Einleitung

Für die Bearbeitung der Übungsaufgaben der Lehrveranstaltung benötigst du Zugang zu einem Rechner mit einem C++-Compiler und einer entsprechenden Entwicklungsumgebung. Der Compiler (zusammen mit dem Linker) übersetzt den in Textform vorliegenden Programmcode in eine ausführbare Datei. Die Entwicklungsumgebung ist im Prinzip nur ein Texteditor mit einigen Zusatzfunktionen, die den Programmierer beim Erstellen des Programmcodes unterstützt und auf Knopfdruck den Compiler startet. Wir haben in den Poolräumen und für die Installation auf Windows-Laptops den *Clang Compiler-Frontend* und die *Geany Entwicklungsumgebung* bereitgestellt.

In diesem Dokument findest du eine Anleitung, wie du auf einem Poolrechner oder auf deinem Laptop das erste Programm kompilieren und ausführen kannst.

> **Zusatzinformation** *Warum setzen wir in der Lehrveranstaltung Geany und Clang ein?* Clang ist Open-Source (also frei zugänglich) und läuft auf fast allen Plattformen (Linux, Windows,...) und kann Binaries für eine noch größere Anzahl von Plattformen erzeugen – inklusive zahlreicher eingebetteter Systeme. Dieser Compiler lässt sich kostenfrei sowohl unter Linux als auch unter Windows einsetzen. Geany ist ebenfalls Open-Source und generell in der Softwareentwicklung weit verbreitet. Im Prinzip funktionieren für die Aufgaben der Lehrveranstaltung auch beliebige andere C++-Compiler und Entwicklungsumgebungen – dafür können wir jedoch keine Unterstützung anbieten.
>
> *Was hat es mit Clang, clang++ und LLVM auf sich?* Bei diesen Begriffen kann es für Einsteiger leicht zu Verwirrgung kommen, da sie eng zusammenhängen und mitunter als Synonyme verwendet werden. LLVM ist ein Projekt, das Compiler für verschiedene Programmiersprachen und Zielsysteme (PC, Smatrphone, etc.) bündelt. Clang ist der C/C++ Compiler aus der LLVM Compilersammlung. Mit dem Programm `clang++` wird C++ Code in (ausführbare) Binärdateien übersetzt. Daher *installieren* wir LLVM, sprechen vom Clang *Compiler* und benuzten das *Programm* `clang++` zum komlilieren.

0.1 Zugang zu den Lehrmaterialien über koaLA

Die Materialien zu der Lehrveranstaltung werden online über das koaLA-System zur Verfügung gestellt. Das System erreichst du unter `https://koala.uni-paderborn.de`. Damit du in koaLA zu der Lehrveranstaltung registriert werden kannst, melde dich auf der koaLA-Seite mit deinem IMT-Login an. Nach der ersten Anmeldung wird die automatische Synchronisierung mit PAUL gestartet, so dass du **ab dem nächsten Tag** in koaLA auf die Daten zugreifen kannst.

0.2 Für Laptopübungen: Eigenen Laptop vorbereiten

Um für eine Übung den eigenen Laptop nutzen zu können, musst du einen C++-Compiler und eine Entwicklungsumgebung installiert haben. Im Rahmen der Übungen verwende bitte die zur Verfügung gestellte Version von Clang und Geany für Windows-Rechner (ab Windows XP) – andere Compiler und Umgebungen können auch funktionieren, wir können dafür aber keinerlei Unterstützung anbieten. Bitte beachte, dass wir hier von einem 64bit-Betriebssystem (Windows) ausgehen. Dies sollte schon seit längerem der Standard sein.

Installation von Clang

1. Clang herunterladen (Version 3.7.1): *Clang for Windows (64-bit)*:
   `http://llvm.org/releases/3.7.1/LLVM-3.7.1-win64.exe`

2. Starte die Installation von Clang. Wähle im Dialog die Option *Add LLVM to the system PATH for all users* (siehe Abbildung 1):

