Tutorübung Grundlagen: Betriebssysteme und Systemsoftware

Moritz Beckel

München, 28. Oktober 2022

Freitag 10:00-12:00 Uhr

Raum 00.11.038

https://zulip.in.tum.de/#narrow/stream/1295-GBS-Fr-1000-A

Bei Fragen könnt ihr mich hier kontaktieren:

moritz.beckel@tum.de

Organisation

- Hausaufgaben werden nicht besprochen (dafür gibt es extra Programmierstunden)
- Linux wird für Hausaufgaben benötigt (Lehrstuhl empfiehlt Fedora)
- Hausaufgaben werden in Gruppen bearbeitet und abgegeben

Organisation

Geplanter Ablauf der Tutorstunden:

- Kurze Wiederholung wichtiger Inhalte der letzten Woche
- Gemeinsame Bearbeitung der Tutoraufgaben
- Hausaufgaben/Zusatzaufgaben werden nicht besprochen!
- Falls noch Zeit bleibt gegebenenfalls passende Altklausuraufgaben besprechen

Aufgabe 1

```java
import java.util.Scanner;

public class Fakultaet
{
  public static void main (String[] args)
  {
    Scanner scan = new Scanner(System.in);

    int fakultaet = fak(scan.nextInt());
    System.out.println("Fakultaet: " + fakultaet);
  }
  private static int fak(int x) {
      return x <= 1 ? 1 : (x*fak(x-1));
  }
}
```

```c
#include <stdlib.h>
#include <stdio.h>

int fak(int);

int main(int argc, char *argv[]) {
  char *buf = malloc(100 * sizeof(char));
  fgets(buf, 100, stdin);

  int fakultaet = fak(atoi(buf));
  printf("Fakultaet: %d\n", fakultaet);
  free(buf);
}

int fak(int x) {
  return x <= 1 ? 1 : (x*fak(x-1));
}
```

Aufgabe 1

```java
import java.util.Scanner;

public class Fakultaet
{
  public static void main (String[] args)
  {
    Scanner scan = new Scanner(System.in);

    int fakultaet = fak(scan.nextInt());
    System.out.println("Fakultaet: " + fakultaet);
  }
  private static int fak(int x) {
      return x <= 1 ? 1 : (x*fak(x-1));
  }
}
```

```c
#include <stdlib.h>
#include <stdio.h>

int fak(int);

int main(int argc, char *argv[]) {
  char *buf = malloc(100 * sizeof(char));
  fgets(buf, 100, stdin);

  int fakultaet = fak(atoi(buf));
  printf("Fakultaet: %d\n", fakultaet);
  free(buf);
}

int fak(int x) {
  return x <= 1 ? 1 : (x*fak(x-1));
}
```

Aufgabe 2

Binär			Dezimal
Kibi	$1024 = 2^{10}$	$1000 = 10^{3}$	Kilo
Mebi	2^{20}	10^{6}	Mega
Gibi	2^{30}	10^{9}	Giga
Tebi	2^{40}	10^{12}	Tera

Aufgabe 2

Binär			Dezimal
Kibi	$1024 = 2^{10}$	$1000 = 10^{3}$	Kilo
Mebi	2^{20}	10^{6}	Mega
Gibi	2^{30}	10^{9}	Giga
Tebi	2^{40}	10^{12}	Tera

a) Übersetzen Sie von Binär- zu Dezimalpräfix: 2 KiB, 3 MiB, 4 GiB

Aufgabe 2

Binär			Dezimal
Kibi	$1024 = 2^{10}$	$1000 = 10^{3}$	Kilo
Mebi	2^{20}	10^{6}	Mega
Gibi	2^{30}	10^{9}	Giga
Tebi	2^{40}	10^{12}	Tera

a) Übersetzen Sie von Binär- zu Dezimalpräfix: 2 KiB, 3 MiB, 4 GiB

- 2KiB = 2,048KB
- 3MiB = 3,145MB
- 4GiB = 4,294GB

Aufgabe 2

Binär			Dezimal
Kibi	$1024 = 2^{10}$	$1000 = 10^{3}$	Kilo
Mebi	2^{20}	10^{6}	Mega
Gibi	2^{30}	10^{9}	Giga
Tebi	2^{40}	10^{12}	Tera

b) Übersetzen Sie von Dezimal- zu Binärpräfix: 4 kB, 3 MB, 2 GB.

