

Installation einer C-Entwicklungsumgebung

Für Erstsemester im Rahmen des C-Vorkurses

Vincent Lutsch Hauke Kersten

Fachschaftsrat Informatik

12. Oktober 2025

Hinweise

- ▶ Bitte immer zuerst zuschauen und zuhören und danach selbst durchführen.
- ▶ Bei Problemen meldet euch. Wir versuchen, so schnell wie möglich zu helfen. Versucht gern auch euch gegenseitig zu helfen.
- ▶ Die Anleitung nimmt an, dass ihr Windows 11 oder ein aktuelles macOS verwendet
- ▶ Das folgende Setup bietet unserer Meinung nach einen guten Kompromiss aus Installationsaufwand und Benutzerfreundlichkeit. Es sei jeder Person selbst überlassen, zu experimentieren und das beste Setup für die eigene Produktivität herauszufinden :)
- ▶ Für größere C-Projekte würden wir dieses Setup nur eingeschränkt empfehlen, in Imperative Programmierung im ersten Semester sollte es hiermit aber keine Probleme geben. Falls doch, würden wir uns über Hinweise freuen.

Table of Contents

Voraussetzungen

Installation eines C-Compilers

Kompilieren einer C-Quellcode-Datei

Installation einer Entwicklungsumgebung

Verwendung des Terminals in VS Code

Verwendung von GCC im Terminal

Voraussetzungen

WLAN einrichten

- ▶ Netzwerkname: eduroam
- ▶ Identität: <Nutzerkürzel>@uni-rostock.de
- ▶ Passwort: <Nutzerpasswort>
- ▶ Verschlüsselung: WPA2-Enterprise
- ▶ Authentifizierung: PEAP → MSCHAPv2

Weitere Informationen und ausführliche Anleitungen:

<https://www.itmz.uni-rostock.de/onlinedienste/uninetz-zugang/wlan/eduroam/>

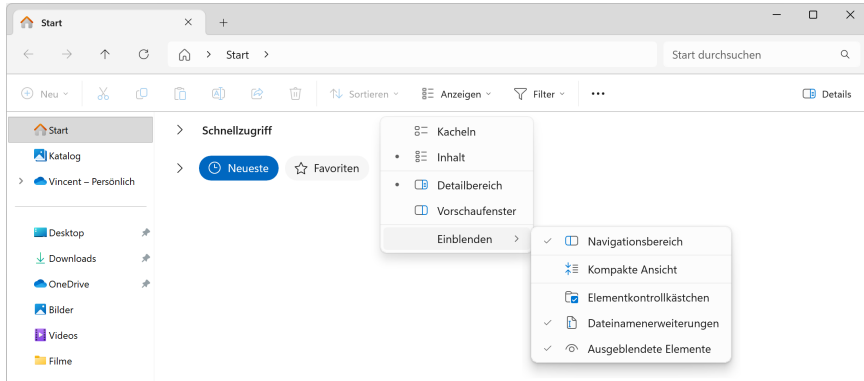
Anzeigen von Dateinamenerweiterungen

- ▶ Die Zeichen nach dem letzten Punkt im Dateinamen werden als Dateinamenerweiterung (*filename extension*) oder Dateinamensuffix bezeichnet
- ▶ Zeigt Format einer Datei an, um sie mit einem passenden Programm zu öffnen
- ▶ Um versehentliche Änderungen zu verhindern, werden sie unter Windows und macOS standardmäßig ausgeblendet
- ▶ Bsp.: `test.exe` → Dateinamenerweiterungen: `exe`
- ▶ Problem: Es wird `test.c` angezeigt, aber tatsächlich heißt die Datei `test.c.txt`

Anzeigen von Dateinamenerweiterungen

Unter Windows

- Öffne den Explorer (Win + E), klicke oben in der Menüleiste auf **Anzeigen** und wähle dann unter **Einblenden** die **Dateinamenerweiterungen** aus



Anzeigen von Dateinamenerweiterungen

Unter macOS

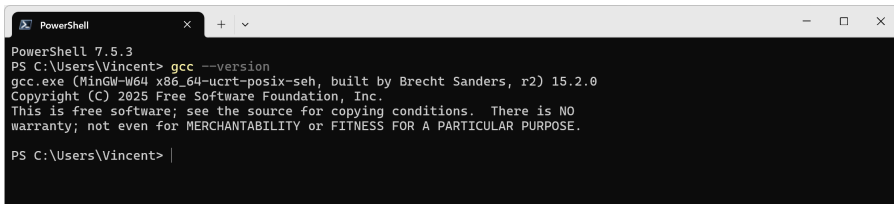
- ▶ Öffne den Finder, klicke oben auf **Finder** und wähle unter **Einstellungen** **Erweitert** aus
- ▶ Aktiviere das Feld **Alle Dateinamensuffixe einblenden**

Installation eines C-Compilers

Installation eines C-Compilers unter Windows

mit WinGet, dem Windows Package Manager

- ▶ Öffne das Startmenü, suche nach `Terminal` und starte es
- ▶ Gib `winget install BrechtSanders.WinLibs.POSIX.UCRT` ein und bestätige mit Enter
- ▶ Wenn der Installationsprozess abgeschlossen ist, starte das Terminal neu und überprüfe die Installation mit `gcc --version`



```
PowerShell 7.5.3
PS C:\Users\Vincent> gcc --version
gcc.exe (MinGW-W64 x86_64-ucrt-posix-seh, built by Brecht Sanders, r2) 15.2.0
Copyright (C) 2025 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

PS C:\Users\Vincent> |
```

