

Hinweise zu Makefiles:

Die Syntax von Makefiles enthält im Wesentlichen zwei Elemente, die Angabe von Targets und Variablen. Ein Target ist ein Ziel, welches durch die Ausführung von Kommandos erreicht wird. Ein Target kann Abhängigkeiten (Dependencies) von anderen Targets oder von Dateien haben. Die Schritte sind üblicherweise Kommandos, die auf der Kommandozeile einer Shell ausgeführt werden.

Syntax Targets:

```
target: dependency1 dependency2
    Kommando1 <Parameter> ...
    Kommando2 <Parameter> ...
```

Nicht sichtbar aber wichtig ist, dass **vor einem Kommando ein Tabulatorzeichen** steht und keine Folge von Leerzeichen. Ein Target und eine Dependency kann z.B. eine Datei sein. Deren Existenz bzw. deren Alter bestimmt, ob das Kommando ausgeführt werden muss. Wenn in den Abhängigkeiten eine Datei jüngerem Datum ist als im Target, heißt dies, dass das Kommando erneut ausgeführt werden muss. Eine Abhängigkeit kann wiederum in einer weiteren Angabe als Target angegeben werden. Die Abhängigkeiten werden von make in beliebiger Tiefe aufgelöst.

Um Angaben zu parametrisieren, lassen sich Variablen definieren. Diese können auch umfangreiche Angaben umfassen wie z.B. die Liste ihrer Dateien. Angaben über mehrere Zeilen lassen sich durch das Zeichen \ am Zeilenende zusammenfassen.

Syntax Variablen:

```
FILES = File1.cpp File2.cpp \
      File3.cpp
OPTIONS = --coverage -g
```

Aufgelöst werden Variablen dann z.B. bei dem Aufruf eines Kommandos:

```
// Kompiliere und linke eine ausführbare Datei aus zwei Quelldateien
// neu, wenn sich mindestens eine der Dateien geändert hat
executable: $(FILES)
    gcc $(OPTIONS) $(FILES)
```

Dies führt auf der Kommandozeile zu folgendem Aufruf

```
gcc -coverage -g File1.cpp File2.cpp File3.cpp
```

Weitere Hinweise finden Sie z.B. hier

<http://www.oreilly.de/german/freebooks/rlinux3ger/ch133.html>