

Kurzeinführung in `make` - Ein einfaches Kompilierwerkzeug

Tobias Sielhorst

24 October 2006

chair for computer aided medical procedures

department of computer science | technische universität münchen

make – Das Wichtigste in Kürze

- Skriptähnliches Programm zur Übersetzung und Erstellung von Programmen
- Standard: Auf allen POSIX konformen Betriebssystemen zu Hause (Wie Unix, Linux, Windows NT, QNX, etc. pp.)
- `make` benutzt die Datei `makefile` oder `Makefile` im gleichen Verzeichnis und wird in der Kommandozeile aufgerufen

Mehrere Dateien kompilieren

`all:` muss im makefile
enthalten sein!

Am Anfang eines
Kommandos muss
ein tab stehen!

```
all:
    g++ -c -o test.o test.cpp
    g++ -c -o main.o main.cpp
    g++ -o main.exe main.o test.o
```

- Dieses `makefile` kompiliert die Dateien `test.cpp` und `main.cpp` zu `main.exe`
- Es erstellt auch die Objektdateien `test.o` und `main.o`
- Ausführen der Datei mit `make` oder `make all`

Abhängigkeiten

```
all: test.cpp test.h
    g++ -c -o test.o test.cpp
    g++ -c -o main.o main.cpp
    g++ -o main.exe main.o test.o
```

- Dieses `makefile` kompiliert die Dateien `test.cpp` und `main.cpp` zu `main.exe`, nur wenn `test.cpp` und `test.h` sich geändert haben

Platzhalter

```
# a hash character marks a comment line
all: *.cpp *.h
    g++ -c -o test.o test.cpp
    g++ -c -o main.o main.cpp
    g++ -o main.exe main.o test.o
```

- Dieses `makefile` kompiliert die Dateien `test.cpp` und `main.cpp` zu `main.exe`, wenn eine der Header oder CPP-Dateien im Verzeichnis sich geändert haben

Mehrere Kompilierziele

```
all: test.o main.o

    g++ -o main.exe main.o test.o
test.o: test.cpp test.h

    g++ -c -o test.o test.cpp
main.o: main.cpp main.h

    g++ -c -o main.o main.cpp
clean:

    rm *.o
```

- Dieses `makefile` kompiliert das Programm
- Wenn eine der Objektdateien schon vorhanden ist und aktuell ist, wird sie nicht neu kompiliert
- Mit `make clean` kann man die überflüssigen Dateien wieder löschen

Makros

```
COMPILER = g++  
all: test.o main.o  
    $(COMPILER) -o main.exe main.o test.o  
test.o: test.cpp test.h  
    $(COMPILER) -c -o test.o test.cpp  
main.o: main.cpp main.h  
    $(COMPILER) -c -o main.o main.cpp
```

- Makros werden vor dem Lauf von `make` in der Datei ersetzt
- Sie dienen der Übersichtlichkeit und der besseren Konfigurierbarkeit (Oben könnte mit wenig Aufwand der GNU Compiler `g++` zu dem Microsoftcompiler `mcc` geändert werden)

Mehr..?

Die Referenzseite auf englisch:

- `man make` in der Kommandozeile eintippen

Auf deutsch im Netz:

- www.ijon.de/comp/tutorials/makefile.html
- www.informatik.hu-berlin.de/~tegos/tutorials/makehelp.html
- www.jfranken.de/homepages/johannes/vortraege/make_inhalt.de.html