Aufgabe 2

Binär			Dezimal
Kibi	$1024 = 2^{10}$	$1000 = 10^{3}$	Kilo
Mebi	2^{20}	10^{6}	Mega
Gibi	2^{30}	10^{9}	Giga
Tebi	2^{40}	10^{12}	Tera

b) Übersetzen Sie von Dezimal- zu Binärpräfix: 4 kB, 3 MB, 2 GB.

- 4KB = 3,906KiB
- 3MB = 2,861MiB
- 2GB = 1,862GiB

Aufgabe 2

Binär			Dezimal
Kibi	$1024 = 2^{10}$	$1000 = 10^{3}$	Kilo
Mebi	2^{20}	10^{6}	Mega
Gibi	2^{30}	10^{9}	Giga
Tebi	2^{40}	10^{12}	Tera

c) Hersteller von Speichermedien preisen diese mit Kapazitäten an, die auf Dezimalpräfixen basieren. Das führt häufig zu Verwirrung, da Software meist Binärpräfixe verwendet, jedoch die falschen Einheiten anzeigt (z.B. GB statt GiB). Wie viel Speicherplatz wird dem Käufer einer externen 2 TB-Festplatte nach dem Anschließen ein solches System angezeigt?

Aufgabe 2

Binär			Dezimal
Kibi	$1024 = 2^{10}$	$1000 = 10^3$	Kilo
Mebi	2^{20}	10^6	Mega
Gibi	2^{30}	10^9	Giga
Tebi	2^{40}	10^{12}	Tera

c) Hersteller von Speichermedien preisen diese mit Kapazitäten an, die auf Dezimalpräfixen basieren. Das führt häufig zu Verwirrung, da Software meist Binärpräfixe verwendet, jedoch die falschen Einheiten anzeigt (z.B. GB statt GiB). Wie viel Speicherplatz wird dem Käufer einer externen 2 TB-Festplatte nach dem Anschließen ein solches System angezeigt?

- 2TB = 1,82TiB

Aufgabe 3

a) Binär → Hex

0b101010

0b111000111

0b1100000011011110

Aufgabe 3

a) Binär → Hex

0b101010	0x2A
0b111000111	0x1C7
0b1100000011011110	0xC0DE

Aufgabe 3

b) Hex → Binär

0xAFFE

0xBADE

0xC0FFEE

Aufgabe 3

b) Hex → Binär

0xAFFE	0b1010 1111 1111 1110
0xBADE	0b1011 1010 1101 1110
0xC0FFEE	0b1100 0000 1111 1111 1110 1110

Aufgabe 3

c) Dezimal → Hex

123

65

262

Aufgabe 3

c) Dezimal → Hex

123	0x7B
65	0x41
262	0x106

Aufgabe 3

d) Dezimal → Binär

255

99

54

Aufgabe 3

d) Dezimal → Binär

255	0b1111 1111
99	0b0110 0011
54	0b0011 0110

Aufgabe 3

e) Hex → Dezimal

0xABC

0x64

0x420

Aufgabe 3

e) Hex → Dezimal

0xABC	2748
0x64	100
0x420	1056

Aufgabe 4

```
1   int arrayXYZ[10] = {0};
2   int i = 0, intVar = 0;
3   int *pi = 0;
4   for (i = 0; i < 10; i++)
5       arrayXYZ[i] = i;
```

