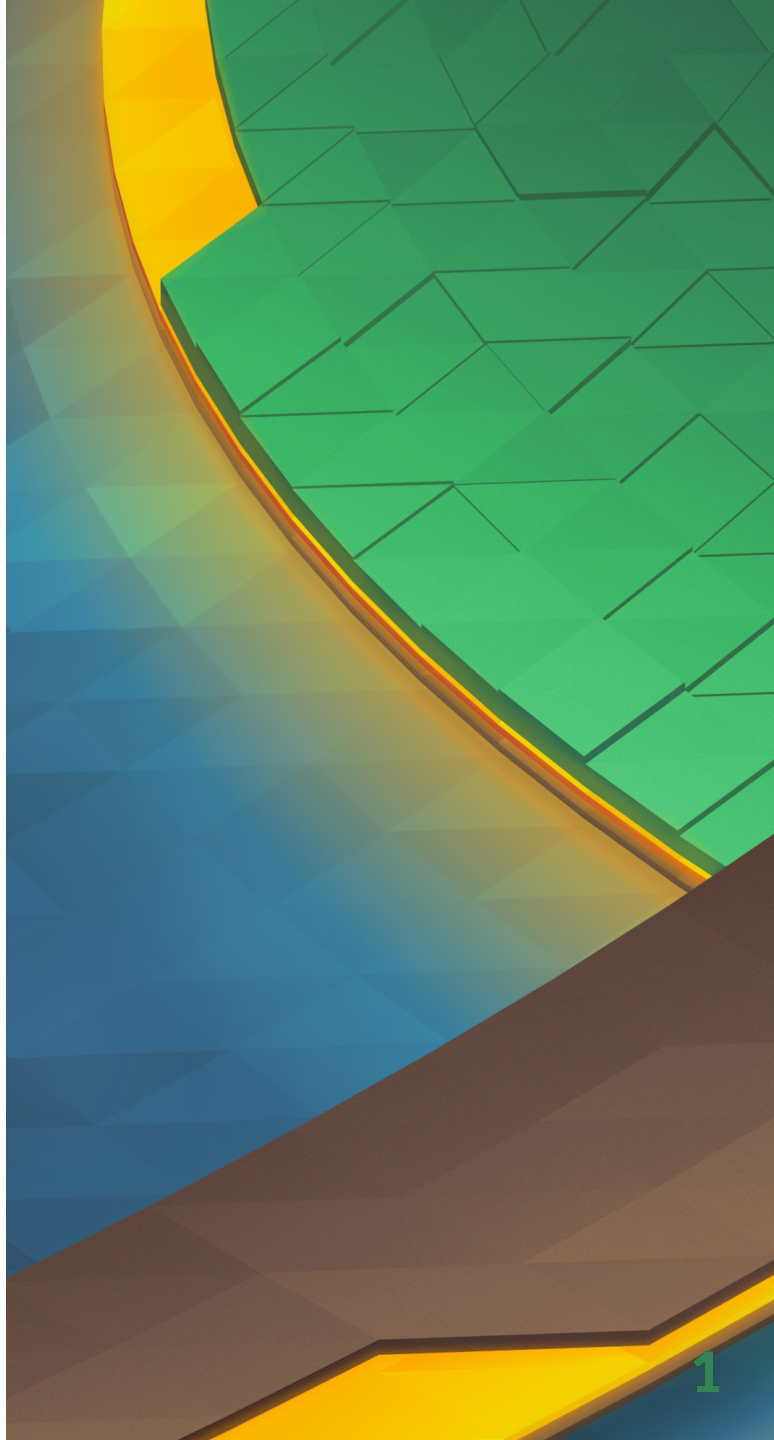
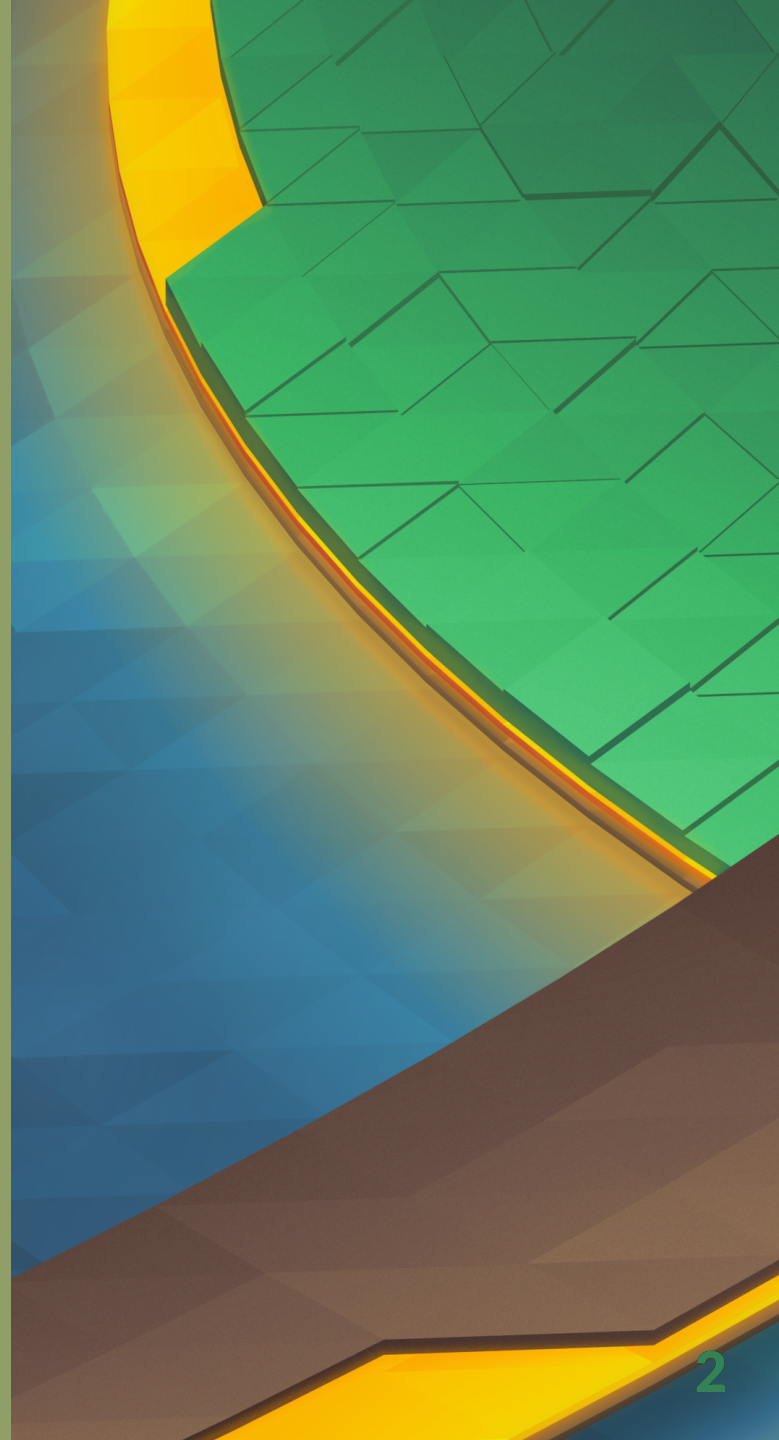
Tutorat 7

