

---

## SPiC-Aufgabe #5: concat

(10 Punkte, keine Gruppen)

Schreiben Sie ein Programm `concat` (concatenate), das die per Kommandozeile übergebenen Parameter in einem dynamisch allokierten Puffer zu einer verketteten Zeichenfolge zusammensetzt und anschließend auf dem Standardausgabekanal ausgibt. Die einzelnen Parameter werden dabei durch ein Leerzeichen voneinander abgegrenzt.

Wenn kein Parameter übergeben wird, soll nichts ausgegeben und das Programm mit dem Statuscode `EXIT_SUCCESS` beendet werden.

### Beispiele:

```
$> ./concat Dies ist ein Test
Dies ist ein Test
$>
```

```
$> ./concat      Foo      Bar
Foo Bar
$>
```

```
$> ./concat
$>
```

### Programmablauf:

Das `concat` Programm soll in folgenden Schritten ablaufen:

1. Iteration über die Parameter und Ermittlung der Gesamtlänge mit Hilfe von `str_len`
2. Dynamische Allokation von Speicher für die zusammengesetzte Zeichenfolge mit `malloc(3)`
3. Zusammensetzen der Zeichenkette mit den Hilfsfunktionen `str_cpy` und `str_cat`
4. Einmaliger Aufruf von `printf(3)` zur Ausgabe des Gesamtstrings
5. Freigabe des reservierten Speichers mit `free(3)`

### Hinweise:

- Implementieren Sie `str_len`, `str_cpy` und `str_cat` analog zu den Bibliotheksfunktionen `strlen(3)`, `strcpy(3)` und `strcat(3)`. Ein Aufruf der Original-Bibliotheksfunktionen ist nicht zulässig. Ihre Funktionen sollen genau das in den Manpages beschriebene Verhalten haben.
- Begründen Sie die Verwendung von allen `volatile` Variablen. Wenn für mehrere Variablen die selbe Begründung gilt, dürfen Sie diese gemeinsam begründen.
- Im Verzeichnis `/proj/i4spic/pub/aufgabe5/` befindet sich die Datei `concat`, welche eine Beispielimplementierung enthält.
- Achten Sie auf aussagekräftige Fehlermeldungen, die alle auf dem Standardfehlerkanal ausgegeben werden sollen. (`fprintf(stderr, ...)(3)` / `perror(3)`)
- Testen Sie Ihr Programm auch mit `valgrind`. Dies kann bei der Suche nach Fehlern helfen. *suppressed Errors* können ignoriert werden. Weitergehende Fehlermeldungen erhalten Sie, wenn Sie `valgrind` mit den Flags `--leak-check=full --show-reachable=yes` aufrufen und das zu analysierende Binary mit Debug-Symbolen bauen.
- Ihr Programm muss mit dem folgendem Aufruf übersetzen:  
`gcc -std=c11 -pedantic -D_XOPEN_SOURCE=700 -Wall -Werror -O3 -o concat concat.c`  
Diese Konfiguration wird zur Bewertung herangezogen.
- Funktionen der `libc`, für die wir keine Fehlerbehandlung erwarten, sind online in der Linux `libc`-Doku entsprechend markiert.

---

## Abgabezeitpunkt

|     |            |          |
|-----|------------|----------|
| T01 | 09.12.2019 | 18:00:00 |
| T02 | 10.12.2019 | 18:00:00 |
| T03 | 12.12.2019 | 18:00:00 |