Paralleles Programmieren für Geowissenschaftler

2. Makefiles

Ulrich Körner

Wissenschaftliches Rechnen
Fachbereich Informatik
Universität Hamburg

10-4-2014

informatik
die zukunft

make ist ein Tool, mit dem komplexe Programme auf einfache Weise kompiliert werden können.

Dabei können Abhängigkeiten von Dateien berücksichtigt werden, die eine bestimmte Reihenfolge beim Kompilieren erfordern.

Das geschieht, indem das Programm "make" die Zeitstempel der Dateien ausgewertet.

Beispiel: Datei B hängt von Datei A ab.

Ist Datei A verändert worden und wird neu kompiliert, so muss auch Datei B neu kompiliert werden, damit das Programm fehlerfrei ablaufen kann, da möglicherweise die Schnittstelle eines subroutine calls geändert wurde.

Diese Abhängigkeiten werden in einer Datei festgehalten, die üblicherweise den Namen 'makefile' oder 'Makefile' trägt.

Das grundlegende Element eines Makefiles ist die "Regel":

Target ... : Voraussetzungen ...
    tab! Kommando
    ...

Ein häufiger Fehler an dieser Stelle ist, statt eines Tabulatorzeichens entsprechend viele Leerzeichen zu setzen. Das sieht optisch genauso aus (einige Editoren lassen sich auch so einstellen, dass tabs in Leerzeichen umgewandelt werden!), führt aber dazu, dass das Makefile nicht das macht, was erwartet wird und z.B. folgenden Fehler meldet:

makefile:13: *** missing separator. Schluss.

Für das Kompilieren des Programms hello.f90 kann folgendes Target aufgesetzt werden.

```
hello.x: hello.f90
        f95 -o hello.x hello.f90
```

Um den Befehl zum Kompilieren anzustoßen, gibt man auf der Kommandozeile ein:

```
$ make hello.x
```

Wenn "hello.x" das erste Target des Makefiles ist, reicht einfach:

```
$ make
```

da dann automatisch das erste Target abgearbeitet wird.

Gibt es eine allgemeine Regel, z.B. für das Kompilieren von Quellcode, also die Erstellung einer Objektdatei "sub.o" aus "sub.f90", kann die Abhängigkeit von zwei Programmteilen auch nur mit den Teilen "Target" und "Voraussetzung" ausgedrückt werden.

sub.o: mo-parameter.o

Um sub.o zu erstellen, wird zuerst geprüft, ob mo-parameter.o existiert und älter ist, als sub.f90. Falls nicht, wird aus mo-parameter.f90 zuerst mo-parameter.o erstellt, bevor sub.f90 kompiliert wird.
Auf diese Weise können komplexe Programmstrukturen erstellt werden, bei denen nur jeweils die Teile neu kompiliert werden, die notwendig sind, um ein akutelles lauffähiges Programm zu erhalten.

Weitere Informationen:
`http://www.gnu.org/software/make/manual/make.html`

oder auf dem cluster
$ man make