

Besprechung der clash¹

Tafelübung Systemprogrammierung

Andreas Schärfl <andreas.schaertl@fau.de>

SS 2021

¹<https://www.schaertl.me/sp1>

Besprechung

- Jetzt Besprechung einer beispielhaften Aufgabe.
- Be excellent to each other.

Funktionen

- Den ganzen Code in eine große Schleife passen ist total überwältigend.
- Solcher Code ist kaum wartbar und macht keinen Spaß!

So geht es besser!

- Funktionen sind *das* Werkzeug zur Zerteilung von Programmen.
- Jede Funktion sollte nur *eine Stufe von Abstraktion* beinhalten.

Funktionen

- Hier leicht vereinfachtes Beispiel für Aufgabe clash.
- Funktion main ist die oberste Abstraktionsstufe. Es ist sofort klar, was in dem Programm passiert.

```
int main(void)
{
    while (true) {
        clean_up_zombies();
        print_prompt();
        char *line = get_line();
        char **argv = get_argv_from(line);
        exec(argv);
    }
}
```

Funktionen

- Funktion `exec` führt `argv` aus.
- Ruft zum Beispiel `execute_cd` oder `execute_jobs` auf.

```
static void exec(char **argv)
{
    if (!argv) {
        return;
    }

    char *cmd = argv[0];

    if (!strcmp(cmd, "cd")) {
        execute_cd(argv);
        return
    }

    // ...
}
```

Funktionen

- Funktion `execute_cd` wechselt dann tatsächlich das Verzeichnis.
- Jede Funktion hat eine und nur eine eindeutige Aufgabe.

```
static void execute_cd(char **argv)
{
    char *path = argv[1];

    if (!path) {
        fputs("cd: missing DIRECTORY\n", stderr);
        return;
    }

    if (chdir(path) == -1) {
        perror("chdir");
    }
}
```

- Viele Entwickler/Innen bevorzugen kurze Funktionen².

Weinberg points out that multiple studies have shown that programmer comprehension is essentially limited to what the programmer can see at any given instant. If the programmer has to scroll, or turn a page, their comprehension drops significantly: they have to remember, and abstract.³

²<https://softwareengineering.stackexchange.com/questions/27798/>. Sehr lesenswerte Diskussion!

³Aus: The Psychology of Computer Programming

- Länge von Funktionen hängt sehr von Programm und Programmiersprache ab.
- Trotzdem muss man sich wohl ab 30 Zeilen schon so langsam irgendwie besonders rechtfertigen.
- Funktionen so selbstverständlich wie Loops verwenden!
 - Erfahrung zeigt: Manche Studierende empfinden Funktionen als besonderen Aufwand. Das ist Unsinn. Funktionen sind so elementar wie Loops und Conditionals.

Funktionen

Oder auch: Einfaches Behandeln von Fehlern mit Funktionen.

```
static void die(const char *str)
{
    perror(str);
    exit(EXIT_FAILURE);
}

static void *xmalloc(size_t size)
{
    void *ptr = malloc(size);
    if (!ptr) {
        die("malloc");
    }
    return ptr;
}
```

Evaluation

- Lehrveranstaltungen werden an der Techfak immer evaluiert.
- Veranstaltungen werden dadurch besser. Ja wirklich!
- Freitextfelder sind am wichtigsten!

Links (klickbar!)

- Vorlesung:
<https://eva.fau.de/evasys/online.php?pswd=PYM2T>
- Tafelübung:
<https://eva.fau.de/evasys/online.php?pswd=14LQZ>
- Rechnerübung:
<https://eva.fau.de/evasys/online.php?pswd=CSX6H>

Fragen?