Aufgabe 4

```
1  int arrayXYZ[10] = {0};
2  int i = 0, intVar = 0;
3  int *pi = 0;
4  for (i = 0; i < 10; i++)
5      arrayXYZ[i] = i;
```

a) Was ist der Inhalt der Variable pi jeweils nach den folgenden Statements?

```
1  pi = &arrayXYZ[7];
2  pi = arrayXYZ;
3  pi = &arrayXYZ[0];
```

Aufgabe 4

```
1  int arrayXYZ[10] = {0};
2  int i = 0, intVar = 0;
3  int *pi = 0;
4  for (i = 0; i < 10; i++)
5      arrayXYZ[i] = i;
```

b) Sind die folgenden Zuweisungen äquivalent? Verdeutlichen Sie sich, wie ein Array-Zugriff in C umgesetzt wird.

```
1  intVar = arrayXYZ[8];
2  intVar = *(arrayXYZ+8);
3  intVar = *(int *) ((void *) arrayXYZ+(8*sizeof(int)));
```

Aufgabe 4

```
1  int arrayXYZ[10] = {0};
2  int i = 0, intVar = 0;
3  int *pi = 0;
4  for (i = 0; i < 10; i++)
5      arrayXYZ[i] = i;
```

b) Sind die folgenden Zuweisungen äquivalent? Verdeutlichen Sie sich, wie ein Array-Zugriff in C umgesetzt wird.

```
1  intVar = arrayXYZ[8];
2  intVar = *(arrayXYZ+8);
3  intVar = *(int *) ((void *) arrayXYZ+(8*sizeof(int)));
```

Alle Zuweisungen sind äquivalent.

Aufgabe 4

```
1  int arrayXYZ[10] = {0};
2  int i = 0, intVar = 0;
3  int *pi = 0;
4  for (i = 0; i < 10; i++)
5      arrayXYZ[i] = i;
```

c) Kompilieren die folgenden Statements ohne Warnings oder Errors?

```
1    i = pi;
2    intVar = i;
3    *(arrayXYZ+3) = 3;
4    arrayXYZ[1320] = "a";
5    arrayXYZ[1] = &*(arrayXYZ + 15);
```

Aufgabe 4

```
1  int arrayXYZ[10] = {0};
2  int i = 0, intVar = 0;
3  int *pi = 0;
4  for (i = 0; i < 10; i++)
5      arrayXYZ[i] = i;
```

c) Kompilieren die folgenden Statements ohne Warnings oder Errors?

```
1    i = pi;
2    intVar = i;
3    *(arrayXYZ+3) = 3;
4    arrayXYZ[1320] = "a";
5    arrayXYZ[1] = &*(arrayXYZ + 15);
```

Aufgabe 4

d) Was macht der folgende Code? Wofür wird malloc in C benutzt?

```
int *array123 = malloc(5 * sizeof(int));
```

Aufgabe 4

d) Was macht der folgende Code? Wofür wird malloc in C benutzt?

```
int *array123 = malloc(5 * sizeof(int));
```

- Speicher wird auf dem Heap reserviert
- Speicher wird unabhängig vom Stack Frame nicht freigegeben
- muss manuell freigegeben werden mit free(array123);

Aufgabe 4

e) Welche Rolle spielt der Operator sizeof()? Wieso wird an malloc() nicht 5 als Parameter übergeben?

Aufgabe 4

e) Welche Rolle spielt der Operator sizeof()? Wieso wird an malloc() nicht 5 als Parameter übergeben?

- malloc() erwartet Angabe in Bytes
- sizeof() gibt die Größe jeweiliger in C definierter Typen in Bytes an

Aufgabe 4

f) Was hat es mit free() auf sich? Inwiefern verhält sich Java hier anders als C, wieso musste man diese Funktion in Java nicht nutzen?

Aufgabe 4

f) Was hat es mit free() auf sich? Inwiefern verhält sich Java hier anders als C, wieso musste man diese Funktion in Java nicht nutzen?

- malloc() reservierter Speicher muss manuell freigegeben werden durch free()
- Java benutzt Garbage Collector, dieser automatisiert diesen Prozess