Zeiger Structs und Stackframes



Betriebssysteme, Tutorat 7, Gruppe 6, juergmatth@gmail.com, Universität Freiburg

Vorbereitungen



Vorbereitungen

GDB

- **Stack ausgeben:** `x /40x $sp`
- `x /20x &p2` (muss Adresse sein)
- **Heap ausgeben:** `x /20x 0x555555559000`
- dump memory file.dump 0xstart 0xstop
- **find out heap location:**
  - find out <pid> with `info inferior`
  - find out heap adress with `info proc mapping <pid>`
- process in `proc/<pid>/maps`
  - pidof, kill -9 <pid>, ps -u <user>, top -U <user>

Vorbereitungen

Hexadezimalsystem

- $$\underline{beef}_{16} = 11 * 16^3 + 14 * 16^2 + 14 * 16^1 + 15 * 16^0$$
  $$= 11 * 4096 + 14 * 256 + 14 * 16 + 15$$
  $$= 48879$$

Hexadezimalsystem $\longrightarrow$ Binärsystem

```
   D    4    F    6    6    E
1101 0100 1111 0110 0110 1110
```

Vorbereitungen
Hexadezimalsystem

Binärsystem $\longrightarrow$ Hexadezimalsystem

```
1101 0100 1111 0110 0110 1110
   D    4    F    6    6    E
```

Vorbereitungen

Hexadezimalsystem

- 8 Bit sind 2 Hexzahlen, also 1 Byte=1char wird durch 2 Hexzahlen dargestellt
  - hex(0b11110000)
  - bin(0xf0)
  - immer 4er Bitvektoren lassen sich direkt in eine Hexzahl übersetzen
- 32 Bit sind 8 Hexzahlen, da 32 Bit=4 Byte und 1Byte entspricht 2 Hexzahlen
  - hex(0b11110000_00001111_00010010_00110100)
  - bin(0xf00f1234)
- bytes actually get read backwards because x86 is **little-endian** and bytes are the smallest unit -> 0xa0 0x92 0x55 0x55 0x55 0x55 0x00 0x00 gets read 0x5555555592a0

Vorbereitungen

Hexadezimalsystem

0	1	2	3	4	5	6
0000	0001	0010	0011	0100	0101	0110

7	8	9
0111	1000	1001

A	B	C	D	E	F
1010	1011	1100	1101	1110	1111

Vorbereitungen
Pointerarithmetik

Vorbereitungen

Stackframes

Vorbereitungen

Ceil und Floor

Abrundungsfunktion (floor)

- $\lfloor 2,8 \rfloor = 2, \lfloor -2,8 \rfloor = -3, \lfloor -2,2 \rfloor = -3, \lfloor 2 \rfloor = 2$
- $\lfloor x \rfloor \leq x < \lfloor x \rfloor + 1$
- $\lfloor x \rfloor = x \Leftrightarrow x \in \mathbb{Z}$
- $\lfloor x + k \rfloor = \lfloor x \rfloor + k, \quad k \in \mathbb{Z}$

