

# PROGC Lab08 Makefile und Unit Test

## Inhalt

PROGC Lab08 Makefile und Unit Test .....	1
1 Einführung.....	1
2 Lernziele .....	1
3 Hintergrund Informationen .....	1
3.1 Tests .....	1
3.2 Hilfe zu Makefile .....	1
3.3 Hilfe zu assert.h .....	1
3.4 Hilfe zu CUnit .....	2
4 Aufgaben.....	2
4.1 Makefile lesen und verstehen.....	2
4.2 Assert.h Anwendung verstehen.....	2
4.3 CUnit Assertions .....	2

## 1 Einführung

Das Make Tool und das CUnit Test Framework werden in den Praktika verwendet um die Programme zu bilden und um die Tests zu definieren und auszuführen.

In diesem Praktikum lernen Sie die Konzepte des Make Tools und schreiben ein einfaches Makefile.

Sie lernen den Unterschied zwischen assert.h Assertions und CUnit Assertions und wann Sie welche der beiden anwenden.

## 2 Lernziele

- Sie können einfache Makefiles lesen und schreiben.
- Sie kennen den Unterschied zwischen assert.h Assertions und CUnit Assertions.
- Sie können assert.h Assertions korrekt anwenden.
- Sie können CUnit Tests mittels CUnit Assertions definieren und ausführen.

## 3 Hintergrund Informationen

### 3.1 Tests

Dieses Praktikum enthält keine automatisierten Abnahmetests.

### 3.2 Hilfe zu Makefile

Siehe <https://www.gnu.org/software/make/manual/>.

### 3.3 Hilfe zu assert.h

Siehe `man 3 assert`.

Siehe in den Vorlesungsunterlagen wie der generierte Core-Dump analysiert werden kann.

### 3.4 Hilfe zu CUnit

Siehe <http://cunit.sourceforge.net/documentation.html>.

Speziell <http://cunit.sourceforge.net/doc/index.html>

Doxygen: lokale Dokumentation unter `CUnit/doc/CUnit/index.html` oder <http://cunit.sourceforge.net/doxdocs/index.html>

Liste aller Assertions: [http://cunit.sourceforge.net/doc/writing\\_tests.html#assertions](http://cunit.sourceforge.net/doc/writing_tests.html#assertions)

## 4 Aufgaben

### 4.1 Makefile lesen und verstehen

- Lesen Sie das Makefile `lab07-calculator/Makefile` und versuchen Sie jede Zeile zu verstehen.
- Schlagen Sie gegebenenfalls im GNU Make Manual nach.
- Führen Sie `make clean` aus. Danach `make -n` und kommentieren Sie den Output.

### 4.2 Assert.h Anwendung verstehen

- Fügen Sie im `lab07-calculator/tests/tests.c` File in irgend einem der Tests eine `assert(0);` ein (vergessen Sie `#include <assert.h>` nicht).
- Lassen Sie die Tests laufen (`make test`) und analysieren Sie den Core-Dump (siehe Vorlesung).

### 4.3 CUnit Assertions

- Lesen Sie das `lab07-calculator/tests/tests.c` File und versuchen Sie jede Zeile zu verstehen.
- Fügen Sie einen zusätzlichen Test hinzu der eine komplexere Expression prüft.
- Führen Sie die Tests aus mit `make test`.