LANDESBERUFSSCHULE 4 SALZBURG

Informatik

Kommandozeilenparameter

LBS 4

Dieses Skript dient als zusätzliche Lernunterlage für Informatik

Inhalt

Kommandozeilenparameter:

Wenn Sie unter Linux und Windows ein Konsolenprogramm aufrufen, geschieht das normalerweise in der Kommandozeile. Das Terminal oder Konsole bietet eine wichtige Schnittstelle zwischen Benutzer und Programm. Beim Aufruf kann man Argumente dem auszuführenden Programm übergeben.

Beispiel:

```
C:\> mycalc.exe 5 + 4
schueler@ubuntu:~ $ ./mycalc 5 + 4
```

Der Aufruf der im Beispiel genannten Befehle übergibt drei Parameter dem Programm mycalc. Zusätzlich wird der Name des Programmes als erster Parameter übergeben. (4 Parameter)

Kommandozeilenparameter sind Argumente die dem jeweiligen Programm übergeben werden!

Aufbau der parametrisierten Main-Funktion:

Bis jetzt haben wir die Argumente in der main() leer gelassen oder mit main(void) angegeben. Die Main-Funktion kann bei Angabe der Standardparameter Argumente entgegen nehmen.
Der erste Parameter zählt die Anzahl der Argumente, der zweite Parameter ist ein Zeiger auf Zeiger und nimmt die zeichenketten entgegen.

Der zweite Parameter übergibt nur Zeichenketten, keine numerischen Werte.

Die Syntax der parametrisierten Main-Funktion wird nachstehend dargestellt.

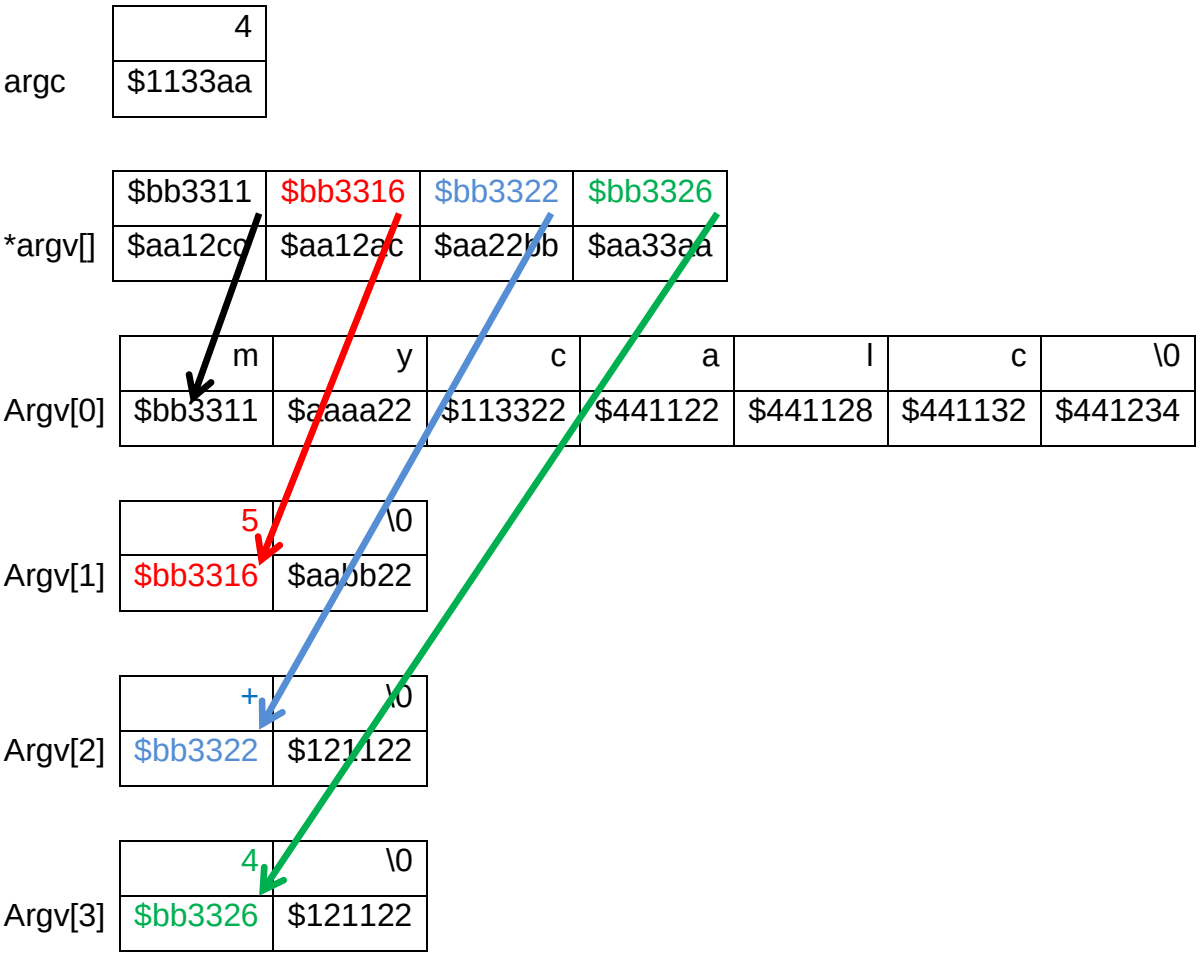
```
int main(int argc, char *argv[ ]) { /* ... */ }
```