Aufrundungsfunktion (ceil)

- $\lceil 2,8 \rceil = 3, \lceil -2,8 \rceil = -2, \lceil -2,2 \rceil = -2, \lceil 2 \rceil = 2$
- $\lceil x \rceil - 1 < x \leq \lceil x \rceil$
- $\lceil x \rceil = x \Leftrightarrow x \in \mathbb{Z}$
- $\lceil x + k \rceil = \lceil x \rceil + k, \quad k \in \mathbb{Z}$

Vorbereitungen

O-Notation

- Relationen $\mathrm{O}, \Omega, \Theta, o, \omega$ sind auf Funktionen, was die Relationen $\leq, \geq, =, <, >$ auf Zahlen sind
- O entspricht $\leq$:
  - $\mathrm{O}(f) = \{g : \mathbb{N} \to \mathbb{R} \mid \exists n_0 \in \mathbb{N} \quad \exists C > 0 \quad \forall n \geq n_0 \quad g(n) \leq C \cdot f(n)\}$
    - oder auch $|f_n| \leq C\,|g_n| \quad \forall n \geq n_0$
  - $f_n = O(g_n)$ bedeutet "$\mathrm{f(x)}$ $\mathrm{wächst\ höchstens\ wie}$ $\mathrm{g(x)}$"
    - $f_n = O(g_n)$ ist keine Gleichheit im mathematischen Sinne, präziser wäre $f_n \in O(g_n)$ als Menge aller Funktionen, die von der Ordnung g_n sind
  - $f = O(g) \Leftrightarrow \lim_{n \to \infty} f(n)/g(n) < \infty$

Vorbereitungen

O-Notation

- Ω entspricht $\geq$:
  - $\Omega(f) = \{g : \mathbb{N} \to \mathbb{R} \mid \exists n_0 \in \mathbb{N} \quad \exists C > 0 \quad \forall n \geq n_0 \quad g(n) \geq C \cdot f(n)\}$
  - $f_n = O(g_n)$ bedeutet "$\mathrm{f(x)}$ wächst mindestens wie $\mathrm{g(x)}$"
  - $f = \Omega(g) \Leftrightarrow \lim_{n \to \infty} f(n)/g(n) > 0$
- Θ entspricht $=$:
  - $\Theta(f) = O(f) \cap \Omega(f)$
  - $f_n = \Theta(g_n)$ bedeutet "$\mathrm{f(x)}$ wächst genauso wie $\mathrm{g(x)}$"
  - $f = \Theta(g) \Leftrightarrow \lim_{n \to \infty} f(n)/g(n) > 0 \text{ und } < \infty$

Vorbereitungen

O-Notation

- o entspricht $<$:
  - $o(f) = \{g : \mathbb{N} \to \mathbb{R} \mid \exists n_0 \in \mathbb{N} \quad \forall C > 0 \quad \forall n \geq n_0 \quad g(n) \leq C \cdot f(n)\}$
  - $f_n = o(g_n)$ bedeutet "$\mathrm{f(x)}$ wächst (strikt) langsamer als $\mathrm{g(x)}$"
  - $f = o(g) \Leftrightarrow \lim_{n \to \infty} f(n)/g(n) = 0$
- ω entspricht $>$:
  - $\omega(f) = \{g : \mathbb{N} \to \mathbb{R} \mid \exists n_0 \in \mathbb{N} \quad \forall C > 0 \quad \forall n \geq n_0 \quad g(n) \geq C \cdot f(n)\}$
  - $f_n = \omega(g_n)$ bedeutet "$\mathrm{f(x)}$ wächst (strikt) schneller als $\mathrm{g(x)}$"
  - $f = \omega(g) \Leftrightarrow \lim_{n \to \infty} f(n)/g(n) = \infty$