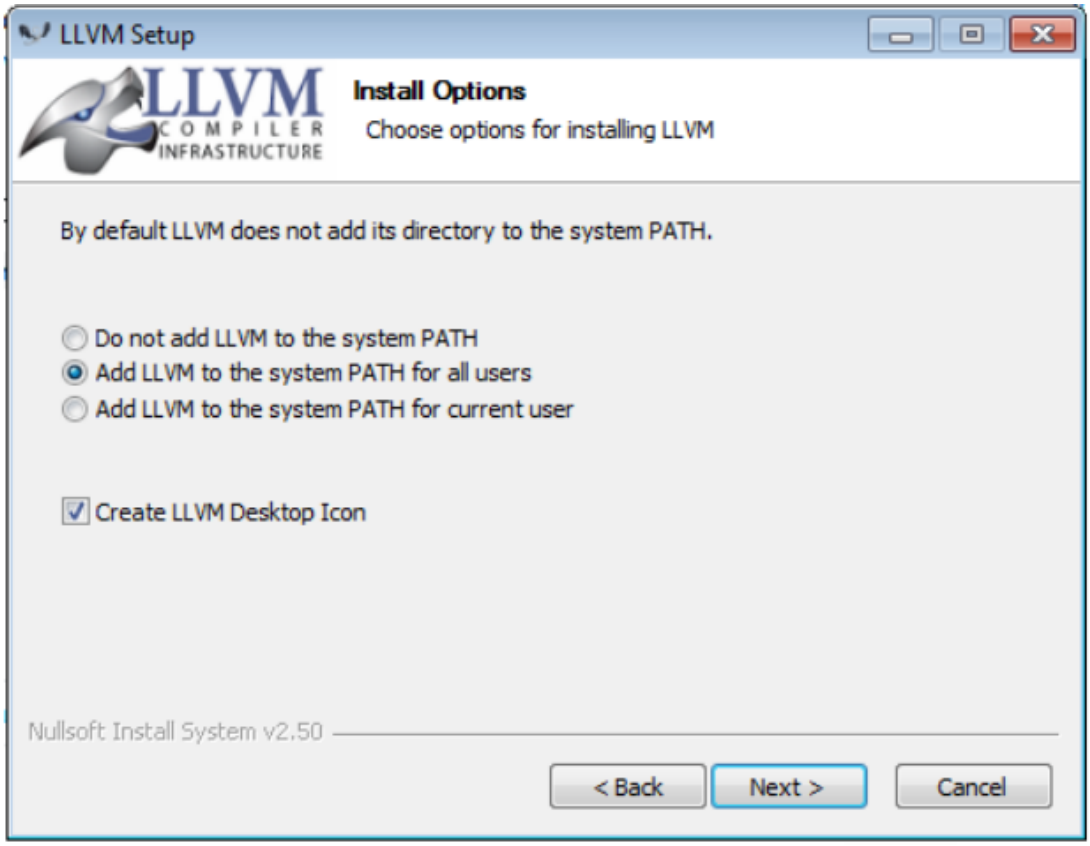

Abbildung 1: LLVM Setup

3. Belasse den Installationspfad unbedingt beim Standard: `C:\Program Files\LLVM`

4. Während der Installation kann es zu der in Abbildung 2 kommen. Diese kann einfach ignoriert und mit einer beliebigen Taste weggedrückt werden.

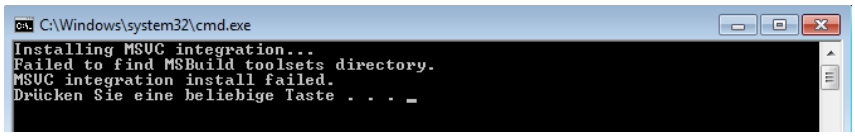

Abbildung 2: Test if clang is in the system PATH

5. Nach der Installation solltest du `clang++.exe` im Verzeichnis `C:\Program Files\LLVM\bin` finden. Das bin-Verzeichnis sollte außerdem deinem PATH angehängt sein.

6. Wie kann ich die Installtation von clang überprüfen?:

   a) Starte den *Ausführen* Dialog: Windows Taste + `r`, `cmd` ein und drücke auf Ausführen (dadurch wird die Kommandozeile gestartet).

   b) Gebe den folgenden Befehl in die Kommandozeile ein und drücke Enter: `clang++ -v`

   c) Wenn du die in Abbildung 3 gezeitge Meldung erhältst, wurde clang korrekt installiert und deinem PATH hinzugefügt.

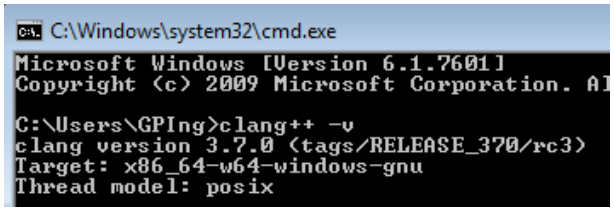

Abbildung 3: Überprüfung der Installation von clang++

Installation von MinGW

Beim Programmieren greifen wir auf bestehende Programmteile und Funktionen, sogenannte Bibliotheken, zu, um nicht ständig das Rad neu erfinden zu müssen. Die wichtigsten Bibliotheken, die zusammen mit C++ definiert sind, nennt man die Standardbibliothek. Leider bringt Clang die Standardbibliothek für Windows nicht selber mit, wir müssen sie also separat installieren. Dazu komt *MinGW* zum Einsatz.

