Speicherabbild, wie im Quellhalbspeicher angegeben. Bei den Objekten mit der Nummerierung **1, 2, 3, 6, 8, 9** handelt es sich um primitive Objekte und die Objekte mit der Nummern **4, 5, 7** sind Verbundobjekte. Die **Wurzelobjekte**(Root-Objekte) haben folgende Nummerierung: **1, 4, 6, 7**.
  - Zeichnen Sie in die beiden Zielhalbspeicher die Objekte ein, die nach dem kompletten Durchlauf der jeweiligen (unter dem Speicher angegebenen) Phase/n in den Zielhalbspeicher kopiert wurden.
    - **WICHTIG:** Übernehmen Sie dabei die **Nummerierung** der Objekte !
    - Die Kopien inder Scan-Phase erfolgen aufsteigend nach der Nummerierung (od. von links nach rechts) !
  - Zeichnen Sie alle sog. Forward-Pointer ein.
  - Zeichnen Sie für alle Verbundobjekte in den Zielspeichern die aktualisierten Zeiger ein.  
(Anmerkungen: Der Quellhalbspeicher ist zweimal angegeben, um eine übersichtlichere Darstellung der Forward-Pointer zu ermöglichen).