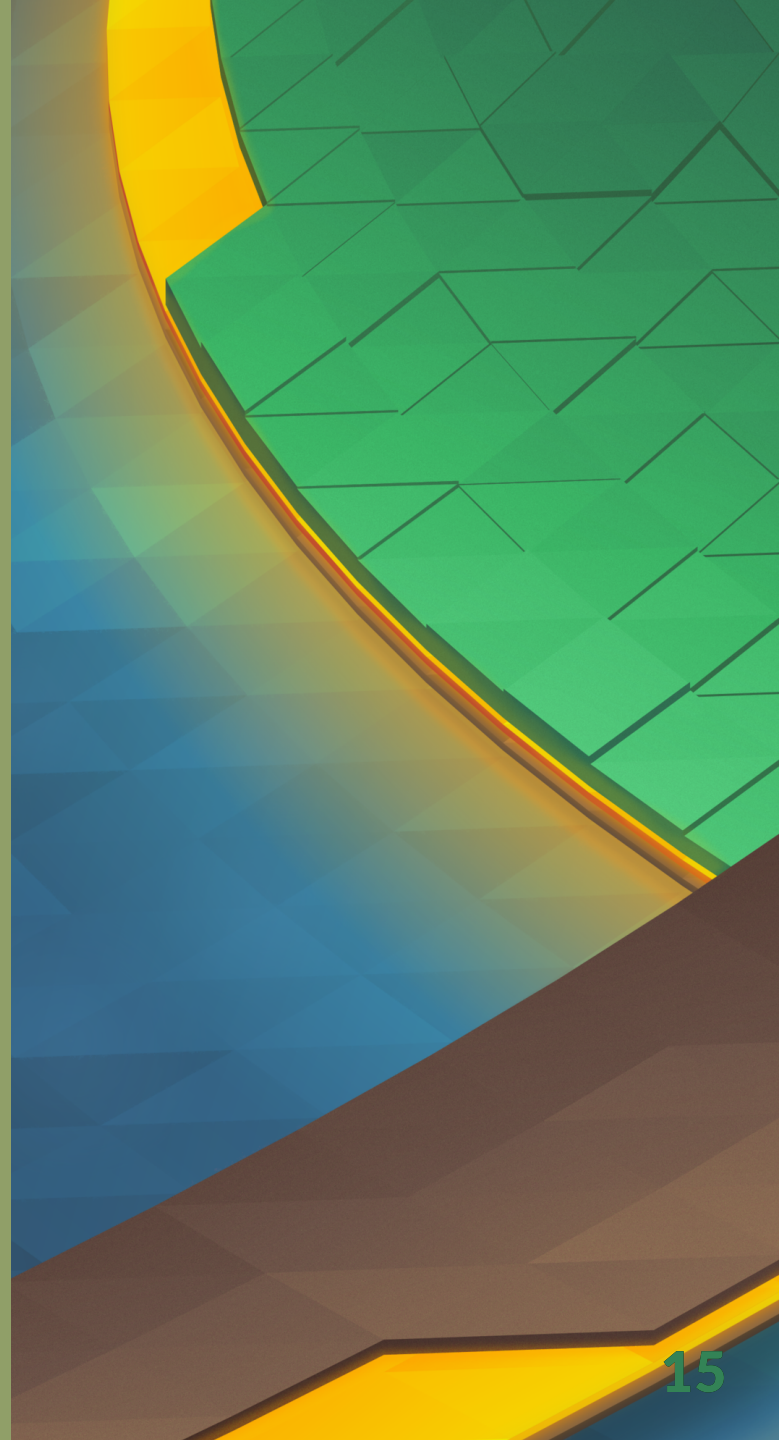
Vorbereitungen

O-Notation

Wachstum elementarer Folgen

- $1 = O\left(\log_a(n)\right) \quad \text{für } a > 0, a \neq 1$
- $\log_a(n) = O\left(n^b\right) \quad \text{für } a > 0, a \neq 1, b > 0$
- $n^{b_1} = O\left(n^{b_2}\right) \quad \text{für } 0 \leq b_1 \leq b_2$
- $n^b = O\left(a^n\right) \quad \text{für } b \geq 0, a > 1$
- $a_1^n = O\left(a_2^n\right) \quad \text{für } 0 < a_1 \leq a_2$
- $a^n = O(n!) \quad \text{für } a > 0$
- $n! = O\left(n^n\right)$

Übungsblatt



Übungsblatt
Aufgabe 1

```
#include <stdlib.h> //Stellt Bibliotheksfunktionen malloc() und free() bereit
void main()
{
  struct point
  {
    int x;
    int y;
  };

  // Annahme Symboltabelleneintrag st(p1) = (var, struct point*, 8)
  struct point *p1;
  // Annahme Symboltabelleneintrag st(p3) = (var, struct point*, 9)
  struct point *p3;
  // Annahme Symboltabelleneintrag st(a) = (var, int*, 10)
  int* a;
```

Übungsblatt
Aufgabe 1

```
//Annahme Symboltabelleneintrag:
//st(p2) = (struct, x -> (int,0), y -> (int,1), 15)
struct point p2;
a = &(p2.x);
p2.x = 7;
p2.y = 4;
/*** MARKE 1 ***/

//Annahme: reserviert zusammenhaengenden Bereich auf dem Heap ab Adresse 33
p1 = (struct point *) malloc(sizeof(struct point));

(*p1).y = *a;

p3 = p1;
p1 = &p2;
/*** MARKE 2 ***/
```

Übungsblatt
Aufgabe 1

```
  if((*p1).y > 5)
  {
    *a = 42;
  }
  else
  {
    *a = 1;
  };
  /*** MARKE 3 ***/

  free(p3);
};
```

Übungsblatt

Aufgabe 1a)

- 8, 9, 10, 15, 16, 34
- **Anmerkung zu 33:** 33 wird nicht gelesen, da für die **Ponterarithmetik** nur die Adresse wichtig ist und die steht in Speicherzelle 10, wo der Zeiger `p1` haust

Übungsblatt

Aufgabe 1b)

- **Marke 1:**

```
8: undefiniert   33: undefiniert
9: undefiniert   34: undefiniert
10: 15
15: 7
16: 4
```

- **Marke 2:**

```
8: 15   33: undefiniert
9: 33   34: 7
10: 15
15: 7
16: 4
```

Übungsblatt
Aufgabe 1b)