1. Lade MinGW-w64 herunter *(64 bit, version 5.1.0 with posix threads and seh exceptions)*:
   http://sourceforge.net/projects/mingw-w64/files/Toolchains%20targetting%20Win64/Personal%20Builds/mingw-builds/5.1.0/threads-posix/seh/x86_64-5.1.0-release-posix-seh-v4-rev0.7z/download

2. Entpacke den Inhalt des 7z-Archivs direkt nach `C:\`

3. Du solltest nun die Datei `g++.exe` im Verzeichnis `C:\mingw64\bin` finden. Wenn nicht, entpacke die Datein noch einmal und achte darauf, den Ordner `mingw64` *direkt* nach `C:\` zu entpacken.

4. MinGW kann sich nicht selbst dem PATH hinzufügen. Führe daher die folgenden (und in Abbildung 4, 5 und 6 gezeigten Schritte) aus, um dies manuell zu erledigen.

   a) Start->*Rechtsklick* auf Computer->Eigenschaften->Erweiterte Systemeinstellungen

b) Im Reiter *Erweitert*, klicke auf *Umgebungsvariablen*

c) Suche im *unteren Teil des Fensters* nach der Variablen `PATH` und klicke auf *Bearbeiten*.

d) Füge den folgenden Text an den *Wert der Variablen* an: `;C:\mingw64\bin`

e) Achte darauf, das Semikolon (vor dem C) nicht zu vergessen!

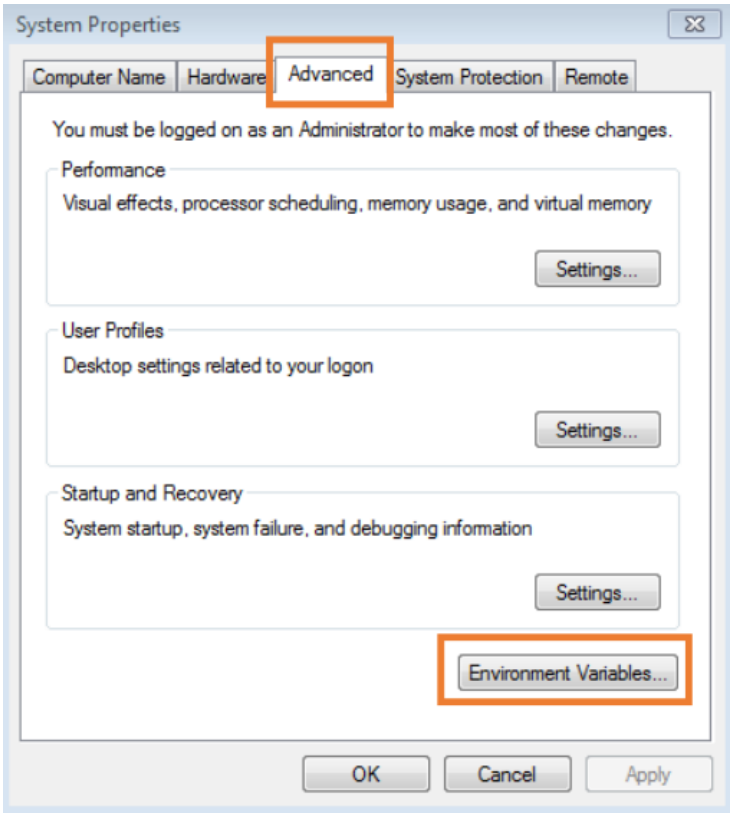

Abbildung 4: Umgebungsvariablen (Environment Variables)

5. Um die Installation zu überprüfen, öffne wieder die Kommandozeile (wie in Abschnitt 6a beschrieben) und gib diesmal `g++ -v` ein. Du solltest in etwa die Ausgabe wie in Abbildung 7 erhalten.

Installation von Geany

Geany ist die Entwicklungsumgebung, in der wir unseren Code schreiben und ausführen.

1. Lade Geany für Windows herunter:
   http://download.geany.org/geany-1.28_setup.exe
2. Führe die Installation mit den Standardeinstellungen durch (siehe Abbildung 8).

