Der Mythos um „int argc und char* argv“

== Einführung ==
In sehr vielen C/C++-Einführungen wird dem Neuling erklärt, dass man (sofern überhaupt verwendet) folgende Zeile Code erstmals ignorieren soll und darauf später zu sprechen kommt: int main(int argc, char argv) {}
Meiner Meinung nach ist das eigentlich eine Codezeile, die direkt nach Variablen und Kontrollstrukturen sofort erklärt werden sollte, jedoch schnell in Vergessenheit gerät und C/C++-Neulinge erst spät lernen einfache Konsolentools zu schreiben.

== Abkürzung ausschreiben ==
Schreiben wir erstmal die Abkürzungen „argc“ und „argv“ aus, um zu erkennen, was dahinter steckt.
Argc: argument count = „Argumentenanzahl“
Argv: argument vector = „Argumentenvektor“ bzw. „Argumentenliste“

== Was steckt in den beiden Variablen? ==
Wenn wir ein Programm ausführen, kann ihm Argumente mitgegeben werden also im Prinzip Variablen, die nicht durch eine Eingabeaufforderung innerhalb des Programms eingelesen werden.
Dadurch können wir mit demselben Programm durch verschiedene Aufrufparameter, ein anderes Verhalten des Programms bestimmen. Haben wir ein Systemprogramm, das unseren Computer neustarten und runterfahren kann, könnten wir auf dem Desktop zwei Verknüpfungen zu demselben Programm anlegen, einmal als Parameter aber „runterfahren“ und bei der zweiten Verknüpfung „neustarten“. Damit ist keine Variableneingabe innerhalb des Programms notwendig.

== Beispielprogramm ==
Hier ein kleines Programm, dass du uns alle Paramter ausgibt.

Quellcode

```
1. #include <iostream>
2. // Für getchar, dass das Programm nicht direkt beendet wird und wir den Konsolenoutput sehen
3. #include <cstdio>
4. //bzw: int main(int argc, char* argv[]) {
5. int main(int argument_count, char* argument_vector[]) {
6. // Wie viele Parameter übergeben wurden
7. // Standardmäßig wird der Pfad des Programms als erster Parameter gesetzt
8. std::cout << "Insgesamt wurden " << argument_count << " Paramter \x81 \bbergeben" << std::endl << std::endl;
9. // Durch jedes Element des Vektors iterieren
10. for(int i = 0; i < argument_count; i++) {
11. std::cout << "Argument " << i << ": " << argument_vector[i] << std::endl;
12. }
13. std::cout << std::endl << "Dr\x81 \bcke eine beliebige Taste, um das Programm zu beenden.";
14. std::getchar();
15. return 0;
16. }
```

Alles anzeigen

== Quellen und weitere Infos ==

- http://stackoverflow.com/questions/3024197/what-does-int-argc-char-argv-mean
- http://publications.gbdirect.co.uk/c_book/chapter10/arguments_to_main.html