- **Marke 3:**

```
8: 15   33: undefiniert
9: 33   34: 7
10: 15
15: 1
16: 4
```

Übungsblatt

Aufgabe 1c)

- Speicheradressen 33 und 34 werden wieder freigegeben
- `p3` zeigt auf den Speicherbereich, der mit `malloc()` reserviert wurde

Übungsblatt

Aufgabe 2a

```
int fib(int n)
{
  int res_f, a, b;
  if (n==0)
    res_f = 0;
  else if (n==1)
    res_f = 1;
  else /* Fall: n > 1. */ {
    a = fib(n-1); // Ruecksprungadresse 200
    b = fib(n-2); // Ruecksprungadresse 300
    res_f = a + b;
  };
  return res_f;
}
```

Übungsblatt

Aufgabe 2b)

- siehe `aufgabe_2b.csv` auf dem **Nextcloud-Server**

Übungsblatt
Aufgabe 2c)

Nachteil

- Teilergebnisse **mehrfach** berechnet → #Funktionsaufrufe wächst **exponentiel** → ineffizient

Abschätzug für #Funktionsaufrufe

- $\forall n \geq 4 : f(n) = f(n-1) + f(n-2) \geq 2 \cdot f(n-2) \overset{*}{\Rightarrow} t(n) \geq 2^{\lfloor n/2 \rfloor}$

  $\star : f(n) > 2f(n-2)$

  $\geq 2 \cdot 2f(n-2 \cdot 2)$
- $\dots$

  $\geq 2^k f(n-2k)$

Übungsblatt

Aufgabe 2c)

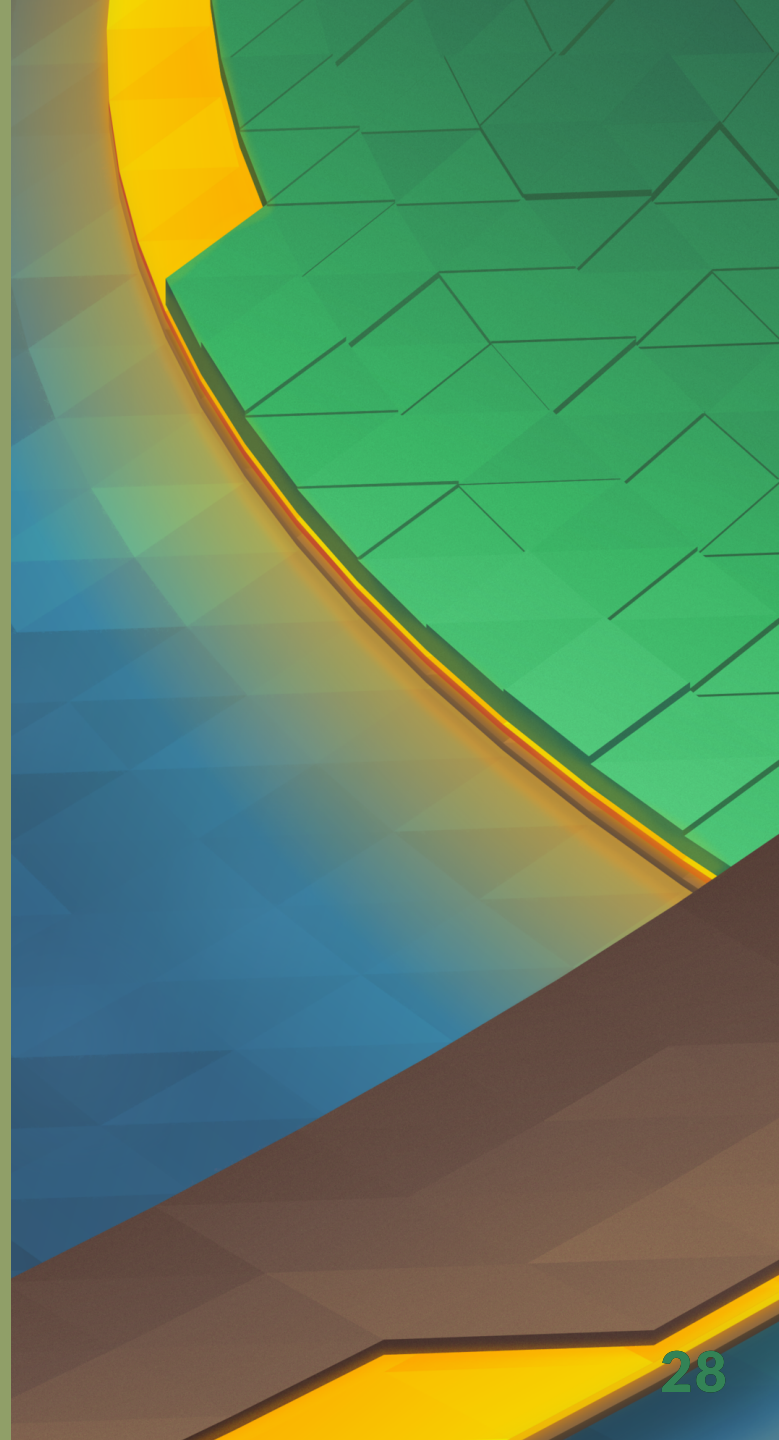
- $n/2$ Verdopplungen, da wir in jedem Schritt $n-2$ nehmen. Für ungerade n muss abgerundet werden
- Wähle nun $k_0 := \left\lfloor \frac{n-1}{2} \right\rfloor$, also $k_0 = \frac{n-1}{2}$ falls n ungerade und $k_0 = \frac{n-2}{2}$, falls n gerade. Dann ist $n - 2k_0 = 1$ oder $n - 2k_0 = 2$ und somit $f\left(n - 2k_0\right) = 1$.
- **Es folgt:**
- $\forall n \geq 4: \quad f(n) > 2^{k_0} f\left(n - 2k_0\right) = 2^{\left\lfloor \frac{n-1}{2} \right\rfloor}$
  - Da $t(n) = f(n+1)$ gilt für $n \geq 4$:
  - $t(n) > 2^{\left\lfloor \frac{n}{2} \right\rfloor} \geq 2^{\frac{n}{2}-1} = \frac{2^{\frac{n}{2}}}{2^1} = 0.5\sqrt{2}^n$
    also $t(n) = \Omega\left((\sqrt{2})^n\right)$

