

Programmieren in C

Benutzereingabe

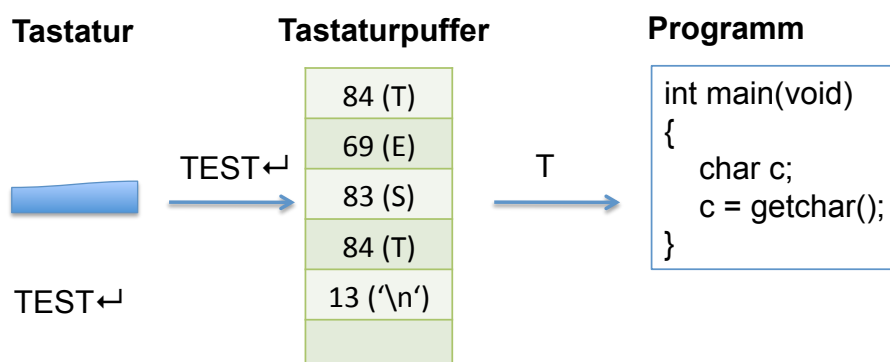
Prof. Scheffler, Beuth-Hochschule
scheffler@beuth-hochschule.de

Tastatureingabe

- Benutzer können Zeichen über die Tastatur an das Programm übergeben

Achtung:

- Die Zeichen der Tastatureingabe werden im ASCII-Code übermittelt
 - 'A' hat den Code 65,
 - 'a' hat den Code 97,
 - '0' hat den Code 48,
 - '9' hat den Code 57
- Das Programm muss also den ASCII-Wert durch Typumwandlung unterschiedlich interpretieren (Zahlen, Buchstaben, etc.).



Funktion für die Zeicheneingabe

getchar() - Funktion:

```
char ch; /* Zeichenvariable */  
ch = getchar(); /* Ein Zeichen aus dem Puffer lesen */
```

scanf() - Funktion:

```
int eingabe; /* Eingabevariable */  
int auswertung; /*tatsächlich gelesene Zeichen*/  
  
auswertung = scanf("%i",&eingabe);  
if (auswertung == 0)  
{
```

- Die scanf()-Funktion arbeitet mit Pointern (letzte Vorlesung), es muss die Adresse der Eingabevariablen übergeben werden
- Der Formatstring legt den erwarteten Datentyp fest (analog zu printf)
- Der Rückgabewert der scanf()-Funktion gibt an, wie viele Zeichen wirklich gelesen wurden (bei Fehler == 0)

Tastaturpuffer

- Die Zeichen werden nicht sofort bei der Eingabe an das Programm übergeben, sondern erst nach Bestätigung der ‚Enter‘-Taste
- Die eingegebene Zeichen + das Enter-Zeichen werden in einen Tastaturpuffer geschrieben, aus den das Programm lesen kann.
- Es gibt mehrere verschiedene Funktionen in C die den Tastaturpuffer auslesen können:
 - `getchar()` liest genau ein (1) Zeichen aus dem Puffer (der Rest bleibt im Puffer)
 - `scanf()` liest ein formatiertes Objekt (Zahl, String, Zeichen), falls kein entsprechendes Zeichen gefunden wird, wird nichts aus dem Puffer gelesen.

Problem:

- Wenn eine nachfolgende Eingabefunktion den Benutzer erneut nach einer Eingabe fragt befinden sich möglicherweise noch Zeichen im Puffer.

Lösung:

- Der Tastaturpuffer muss nach Beendigung der Eingabefunktion gelöscht werden.
- C stellt keine komfortable Löschfunktion bereit (C++ - `flush`). Es muss eine eigene Löschfunktion erstellt werden.

Funktion zum Löschen des Puffers

```
void ClearInputBuffer(void)
{
    char ch; /* helper variable stores discarded chars*/

    /* now we empty the input buffer*/
    while ((ch = getchar()) != '\n' && ch != EOF);
}
```

Aufruf der Funktion in ihrem Programm:

```
int main(void)
{
    ...
    auswertung = scanf("%i",&eingabe);
    ClearInputBuffer();
    ...
}
```