

Programmieren C: Strings, Pointer, Rekursion, ...:

Textvergleich mit Platzhalter für beliebige Zeichen

Klaus Kusche

Wir wollen eine Funktion schreiben, die genau das macht, was der Computer auch macht, wenn er prüft, ob ein Filename zu einem Filename-Muster mit * und ? passt.

Wir verwenden allerdings @ statt * und % statt ?, weil das System * und ? auf der Befehlszeile gleich selbst umwandeln würde.

Unser Hauptprogramm wird mit 2 Worten auf der Befehlszeile aufgerufen: Einem normalen Text und einem Mustertext, der @ und % (auch mehrere!) enthalten kann.

- @ steht für beliebig viele beliebige Zeichen (oder auch gar kein Zeichen).
- % steht für genau ein beliebiges Zeichen.

Das Programm soll ausgeben, ob der Text zum Mustertext passt oder nicht: Es ruft die unten beschriebene Funktion mit den beiden Befehlszeilen-Worten auf und gibt ihr Ergebnis aus.

Der Kern unseres Programmes ist eine Funktion, die mit 2 char-Pointern aufgerufen wird: Sie zeigen auf den noch zu prüfenden Rest des Textes und den Rest des Mustertextes. Als Ergebnis liefert die Funktion einen bool-Wert (passt / passt nicht).

Die Funktion muss in einer Schleife sowohl den Text als auch den Mustertext Zeichen für Zeichen durchgehen und für jede Position der Reihe nach folgende Fälle prüfen:

- **Der Mustertext ist zu Ende:**
Ist der normale Text auch zu Ende, liefert die Funktion **true**, sonst **false**.
- **Das aktuelle Zeichen im Mustertext ist ein @:**
Das ist der komplizierte Fall, er erfordert eine weitere Schleife über alle Positionen im normalen Text von der aktuellen Position bis zum Ende (einschließlich der Ende-Markierung, denn auch der leere Rest des normalen Textes muss geprüft werden!).
Der Rest des normalen Textes ab jeder dieser Schleifen-Positionen wird mit dem Rest des Mustertextes hinter dem @ verglichen, dazu rufen wir jedesmal unsere Funktion rekursiv auf.
Liefert irgendein Aufruf **true**, können wir sofort **true** zurückgeben.
Liefere alle Aufrufe **false**, ist das Ergebnis **false**.
- **Der Text ist zu Ende (aber der Mustertext nicht):**
Das Ergebnis ist **false**.
- **Das aktuelle Zeichen im Mustertext ist ein %:**
Da jedes Text-Zeichen zu % passt, wird nichts geprüft: Die Schleife macht einfach normal mit dem nächsten Zeichen von Text und Mustertext weiter.
- **Keiner dieser Fälle trifft zu:**
Dann ist das aktuelle Zeichen im Text und im Mustertext ein normaler Buchstabe. Sind die beiden verschieden, ist das Ergebnis **false**, sind sie gleich, macht die Schleife normal mit dem nächsten Zeichen von Text und Mustertext weiter.

Zusatzaufgabe: Nimm als Schleifenzähler in beiden Schleifen Pointer statt **int**'s.