Übungsblatt

Aufgabe 2d)

```
int fib_efficient(int n, int* res) {
  if (n == 0) {
    return 0;
  }
  else if (n == 1) {
    res[0] = 0;
    res[1] = 1;
    return 1;
  }
  res[n] = fib_efficient(n-1, res) + res[n-2];
  return res[n];
}
```

Ergänzungen



Ergänzungen

Addition binär und dezimal

```
  011011 (27)          17718
+ 011101 (29)        +  6524
  11111                 11 1
  ======               ======
  111000 (56)          24242
```

```
00 + 00 = 00          00 + 00 (+ 01) = 01
00 + 01 = 01          00 + 01 (+ 01) = 10
01 + 00 = 01          01 + 00 (+ 01) = 10
01 + 01 = 10          01 + 01 (+ 01) = 11
```

Ergänzungen

Subtraktion binär und dezimal (nicht empfohlen, dient Vergleich mit nächster Folie)

```
(1)
  0111000 (56)          24242
- 0011011 (27)        - 17718
   11111                 11 1
  =======               ======
  0011101 (29)           6524
```

```
10 - 00 = 10                    10 - 00 (- 01) = 01
10 - 01 = 01                    10 - 01 (- 01) = 00
11 - 00 = 11                    11 - 00 (- 01) = 10
11 - 01 = 10                    11 - 01 (- 01) = 01
```

Ergänzungen

Subtraktion binär und dezimal (funktioniert immer, egal was für Vorzeichen Zahlen haben)

```
(2)
   0111000 (56)
 + 1100101 (27) (0011011 negiert und +1)
  11
   =======
   0011101 (29)
```

- **Zweierkomplement Negation:** `11011 -> 011011 -> 100100 -> 100101`
  - `0`en hinzufügen bis **Minuend** und **Subtrahend** beide gleiche Länge haben und Platz für ihr **Vorzeichenbit** ist und dieses korrekt gesetzt ist
  - **1er Komplement Negation** und `+1` nicht vergessen für den **Subtrahenden**

Ergänzungen
Multiplikation binär und dezimal

```
1101 x 1001 (13 * 9)          1304 x 12
       1101                          48
      0000                     +    0
     0000                      + 36
    1101                       +12
   ========                    ======
    1110101 (117)               15648
```

- **Verschiebung** ist aufgrund der `0`en, die hier ausgelassen sind

Ergänzungen
Division binär

```
  1110101 / 1011 (117 : 11) = 1010 (10) Rest: 111 (7)
- 1011|||
 =====|||
    111||
-     0||
   ====||
    1110|
-   1011|
   =====|
      111
-       0
     ====
      111
```

Ergänzungen
Division dezimal

```
15658 / 12 = 1304,833...
12|||
==|||
 36||
 36||
 ==||
  05|
   0|
  ==|
   58
   48
    ...
```

Ergänzungen
Division dezimal

```
==
10|0  oder Rest: 10
 9 6
 ===
   40
   36
   ==
    40
    36
    ==
     4...
```

Ergänzungen

Division binär

- bei **binärer Division** gibt es nur **2 Zuständen** (`1` oder `0`), dementsprechend wird entweder die Zahl so übernommen (Zahl · `1`) oder die Zahl ist `0` (Zahl · `0`)