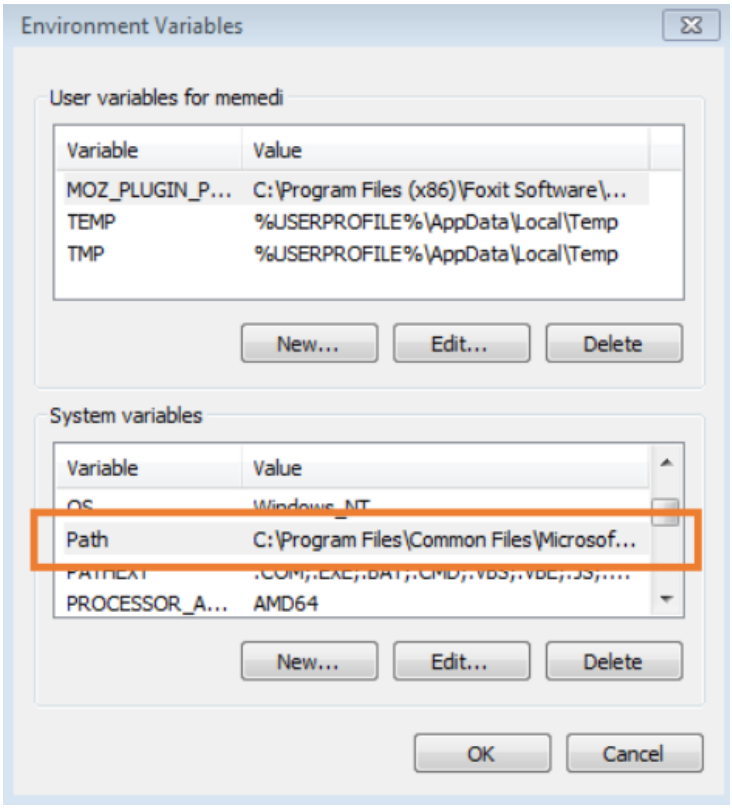

Abbildung 5: Die PATH Variable

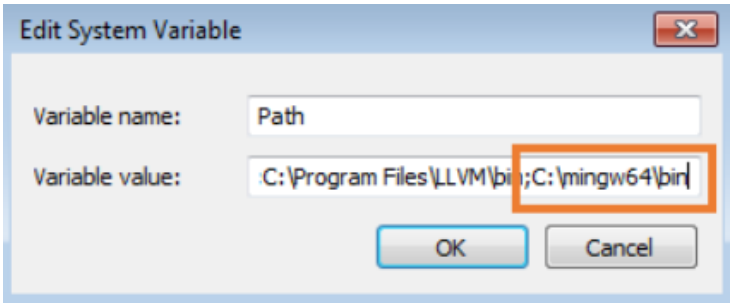

Abbildung 6: Pfad für MinGW dem PATH anhängen

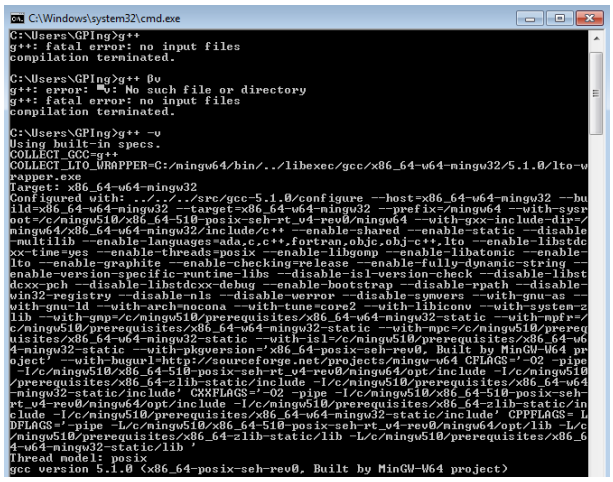

Abbildung 7: Überprüfung der Installation von MinGW

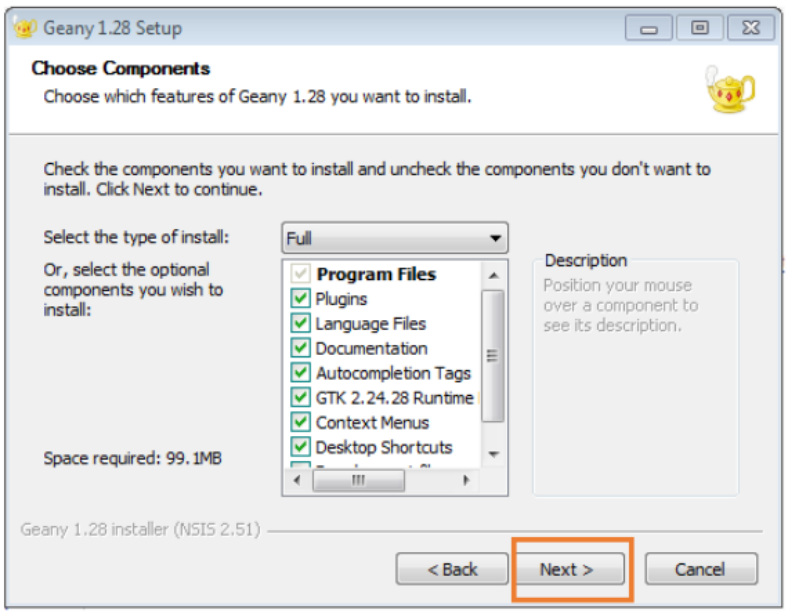

Abbildung 8: Geany setup

0.3 Für Poolräume: Freischalten des Netzlaufwerks und des Windows Logins

Jeder Nutzer mit gültigem IMT-Login verfügt über einen Speicherbereich im IMT. Dieser wird benötigt um in einem Poolraum arbeiten zu können und Daten (wie z.B. die Übungsaufgaben) dauerhaft speichern zu können. Um auf dieses Laufwerk zugreifen zu können, muss jeder Nutzer seinen Bereich zunächst einmalig selber freischalten.

1. Dazu brauchst du einen Rechner mit Internetzugriff. (Wenn du dich selbst nicht einloggen kannst, frage einfach einen hilfsbereiten Kommilitonen der sich bereits anmelden kann).

2. Dann öffnest du im Web-Browser den Bereich Benutzerverwaltung des IMT:
   https://benutzerverwaltung.uni-paderborn.de Auf dieser Seite wählst du „Benutzerdaten selbst verwalten“ aus und loggst dich mit deinem IMT-Login und Passwort an.

3. Wähle den Punkt „Rechnerzugang und Homepage“. Hier kannst du das Verzeichnis für private und öffentliche Daten freischalten. Wenn hier steht, wie viel Speicherplatz auf deinem Laufwerk frei ist, hast du diesen Schritt bereits erledigt.

4. Melde dich von der Seite mittels des Abmelde-Knopfes ab.

5. Warte fünf Minuten (in der Zeit kann sich der Nächste einloggen).

6. Wenn du auf einer virtuelle Maschine angemeldet bist, fahre diese herunter (Start -> Runterfahren) und starte sie neu.

Sobald du dich jetzt auf einem Windows-Rechner mit deinem IMT-Zugang eingeloggt hast, solltest du automatisch ein Laufwerk **U:** haben, dass mit deinem CIF-FS-home-Verzeichnis verknüpft ist. Hier gespeicherte Daten bleiben dauerhaft erhalten und du kannst auch von deinem eigenen Rechner darauf zugreifen (siehe dazu die Hinweise beim IMT).

