

Beispiel 03_cppintro3

Dr. Günter Kolousek

21. September 2018

1 Allgemeines

- Im ersten Beispiel gibt es genaue Anweisungen zum Aufbau und der Durchführung eines Beispiels. Bei Bedarf nochmals durchlesen!
- In diesem Sinne ist jetzt ein neues Verzeichnis 03_cppintro3 anzulegen.

2 Aufgabenstellung

Wir sagen, dass ein Wort der Länge n die Periode p hat, wenn es aus mehreren gleichen Teilwörtern der Länge p besteht, die durch Verkettung das ursprüngliche Wort ergeben. Das Programm `periods` soll die kürzeste Periode finden und davon diese Periode und das Teilwort ausgeben.

Vorerst sind *keine* Fehler abzufangen oder zu behandeln. Beispiele sind:

```
$ periods abcabc
3 abc
$ periods aaa
1 a
$ periods abcabcd
7 abcabcd
$
```

3 Anleitung

1. Von diesem Beispiel an werden wir das Build-Automation-Tool meson (<http://mesonbuild.com>) verwenden, das aus einer Konfiguration (im Sinne des Kompilierens, Binden,...) in einer Datei `meson.build`, Standard Build-Dateien für ein natives Build-Tool (wie z.B. für ninja oder Windows-Visual Studio) erstellt.

Installiere unter Manjaro das Paket `meson`.

Entpacke **im** Verzeichnis `<lastname>_<studentnumber>` das Archiv `template.tar.gz` und nenne das Verzeichnis `template` in `03_cppintro3` um. Weiters ist die Datei `meson.build` entsprechend anzupassen.

Ich habe ein eigenes Tutorial für meson erstellt... Zu finden an gewohnter Stelle!

2. Schreibe das Programm als Datei `periods.cpp` im Verzeichnis `src` und erstelle im Verzeichnis `build` das Programm. Gehe dazu in das Verzeichnis `build`. Einmalig (oder bei Änderungen des Projektes) ist dort das Kommando `meson ..` einzugeben. Die beiden Punkte geben das Verzeichnis an wo sich die Datei `meson.build` befindet. Jetzt wird die vorhandene Software überprüft, die Abhängigkeiten der Source-Dateien festgestellt und die notwendigen Dateien zum Übersetzen generiert.

Danach wird mittels dem Befehl `ninja` das Projekt übersetzt.

Zusammenfassend:

```
$ meson ..  
... Ausgaben von meson ...  
$ ninja  
... Ausgaben von ninja ...
```

Hinweise:

- Nur ein Teiler der Länge des Wortes kommt als Periode in Frage.
 - $\text{word}[k\%i] = \text{word}[k]$ für alle $k = i, i + 1, \dots, n - 1$, wobei i von 1 bis $n - 1$ läuft (weitere Optimierung möglich).
 - Die Klasse `string` hat viele nützliche Methoden:
 - `size_t size()` liefert die Länge des Strings zurück. `size_t` ist ein Typ, der die Größe eines Objektes aufnehmen kann und groß genug ist um die Größe des größten Objektes aufzunehmen.
 - `string substr(size_t pos=0, size_t len=npos) const` liefert den Substring von der Position `pos` mit einer Länge `len` zurück. `string::npos` ist der größte Wert von `size_t`. Das bedeutet in diesen konkreten Fall, dass der Substring bis zum Ende des Strings geht. Du siehst in C++ gibt es Default-Parameter!
 - Beachte, dass der Operator `[]` immer mit einem gültigen Index aufgerufen werden muss, da die Größe nicht überprüft wird.
3. Erweitere jetzt das Programm, um die Verarbeitung einer beliebigen Anzahl von Wörtern. D.h. auf der Kommandozeile können eine beliebige Anzahl an Wörtern mitgegeben werden und die Ausgabe erfolgt untereinander, je in einer Zeile.

4. Schaue dir jetzt die Dokumentation der Klasse `std::string` genau an und schreibe selber ein kleines Testprogramm, das die verschiedenen Möglichkeiten dieser Klasse zeigt. Dieses Testprogramm soll in einem Unterverzeichnis `string_testing` von `03_cppintro3` sein, ebenfalls ein meson-Projekt sein und die Möglichkeiten von `string` demonstrieren (Ausgabe auf Konsole).

4 Übungszweck dieses Beispiels

- Verwenden und kennenlernen von meson
- Kennenlernen der Klasse `std::string`