Division allgemein

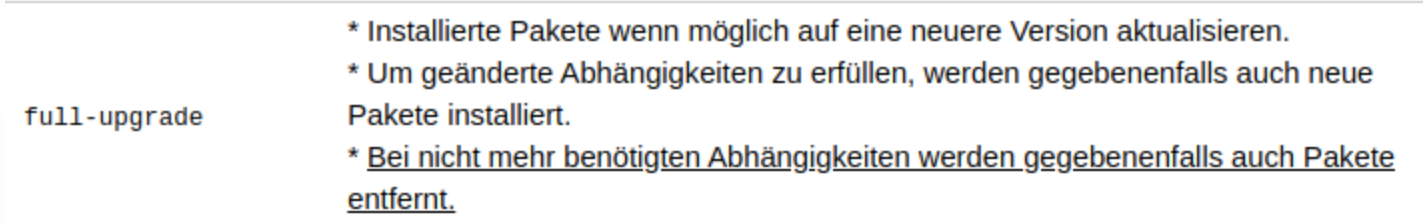
- nach jeder Addition ein Zahl runterholen, bis keine mehr runtergeholt werden kann $\rightarrow$ dann Ende (bei **ganzzahliger Division**). Was unten stehen bleibt ist der **Rest**
- bei Division mit Nachkommastellen, 0en runterbringen, bis einmal **kein Rest** mehr rauskommt oder Grenze setzen bis zu der man weiter macht $\rightarrow$ dann Ende
- ist der **Dividend** trotz runtergebrachter weiter Stelle (weil einmal kein Rest übrig blieb) immernoch kleiner als der **Divisor**, so ist der **Quotient** 0, weil nur durch $\cdot 0$ rechnen kann der **Divisor** noch kleiner sein als der **Dividend**

Ergänzungen

Packages installieren mit `apt`

updating

- `sudo apt update`: update package lists
- `sudo apt update -y && sudo apt full-upgrade`:



- `sudo apt update -y && sudo apt full-upgrade qutebrowser`: update a program

“ • `full-upgrade` is the recommended way over `upgrade` ”

Ergänzungen

Packages installieren mit `apt`

installing

- `sudo apt update -y && sudo apt install gcc -y`: install package from repo
- `sudo apt update -y && sudo apt install ./foo_1.0_all.deb -y`: install local package

removing

- `sudo apt update -y && sudo apt purge gcc -y`: uninstalls package, es werden alle Konfigurationsdateien gelöscht
- `sudo apt update -y && sudo apt autoremove -y` uninstalls all packages, that are not needed anymore and have no dependencies to other packages

“ • `purge` is the recommended way over `remove` ”

Ergänzungen

Packages installieren mit `apt`

searching

- autocomplete application name, e.g. `sudo apt install openjdk`, double tab
- `apt list gcc`: lists als packages with which fit the search term
- `apt list gcc --installed`: only list packages that are installed
- `apt show gcc`: shows desciption of package matching the search term
- `apt search gcc`: lists alls packages which the search term in their discription or name

“ • glob-pattern or regex as search pattern ”

Ergänzungen

Packages installieren mit `apt`

other

- `sudo apt download emacs`: download `.deb`-package
- `sudo apt install alacritty -y`: no `y` each time
- `sudo do-release-upgrade`: upgrade **Distro** to a newer release

“
- instead of confirming with y, once can also just spam enter
- access packages over `/var/cache/apt/archives`

”

Ergänzungen

Packages installieren mit `apt`

comparisson to apt-get

Vergleich apt/apt-get

	apt install	apt-get install	apt upgrade	apt-get upgrade	apt full-upgrade	apt-get dist-upgrade
installierte Pakete wenn möglich auf eine neuere Version aktualisieren		ja		ja		ja
ggf. Installation neuer Pakete		ja	ja	nein		ja
ggf. Löschung unnötig gewordener Abhängigkeiten		nein		nein		ja
installiert ein lokales Paket und dessen Abhängigkeiten	ja	nein		–-		–-

Ergänzungen

Packages installieren mit `pacman`

Synchronising with the repositories

- `sudo pacman -Sy`: As new packages are added to the repositories you will need to regularly synchronise the package lists. This will only download the package lists if there has been a change (sudo apt update)
- `sudo pacman -Syy`: Occasionally you may want to force the package lists to be downloaded

Updating software

- `sudo pacman -Su`: perform an update of software already installed (sudo apt upgrade)
- `sudo pacman -Syu`: check whether the package lists are up-to-date at the same time

Ergänzungen

Packages installieren mit `pacman`

Searching for software

- `pacman -Ss ^hunspell`: searching a package by name in repos. Supports Regex
- `pacman -Qs hunspell`: searching package locally
- `pacman -Q`: list all packages installed on computer
- `pacman -Qeq`: self installed programs (e), only the program names, not the version number (q)
- `pacman -Qen`: packages self installed from main repos (n)
- `pacman -Qem`: packages self installed from aur (m)
- `pacman -Qdt`: orphans, unneeded dependencies

Find out where package installed

- `pacman -Ql handbrake`: look up where application gets installed

Ergänzungen

Packages installieren mit `pacman`

Installing software

- `sudo pacman -S gimagereader-gtk`: install package from repo
- `sudo pacman -U /var/cache/pacman/pkg/rofi-1.6.1-1-x86_64.pkg.tar.zst`: install local package

Removing software

- `sudo pacman -Rns dmenu`: remove a package (R), dependencies (s) and configuration files (n)
- `sudo pacman -Rns $(pacman -Qtdq)`: if at a later date you want to remove all orphan packages and configuration files for packages that you removed some time ago
- `sudo pacman -Sc`: remove unused packages and repos from cache

Ergänzungen

Packages installieren mit `pacman`

Finding out version number of local and remote packages

- `pacman -Qi python`: for **local** packages
- `pacman -Si python`: for **remote** packages

Misc

- If a package in the list is already installed on the system, it will be reinstalled even if it is already up to date. This behavior can be overridden with the `--needed` option.