Parameter:

Der erste Parameter int argc (**arg**ument **c**ount) ist vom Datentyp Integer und beinhaltet die Anzahl der Argumente die wir einem Programm übergeben. Der Programmname selbst wird auch mitgezählt.

Der zweite Parameter char *argv[ ] ist vom Datentyp char Zeiger auf Zeiger.

Speicherabbild des Aufrufes: schueler@ubuntu:~ $ ./mycalc 5 + 4



Nachstehend wird ein Beispiel für das Auswerten der Parameter gezeigt.

Beispiel C-Code:

```
//nimmt alle Parameter entgegen
#include <stdio.h>
#include <stdlib.h>
//argc hat den Wert  der eingegebenen Parameter + der name des Programmes
int main(int argc, char *argv[]) {
   int i;

   for(i=0; i < argc; i++) {
      printf("argv[%d] = %s ", i, argv[i]); //
      printf("\n");
   }
   return 0;
}
```

Wenn Sie keine Zeichenketten, sondern Zahlenwerte benötigen, müssen diese konvertiert werden. C stellt dafür Funktionen atoi(), atof, strtod(),….

```
int parameter = atoi(argv[1]);
//konvertiert den Parameter argv an der Indexstelle 1 in eine Ganzzahl um
```

Beispiel C-Code (Rechner):

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int main(int argc, char *argv[]){
 int zahl1,zahl2;

 if(argc < 4){
   printf("Benötige mindestens 4 Argumente!\n");
   printf("Aufruf: %s <zahl> <op> <zahl>\n", argv[0]);
   exit(1);
  }

 zahl1 = atoi(argv[1]); /*die 1. Zahl in einem Integer umwandeln*/
 zahl2 = atoi(argv[3]); /*die 2. Zahl in einem Integer umwandeln*/

 if(strcmp(argv[2],"+") ==0){ // oder       if (argv[2][0] == '+')
       printf("Das Ergebnis der Addition lautet:%d %c %d = %d\n",zahl1, argv[2][0],
zahl2, zahl1 + zahl2);
 }
 else if(strcmp(argv[2],"-") ==0){
       printf("%d\n", zahl1 - zahl2);
 }
 //.....
 return 0;
}
```

Anmerkung: Wenn sie den ***** Operator verwenden muss die Kommandozeile folg. Aufgerufen werden: calc 5 "*" 6

Grund: ***** wird von der Shell als Wildcharacter-Zeichen verwendet.