Weitere Hinweise findest du beim IMT unter: https://hilfe.uni-paderborn.de/Netzwerkspeicher
Bei Problemen kannst du dich auch an das Notebook-café wenden: http://nbc.uni-paderborn.de/

0.4 Im Poolraum anmelden

Je nach Poolraum müsst ihr verschiedene Schritte durchführen, um Geany zu starten:

Raum P1.204 (Maschinenbau Poolraum) Die Rechner in diesem Raum sind Terminals, die sich mit virtuellen Maschinen verbinden – d.h. die Rechner selber dienen nur zur Anzeige, eure Programme werden im Rechenzentrum der Uni ausgeführt. Meldet euch zunächst mit eurem IMT-Account am System an. Dann bekommt ihr eine Liste von möglichen virtuellen Maschinen zur Auswahl. Wählt hier die Maschine **THET-Lehrveranstaltung** aus. Bitte weiterlesen bei Abschnitt *Virtuelle Maschine*.

Raum N5.206 und N2.216 Auf den Rechnern in diesem Räumen läuft ein normales Windows – in den Übungen arbeiten wir jedoch nicht direkt auf den Rechnern sondern auf einer virtuellen Maschine die im Rechenzentrum der Uni ausgeführt wird. Meldet euch zunächst mit eurem IMT-Account am System an. Startet dann das Programm **VMware Horizon View Client** auf dem Desktop und bestätigt so lange, bis ihr eine Auswahl von virtuellen Maschinen bekommt. Wählt hier die Maschine **THET-Lehrveranstaltung** aus.

Virtuelle Maschine: THET-Lehrveranstaltung Standardmäßig ist euer Netzwerkspeicher (siehe Abschnitt 0.3) an das Laufwerk `U:` gebunden; wobei Daten im Unterverzeichnis `data` abgelegt werden.

Legt euch am Besten ein Verzeichnis an, in dem ihr eure Projekte abspeichert. Innerhalb dieses Verzeichnis legt ihr am Besten für jede Übungszettel ein Unterverzeichnis an. Das Ergebnis sieht dann beispielsweise so aus: `U:\data\gping\uebung-01`

In der virtuellen Maschine *THET-Lehrveranstaltung* findet ihr auf dem Desktop eine Verknüpfung mit Geany.

