Besprechung fängt um 12:15 Uhr an.

Besprechung der `snail`
Tafelübung Systemprogrammierung

Andreas Schärfl <andreas.schaertl@fau.de>

WS 2020

Hallo.

- In diesem Zoommeeting wollen wir eure Abgaben genauer angucken.
- Typischerweise machen viele Studierende ähnliche Fehler und haben ähnliche Fragen.
- Beim nächsten mal besser machen und sich freuen.
- Die Folien sind eher als Erinnerung gedacht. Diese Übung soll interaktiv sein, nicht Frontalunterricht.

Typischer Fehler: Dynamischer Speicher

- Oft muss man Zeilen, Eingaben, E-Mails und so weiter bearbeiten.
- Viele haben mit `fgets(3)`, `malloc(3)` und `realloc(3)` gearbeitet.
- Das bedeutet aber Fehler checken, `free(3)` nicht vergessen
→ nervig!

Typischer Fehler: Dynamischer Speicher

Alternative 1: Mit dynamischen Arrays arbeiten.

```
char buf[LEN + 1];

if (fgets(buf, sizeof(buf), stdin)) {
    // mit buf arbeiten
}

if (ferror(stdin)) {
    perror("fgets");
    exit(EXIT_FAILURE);
}
```

Disclaimer: buf landet auf dem Stack und man kann eigentlich nicht erkennen, ob alles in Ordnung ist. Für SP aber in Ordnung.

Typischer Fehler: Dynamischer Speicher

Alternative 2: Byteweise arbeiten. Unterstützt unendlich große Ein- und Ausgaben!

```
int d;

while ((d = fgetc(stdin)) != EOF) {
    const char c = (char) d;

    // mit c arbeiten, zum Beispiel rausschreiben:

    if (fputc(c, stdout) == EOF) {
        perror("fputc");
        exit(EXIT_FAILURE);
    }

    // und soweit, ggf. hier bereits fflush(stdout)
}

if (ferror(stdin)) {
    perror("fgetc");
    exit(EXIT_FAILURE);
}
```

Typischer Fehler: `getpwuid(3)`

Richtige Fehlerbehandlung: Je nach `errno` unterschiedlich.

```
errno = 0;
const struct passwd *userInfo = getpwuid(getuid());
if (userInfo == NULL) {
    if (errno) {
        perror("getpwuid");
        exit(EXIT_FAILURE);
    } else {
        fputs("user not found\n", stderr);
        exit(EXIT_FAILURE);
    }
}
```

Typischer Fehler: getaddrinfo(3)

Richtige Fehlerbehandlung: Je nach Rückgabewert unterschiedlich.

```
const int error = getaddrinfo(addr, NULL, &hi, &res);

if (error != 0) {
    if (error == EAI_SYSTEM) {
        perror("getaddrinfo");
    } else {
        fprintf(
            stderr,
            "getaddrinfo: %s\n", gai_strerror(error)
        );
    }

    exit(EXIT_FAILURE);
}
```

- Jetzt Besprechung einer beispielhaften Aufgabe.