Ergänzungen

Packages installieren mit `pacman`

Prinzip

- capital letter at beginning
- `S`: sync with repository in some way
- `Q`: search locally
- `R`: remove

Yay

- commands are the same as in `pacman`
- adds search in the **AUR (Arch User Repository)**: https://aur.archlinux.org/ (**Duckduckgo:** `!au`)
- `yay polybar` erlaubt auswahl an packages, die z.B. Discord im Namen haben

Ergänzungen

Packages installieren mit `pacman`

Anmerkungen

- **PAC**kage **MAN**ager
- always make `sudo pacman -Syu` before installing new software

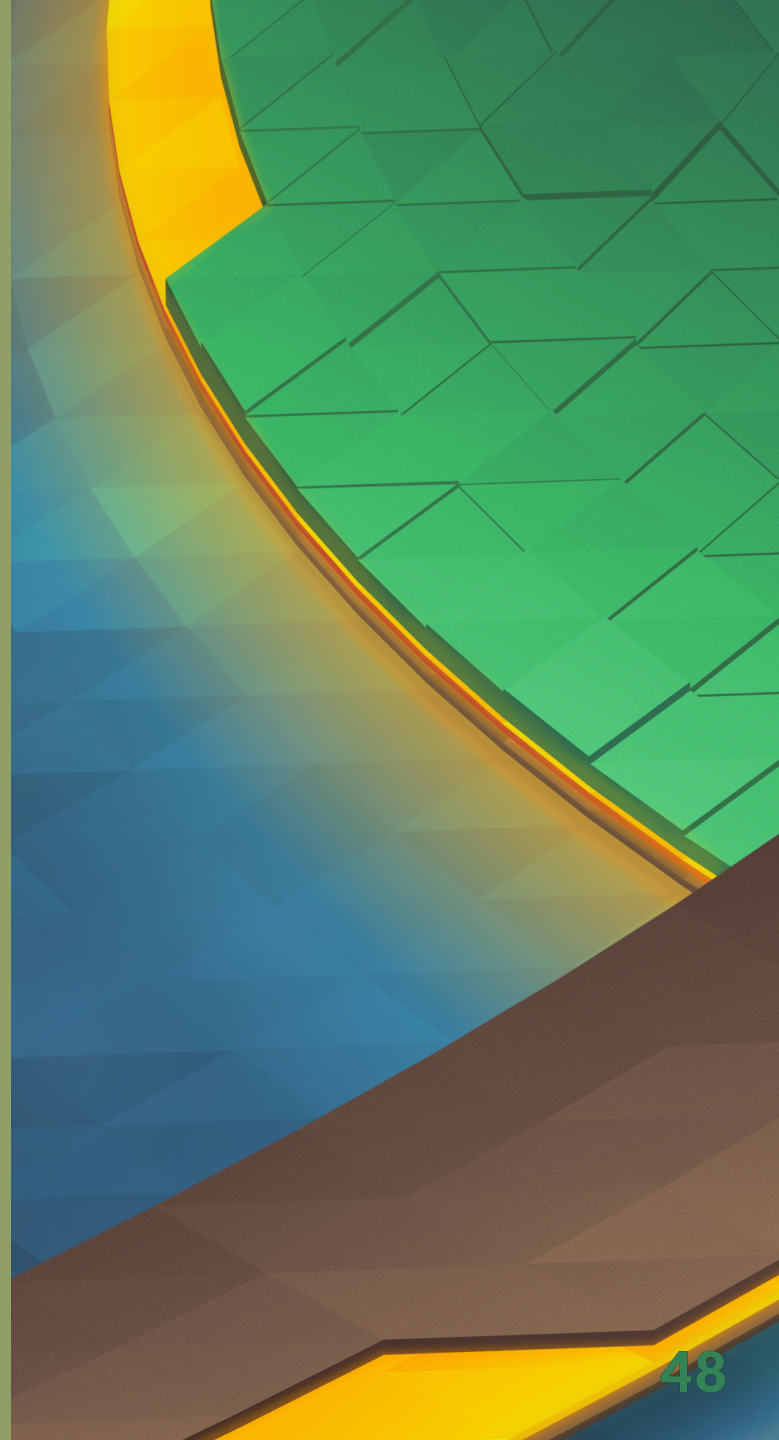
Edit configuration files

- `sudo nvim /etc/pacman.conf`

```
# Misc options
#UseSyslog
Color
#TotalDownload
# We cannot check disk space from within a chroot environment
CheckSpace
#VerbosePkgLists
ILoveCandy
```

- `sudo nvim /etc/pacman.d/mirrorlist`

Quellen



Quellen

Wissenquellen

- 👱

Quellen

Bildquellen

- 👱

Vielen Dank für eure Aufmerksamkeit!