Raum P1.6.12.4 und P7.2.02.01 Auf den Rechnern in diesen Räumen läuft Linux. Du kannst dich auch hier mit deinem IMT-Account anmelden. Geany findest du im Startmenü: `App -> Development (oder Entwicklung) -> Geany`.

Legt euch am Besten ein Verzeichnis an, in dem ihr eure Projekte abspeichert. Innerhalb dieses Verzeichnis legt ihr am Besten für jede Übungszettel ein Unterverzeichnis an. Das Ergebnis sieht dann beispielsweise so aus: `~/data/gping/uebung-01`

Falls das nicht funktioniert, du deinen Netzwerkspeicher aber wie in Abschnitt 0.3 eingerichtet hast, probiere bitte den Pfad `~/gping/uebung-01` aus.

0.5 Geany starten und das erste Programm

- Windows: *Start -> Alle Programme -> Geany -> Geany*
  Es sollte auch eine Verknüpfung auf dem Desktop angelegt worden sein.
- On Linux: Startmenü: *App -> Development (oder Entwicklung) -> Geany*

Wenn Geany läuft kannst du deine erste Datei erstellen:

1. Erzeuge eine neue Datei: *Datei -> Neu* (siehe Abbildung 9)

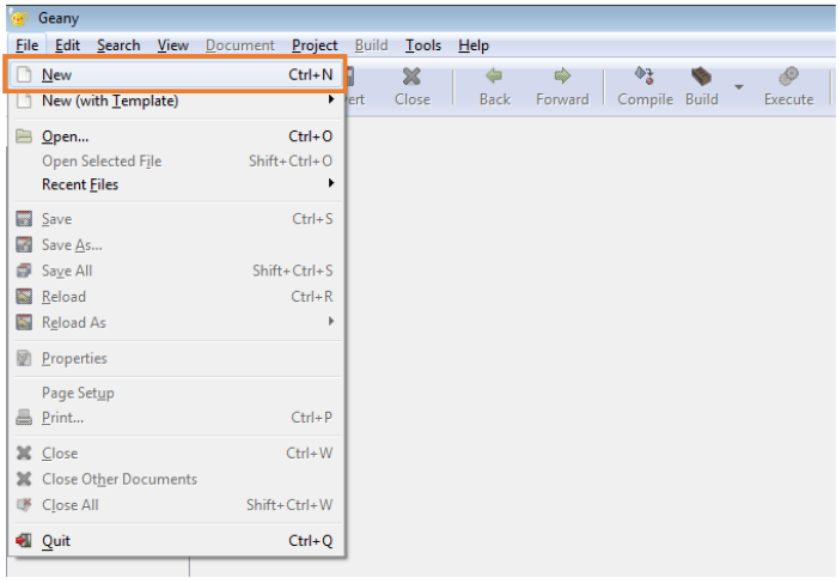

Abbildung 9: Neue Datei erzeugen

2. Tippe den folgenden Code in die neu angelegte Datei (siehe Abbildung 10):

```cpp
#include <iostream>

int main(){
    std::cout<<"Hello World!"<<std::endl;
    return 0;
}
```

3. Speichere die Datei mit der Endung `.cpp`, z.B. als `test.cpp`: *Datei -> Speichern unter* (siehe Abbildung 11):

Geany für clang konfigurieren

Konfiguriere Geany, um clang als C++ compiler zu verwenden. Klicke auf *Erstellen -> Kommandos zum Erstellen konfigurieren* (siehe Abbildung 12). Gebe dort die folgenden Kommandos ein:

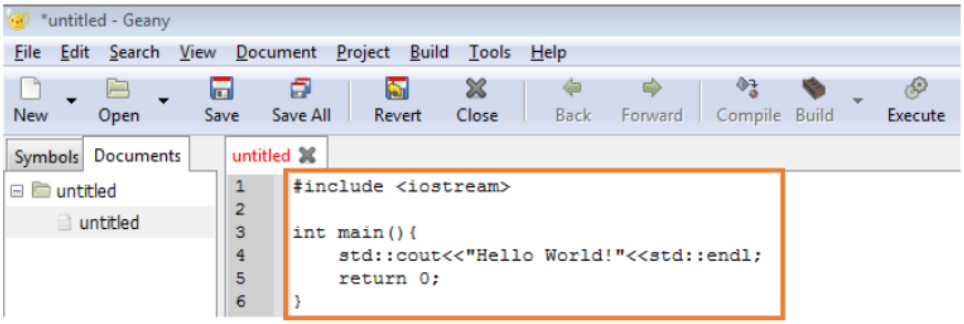

Abbildung 10: Hello World!

