

Kurzanleitung zur Benutzung von FORTRAN bzw. C unter LINUX

Um aus einem in einer höheren Programmiersprache geschriebenen Programm einen auf einem Computer lauffähigen Code zu erzeugen, sind zwei Schritte nötig:

Kompilieren

Im ersten Schritt wird das Programm von einem sog. Compiler in eine von der Maschine abhängige niedrigere Programmiersprache, der sog. Assembler-Sprache übersetzt.

Binden

Im zweiten Schritt werden alle benötigten, in Assembler-Sprache vorliegenden Teile (insbesondere die evtl. benötigten, in einer Bibliothek vorhandenen Standardfunktionen) zu einem binären, direkt auf dem Prozessor ausführbaren Code umgewandelt und zusammengefaßt.

Unter LINUX werden FORTRAN-Programme unter der Namenskonvention *name.f* erwartet. Der Befehl

```
g77 -c [optionen] name.f
```

führt für *name.f* den ersten Schritt durch und speichert das übersetzte Programm in *name.o* ab. Bei Verwendung der Option `-O` wird zusätzlich das übersetzte Programm optimiert, um einen effizienteren Code zu erzeugen. Der Befehl

```
g77 -o prog name1.o ... namek.o -lm
```

führt dann den zweiten Schritt durch, indem es die Teile *name₁.o* bis *name_k.o* und evtl. benötigte Bibliotheksroutinen zu einem lauffähigen Programm zusammenbindet und unter dem Namen *prog* ablegt. Dieses kann dann mit `./prog` oder (wenn der Suchpfad entsprechend gesetzt ist) mit *prog* ausgeführt werden.

Entsprechend werden unter *name.c* C-Programme erwartet. Die jeweiligen Befehle sind

```
gcc -c [optionen] name.c
```

für den ersten Schritt und

```
gcc -o prog name1.o ... namek.o -lm
```

für den zweiten Schritt.

Unter LINUX kann die von einem Befehl *befehl* benötigte Rechenzeit mit

```
time befehl
```

ermittelt werden. Man beachte, daß die Genauigkeit dieser Zeitmessung bei etwa 0.01 s liegt. Benötigt *befehl* weniger Zeit, so muß man durch Verwendung einer Schleife innerhalb von *befehl* dafür sorgen, daß man in einen sinnvollen Meßbereich gelangt.