

Bauen einer Bibliothek

Beispiel: Zufallsahlen

Schritt 0: Die Quelldateien

Wir haben die Dateien

- `random.h` (Header)
- `random.c` (Implementierung)
- `random_pi.c` (Anwendung)

im Arbeitsverzeichnis.

Schritt 1: Verzeichnisse erzeugen

Erstelle Unterverzeichnisse

- `librandom`
- `librandom/include`
- `librandom/include/IFM`
- `librandom/lib`

z.B. im Heimatverzeichnis `/home/yourhome`

Schritt 2: Kopieren der Dateien

- Kopiere `random.h` nach `librandom/include/IFM`
- Kopiere `random.c` nach `librandom/lib`

Schritt 2: Kopieren der Dateien

- Kopiere `random.h` nach `librandom/include/ifm`
- Kopiere `random.c` nach `librandom/lib`

Normalerweise hat man für Letzteres noch ein Extraverzeichnis `librandom/src`, aber der Einfachheit halber sparen wir uns das hier.

Schritt 3: Bauen der Bibliothek

Im Verzeichnis `librandom/lib`:

- `g++ -I../include -c random.C`
(erzeugt den Objekt-Code `random.o`)
- `ar r librandom.a *.o`
(erzeugt die Bibliothek `librandom.a`)

Schritt 4: Kompilieren der Anwendung

Im Arbeitsverzeichnis:

- `g++ -I/home/yourhome/librandom/include
-L/home/yourhome/librandom/lib
random_pi.C -lrandom -o random_pi`
(erstellt das ausführbare Programm, das dann mit `./random_pi` gestartet wird)

Schritt 4: Verbesserung durch Makefile

- Eintippen des `g++`- Kommandos ist mühsam
- Nach Abspeichern der folgenden Datei **Makefile** im Arbeitsverzeichnis können wir einfach `make random_pi` eintippen
- F9 im `emacs` geht dann auch...



Makefile

```
# Makefile for using our small random numbers library
# =====
# to compile my_prog.C, type make my_prog

# directory where you installed the library
RANDOM_DIR = ${HOME}/librandom

# include path
IPATH = -I${RANDOM_DIR}/include

# lib path
LDPATH = -L${RANDOM_DIR}/lib

# libs
LIBS = -lrandom

# make targets
%: %.C
    g++ ${IPATH} ${LDPATH} *.C ${LIBS} -o $*
```