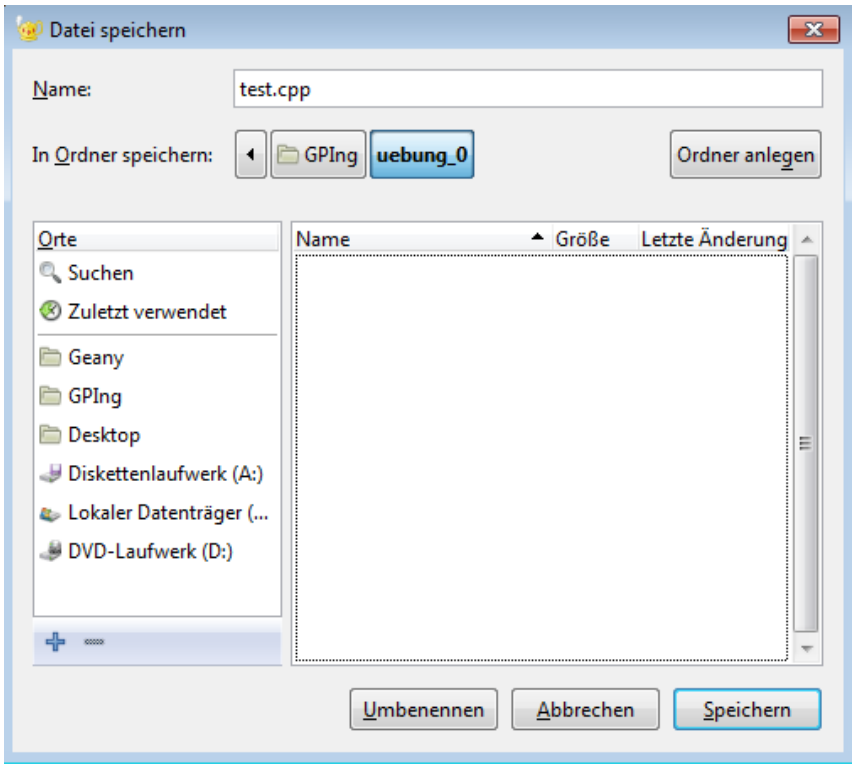

Abbildung 11: Erste Quellcodedatei abspeichern

Compile `clang++ -Wall -c "%f" -std=c++14`

Build `clang++ -Wall "%f" -o "%e.exe" -std=c++14`

Execute `"%e.exe"`

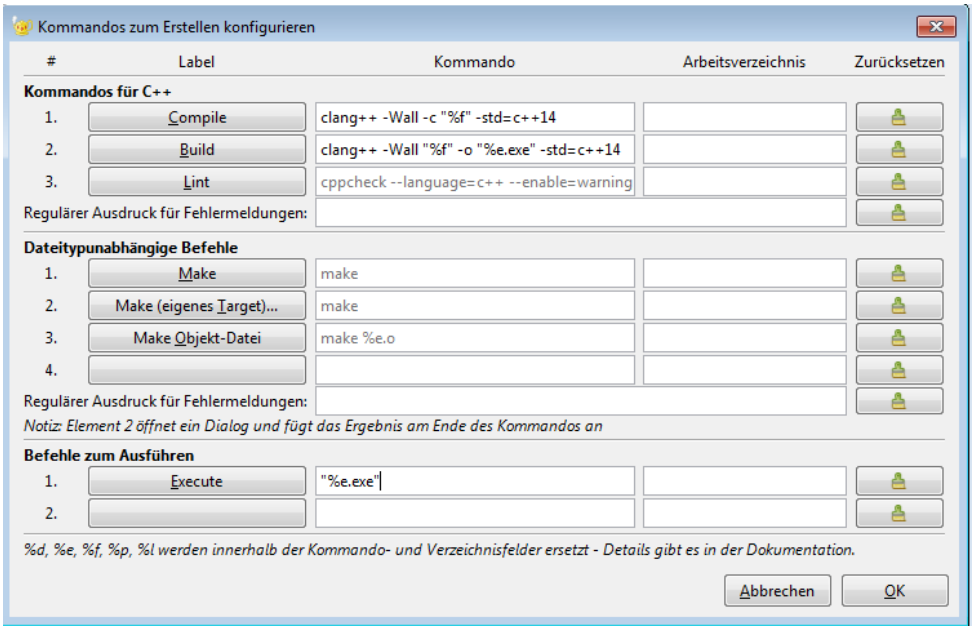

Abbildung 12: Kommandos zum Erstellen konfigurieren um clang zu verwenden

Programm kompilieren

Das vorgefertigte Programm schreibt `Hello World!` auf den Bildschirm (die klassische Ausgabe fast aller ersten Programme). Bevor das Programm ausgeführt werden kann, muss es erst vom Quellcode (Text) in ein ausführbares Binärprogramm übersetzt werden (d.h. das Programm wird kompiliert und gelinkt). Dies könnte z.B. darüber erfolgen, dass man die entsprechenden Compiler-Befehle über die Kommandozeile eingibt.

