

# 1. Übungsblatt zur Vorlesung „Einführung in die Programmiersprache C++“

Wintersemester 2018 / 2019

## Ziele des Aufgabenblattes

In dieser Übung machen Sie sich mit den Grundlagen der Ein- und Ausgabe von Daten in C-Programmen vertraut. Sie lernen die wichtigsten Compiler-Argumente des GNU C-Compilers kennen und informieren sich, wie in Linux Header und Bibliotheken organisiert sind. Im letzten Teil machen Sie sich mit `git` vertraut und lernen, wie Sie Ihre Aufgaben auf dem bereitgestellten Gitlab-Server abgeben können.

## Vorraussetzungen

Es wird erwartet, dass Sie sich Zugang zu einer Linux-Umgebung verschaffen und dass Sie grundlegende Dateioperationen beherrschen. Wir empfehlen Ihnen Ubuntu 16.04 zu nutzen. Eine Liste mit den für die Veranstaltung benötigten Paketen wird Ihnen in StudIP zur Verfügung gestellt.

Sie dürfen eine Entwicklungsumgebung Ihrer Wahl benutzen. Wie empfehlen Ihnen, Sich mit Eclipse für C++ vertraut zu machen. Das CDT-Plugin zum Erstellen von C/C++-Projekten ist auf den Rechnern im CIP-Pool installiert. Sollten Sie lieber einen Texteditor benutzen wollen, empfehlen wir Ihnen sich mit `emacs` oder `vim` vertraut zu machen.

## Aufgabe 1.1: Ein- und Ausgabe in C (60 Punkte)

- Schreiben Sie ein Programm `first_diff`, das zwei Textdateien zeilenweise vergleicht und die erste Zeile ausgibt, an denen Sie sich unterscheiden. Die Textdateien sollen per Kommandozeilenparameter übergeben werden. Testen Sie auf korrekte Anzahl der Parameter. (20 Punkte)
- Schreiben Sie ein Programm `break_words`, das die Eingabe des Benutzers wortweise auf der Konsole ausgibt. Nutzen Sie für die Ein- und Ausgabe ausschließlich die Funktionen `getchar()` und `putchar()` (20 Punkte)
- Schreiben sie ein Programm `line_mean`. In StudIP finden Sie die beiden Dateien `numbers.txt` und `numbers_broken.txt`. Diese bestehen aus Zeilen mit vier getrennten Ganzzahlen pro Zeile. Geben Sie den Inhalt jeder Zeile zusammen mit dem Mittelwert der Zahlen aus. Testen Sie zunächst mit `numbers.txt`. Die Datei `numbers_broken.txt` besitzt eine Zeile, die nicht den Anforderungen entspricht. Schreiben Sie einen Test, ob das Einlesen einer Zeile erfolgreich war. Fangen Sie die fehlerhafte Zeile ab und testen Sie Ihre Implementierung mit der Datei `numbers_broken.txt`. (20 Punkte)

## Aufgabe 1.2: Einbindung externer Bibliotheken (20 Punkte)

In dieser Aufgabe machen Sie sich mit den grundlegenden Voraussetzungen zum Einbinden externer Bibliotheken vertraut. Beantworten Sie dazu die folgenden Fragen:

- Wo liegen unter Linux standardmäßig die vorhandenen Header- und Bibliotheksdateien?
- Welche Unix-Umgebungsvariablen legen die Pfade fest, in denen gcc nach verfügbaren Header-Dateien und Bibliotheken sucht?
- Erklären Sie die Bedeutung der gcc-Parameter `-I`, `-L`, `-l`
- Was ist der Unterschied zwischen `#include "header.h"` und `#include <header.h>`?
- Erklären Sie den Unterschied zwischen statischem und dynamischen Linken. Wie können Sie statisches Linken erzwingen? Welche Konsequenzen hat das?
- Welche Informationen liefert Ihnen das Linux-Tool `ldd`?

