

Übungsaufgaben – Kapitel 1

Aufgabe 1: Einlesen von der Konsole

Schreiben Sie ein C-Programm, das ein Zeichen (`char`) von der Konsole einliest und anschließend wieder auf der Konsole ausgibt.

Hinweis: Verwenden Sie zum Einlesen die `scanf()` Funktion. Welche Eigenschaften hat diese?

Aufgabe 2: Pufferung

Übernehmen Sie das folgende Programm.

```
#include <stdio.h>

int main(void)
{
    char a,b,c;

    printf("1. Buchstabe : ");
    scanf("%c", &a);
    printf("2. Buchstabe : ");
    scanf("%c", &b);
    printf("3. Buchstabe : ");
    scanf("%c", &c);
    printf("Sie gaben ein : %c %c %c \n", a, b, c);

    return 0;
}
```

Übersetzen Sie das Programm, führen Sie es aus und geben Sie gemäß der Aufforderung Daten ein. Es sollte so sein, dass aufgrund der Pufferung von `scanf()` eine „unerwartetes“ Verhalten auftritt. Was ist passiert?

Mögliche Ausgabe.

```
pi@raspberrypi:/tmp $ gcc puffer.c -o puffer
pi@raspberrypi:/tmp $ ./puffer
1. Buchstabe : a
2. Buchstabe : 3. Buchstabe : c
Sie gaben ein : a
pi@raspberrypi:/tmp $
```

Recherchieren Sie die Funktion `fflush()` und überlegen Sie, wo Sie diese Funktion integrieren müssen, so dass das erwartete Verhalten (Ein- und Ausgabe von drei Zeichen) des Programms vorliegt.

Unter Linux wird `fflush()` das Problem vermutlich nicht lösen. Recherchieren Sie die Funktion `gets()` und integrieren Sie diese, so dass das erwartete Verhalten (Ein- und Ausgabe von drei Zeichen) des Programms vorliegt.

Hinweis: Ignorieren Sie die zunächst Warnung zu `gets()`. Implementieren Sie anschließend das Programm neu mit der Funktion `fgets()` anstatt `gets()`.

```
pi@raspberrypi:/tmp $ gcc puffer.c -o puffer
puffer.c: In function 'main':
puffer.c:10:2: warning: implicit declaration of function 'gets'; did you mean 'fgets'? [-Wimplicit-function-declaration]
    gets(buf);
    ^~~~~
    fgets
/usr/bin/ld: /tmp/ccKw0zZi.o: in function `main':
puffer.c:(.text+0x37): Warnung:the `gets' function is dangerous and should not be used.
pi@raspberrypi:/tmp $
```