Geany unterstützt den Entwickler bei diesem Vorgang, indem es die Befehle automatisch ausführt, sobald man auf Erstellen klickt (siehe Abbildung 13). Achte darauf Erstellen und nicht Komplilieren anzuklicken, um ein ausführbares Programm zu erhalten. Bei Änderungen am Quelltext das Speichern vorher nicht vergessen!

In der Ausgabeanzeige (Compiler-Tab am unteren Rand) kann man die ausgeführten Befehle sehen. Sollte die Erstellung des ersten Programms gelungen sein, steht in der Ausgabe etwas wie: `clang++ -Wall ‘‘test.cpp’’ -o ‘‘test.exe’’ -std=c++14 (in directory: C:\Users\memedi\Documents\test)` und `Compilation finished successfully.`

Programm ausführen Um das Progamm auszuführen, klicke auf Ausführen oder drücke F5 (siehe Abbildung 14). Es erscheint ein Kommandozeilenfenster, in dem dein Programm ausgeführt wird. Es sollte in etwa wie in Abbilung 15 aussehen und den Text `Hello World!` anzeigen.

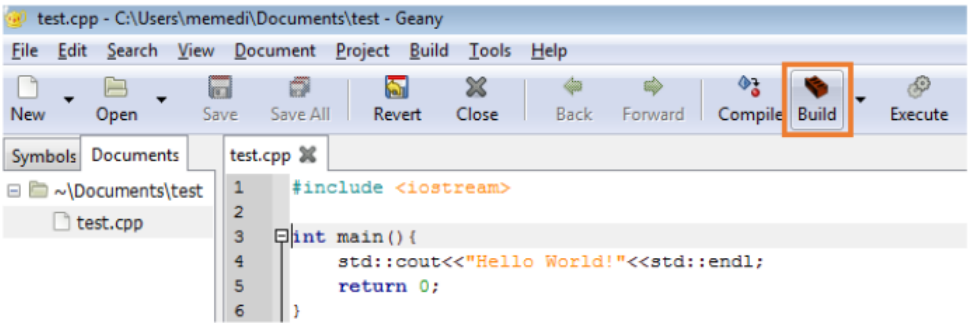

Abbildung 13: Build

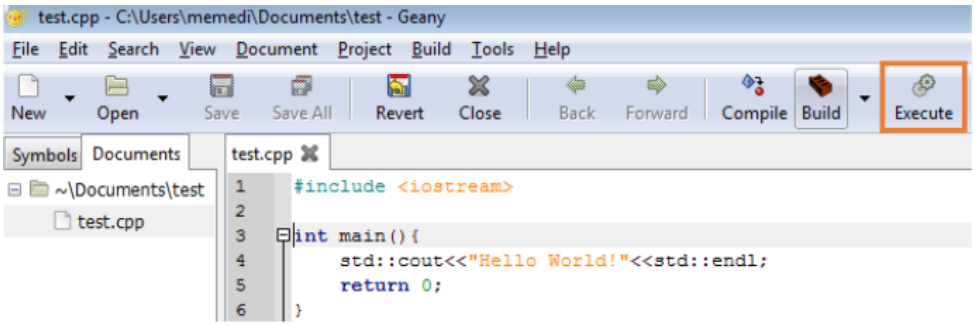

Abbildung 14: Execute

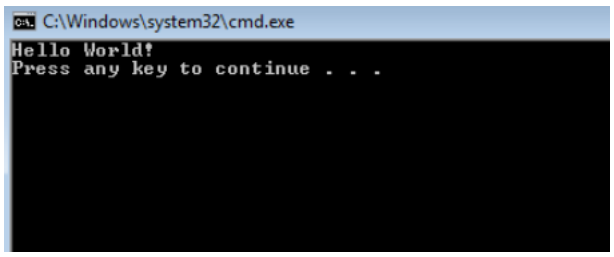

Abbildung 15: Hello World! output

0.6 Aufgabe

Ändere das Programm so, dass es `Hallo Paderborn!` ausgibt.