## Aufgabe 1.3: Testprogramm mit gsl (20 Punkte)

Nutzen Sie die GNU Scientific Library (GSL), um die Werte einer Normalverteilung mit einer Varianz von 1.0 Sigma zu plotten. Nutzen Sie dazu die Funktion `gsl_ran_gaussian(...)`, die im Header `<gsl/gsl_randist.h>` deklariert ist. Die Dokumentation finden Sie unter [https://www.gnu.org/software/gsl/manual/html\\_node/The-Gaussian-Distribution.html](https://www.gnu.org/software/gsl/manual/html_node/The-Gaussian-Distribution.html). Die Header und Bibliotheken der GSL finden sie unter Ubuntu im Paket `libgsl-dev`. Um ein ausführbares Programm zu erhalten, müssen sie die Bibliotheken `libgsl` und `libblas` dazu linkern.

Geben Sie die Werte der Normalverteilung im Wertebereich -5.0 bis +5.0 in Schritten von 0.1 wie im folgenden Beispiel veranschaulicht auf dem Terminal aus:

```
-0.400000 0.368270
-0.200000 0.391043
-0.000000 0.398942
0.200000 0.391043
0.400000 0.368270
```

Speichern Sie diese Werte durch Umlenkung der Ausgabe in eine Datei `gaussian.txt` und plotten Sie diese mit `gnuplot`.

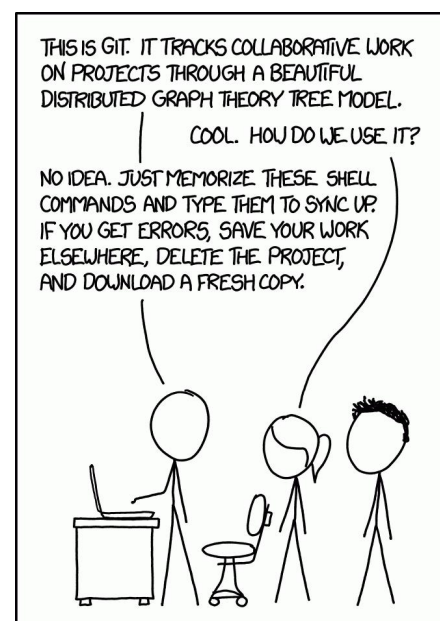
## Abgabe

Die Abgabe erfolgt über das GitLab-Repository ihrer Gruppe. Damit wir für Sie eins einrichten können, benötigen Sie ein Benutzerkonto auf unserem GitLab-Server. Dazu melden Sie sich mit Ihrem RZ-Login unter <https://gitlab.informatik.uni-osnabrueck.de> an. Bei ersten Login wird Ihr Benutzer geblockt, er muss zunächst von uns freigeschaltet werden. Bitte melden Sie sich frühzeitig an, damit wir Sie zeitnah unblocken und den Repositories zuordnen können. Nachdem Ihnen Zugang gewährt wurde, können Sie Ihr Repository mit folgendem Befehl klonen:

```
git clone
git@gitlab.informatik.uos.de:cpp2018/<GRUPPENID>.git
```

Dazu müssen Sie in Ihrem Benutzerprofil einen SSH-Schlüssel hinterlegen. Entweder verwenden Sie einen bereits vorhandenen oder erzeugen sich einen neuen mittels:

```
ssh-keygen
```



und tragen den Inhalt der Datei `.ssh/id_rsa.pub` unter dem Punkt „SSH Keys“ in Ihrem GitLab-Benutzerprofil ein. Wenn Sie Ihr Repository erfolgreich ausgecheckt haben, finden Sie dort einen Ordner `uebung01` mit einer Datei `welcome.txt`. Fügen Sie Ihrem Repository in diesem Verzeichnis die Dateien für Ihre Abgaben zu den Aufgaben 1.1 bis 1.3 hinzu. Dazu verwenden Sie die Befehle `git add` und `git push`. Machen Sie sich mit diesen Befehlen vertraut. Mittels `git push` werden die Daten von Ihren lokalen Repo auf den GitLab-Server übertragen. Machen Sie sich mit den grundlegenden `git`-Befehlen vertraut und erklären Sie diese. Legen Sie dazu eine Datei `gitusage.txt` mit den entsprechenden Erläuterungen in Ihren Repository an. Stellen Sie sicher, dass Ihre Abgabe bis Montag, 29.10.2018 8:00 Uhr auf den Server übertragen wurden. Einstiegstutorials zur Nutzung von `git` finden Sie z.B. unter <https://training.github.com>. Bringen Sie pro Gruppe einen Ausdruck der von Ihnen bearbeiteten Dateien - mit Gruppen-ID und Namen versehen - mit.