

Systemnahe Programmierung in C (SPiC)

23 Ergänzungen – Fehlerbehandlung

Jürgen Kleinöder, Daniel Lohmann, Volkmar Sieh

Lehrstuhl für Informatik 4
Verteilte Systeme und Betriebssysteme

Friedrich-Alexander-Universität
Erlangen-Nürnberg

Sommersemester 2020

http://www4.cs.fau.de/Lehre/SS20/V_SPiC



- Fast jeder Systemaufruf/Bibliotheksaufruf kann fehlschlagen
=> **Fehlerbehandlung unumgänglich!**
- Ziel:
Es darf kein Programm ohne Fehlermeldung abstürzen!



- Vorgehensweise:
 - Rückgabewert von Systemaufruf/Bibliotheksaufruf abfragen
 - Im Fehlerfall (häufig durch Rückgabewert -1 oder **NULL** angezeigt): Fehlercode steht in globaler Variablen **errno**
- Fehlermeldung kann mit der Funktion **perror** auf die Fehlerausgabe ausgegeben werden:

```
#include <errno.h>
void perror(const char *s);
```

- Zwischenergebnisse auf Plausibilität überprüfen

```
#include <assert.h>
void assert(int condition);
```

Wenn Bedingung **condition** nicht „wahr“ ist, wird das Programm mit Fehlermeldung abgebrochen.



- Fehlerbehandlung dem Kontext anpassen; Beispiele
 - Fehler aufgrund von Benutzer-Fehlern (z.B. Benutzer gibt falschen Dateinamen oder falsche URL ein)
 - Benutzer auf Fehler hinweisen
 - Benutzer neue Eingabe ermöglichen
 - fehlgeschlagenen Programmteil wiederholen
 - Fehler aufgrund fehlender Ressourcen (z.B. Speicher oder Platte voll)
 - Benutzer auf Fehler hinweisen
 - Benutzer Möglichkeit geben „aufzuräumen“
 - fehlgeschlagenen Programmteil wiederholen
 - Programmierfehler (z.B. Zwischenergebnisse falsch)
 - Fehlermeldung ausgeben
 - Programm abbrechen
 - ...



Fehlerbehandlung – Beispiel

```
...  
  
assert(argv[1] != NULL);  
  
/* Open file for writing. */  
FILE *fp = fopen(argv[1], "w");  
if (fp == NULL) {  
    perror(argv[1]);  
    exit(EXIT_FAILURE);  
}  
  
/* Write to file. */  
...  
  
/* Close file. */  
int ret = fclose(fp);  
if (ret == EOF) {  
    perror("fclose");  
    exit(EXIT_FAILURE);  
}  
  
...
```

```
~> ./test  
test.c:9: main: Assertion  
        'argv[1] != NULL' failed.  
~>
```

```
~> ./test /etc/shadow  
/etc/shadow: Permission denied  
~>
```

```
~> ./test hallo.txt  
fclose: Quota exceeded  
~>
```