Installation eines C-Compilers unter macOS

- ▶ Öffne ein Terminal, gib den Befehl `gcc --version` ein und drücke Enter
- ▶ Es sollte ein Vorschlag erscheinen, GCC jetzt zu installieren
- ▶ Bestätige die Installation, starte nach der Installation das Terminal neu und gib erneut den obigen Befehl ein, um die Installation zu überprüfen

Kompilieren einer C-Quellcode-Datei

Schreiben eines ersten Programms in C

Erstellen der Datei im Terminal

- ▶ Öffne ein Terminal und navigiere zum Desktop, indem du `cd Desktop` eingibst (Falls du unter Windows deinen Desktop per OneDrive gesichert hast, musst du erst in den OneDrive-Ordner navigieren und dann zum Desktop. Z. B. mit `cd OneDrive\Desktop`)
- ▶ Erstelle einen neuen Ordner mit dem Befehl `mkdir c-kurs` (verwende keine Leerzeichen in Datei- und Ordernamen)
- ▶ Navigiere in den neuen Ordner mit `cd c-kurs`
- ▶ Erstelle eine neue Textdatei und öffne sie, indem du Folgendes eingibst:
 - ▶ Windows: `notepad test.c`
 - ▶ macOS: zuerst `touch test.c`, dann `open -e test.c`

Schreiben eines ersten Programms in C

Den ersten C-Code schreiben

- ▶ Schreibe die folgenden 6 Zeilen in die Datei:

```
#include <stdio.h>
int main()
{
printf("Hallo Rostock!");
return 0;
}
```

- ▶ Speichere die Datei

Schreiben eines ersten Programms in C

Den Code kompilieren und das Programm ausführen

- ▶ Gehe zurück in das noch offene Terminal und gib `gcc test.c` ein
- ▶ Gib `ls` ein, um dir die Inhalte des Ordners anzeigen zu lassen
- ▶ In der Liste solltest du die Datei `a.exe` (Windows) bzw. `a.out` (macOS) sehen, die gerade durch den Aufruf von GCC erzeugt wurde
- ▶ Du kannst das Programm im Terminal ausführen, indem du den Namen `.\a.exe` bzw. `./a.out` eingibst
(`.\` bzw. `./` zeigt an, dass sich die auszuführende Datei im aktuellen Verzeichnis befindet)
- ▶ Als Ausgabe des Programms solltest du `Hallo Rostock!` im Terminal sehen
- ▶ Du hast erfolgreich dein erstes Programm geschrieben, kompiliert und ausgeführt!

Installation einer Entwicklungsumgebung

Installation einer Entwicklungsumgebung

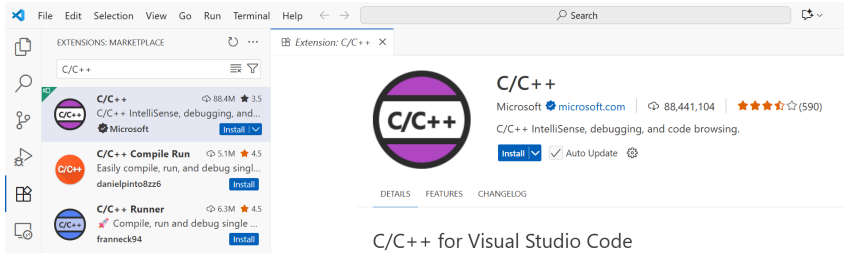
Visual Studio Code installieren und starten

- ▶ Visual Studio Code herunterladen (<https://code.visualstudio.com>) und installieren (alle Einstellungen im Setup können beibehalten werden) (Unter Windows mit `winget install Microsoft.VisualStudioCode` möglich)
- ▶ VS Code starten

Installation einer Entwicklungsumgebung

VS Code starten und Erweiterung C/C++ installieren

- ▶ Wir empfehlen euch die Nutzung von VS Code auf Englisch, da dies die Fehlersuche im Internet einfacher macht.
- ▶ Erweiterung **C/C++** installieren
 - ▶ Extension Menu öffnen (4 Bauklötze in der linken Seitenleiste)
 - ▶ nach **C/C++** suchen und auf **Install** klicken



Installation einer Entwicklungsumgebung

Ordner und C-Datei öffnen

- ▶ Explorer öffnen (oberstes Symbol in der linken Seitenleiste)
- ▶ auf `Open Folder` klicken und im sich öffnenden Fenster nach dem Ordner `c-kurs` suchen, diesen auswählen und auf `Ordner auswählen` klicken
- ▶ `Yes, I trust the authors` auswählen
- ▶ links im Explorer sollten nun der Ordner und die enthaltenen Dateien angezeigt werden
- ▶ klicke auf die Datei `test.c`, um sie zu öffnen

Installation einer Entwicklungsumgebung

Code-Formatierung

- ▶ Drücke `Shift + Alt + F`, um den Code automatisch zu formatieren
- ▶ Dies ändert nur das Aussehen, nicht die Funktionalität des Codes
- ▶ Generell gilt es als guter Stil, den Quellcode einheitlich und gut leserlich zu formatieren, damit sich auch andere schnell darin zurechtfinden
(verschiedene Programmiersprachen haben verschiedene Best Practices in Bezug auf Code-Formatierung und Benennung von Variablen)
- ▶ Immer wenn ein Punkt neben dem Dateinamen erscheint, gibt es nicht gespeicherte Änderungen
- ▶ Drücke `Strg + S` bzw. `Cmd + S`, um die Änderungen zu speichern

Verwendung des Terminals in VS Code

Verwendung des Terminals in VS Code

- ▶ Klicke in der Menüleiste oben in VS Code auf `View` und im Dropdown unten auf `Terminal`
- ▶ Das Terminal startet immer in dem Ordner, der gerade links geöffnet ist
- ▶ Es kann genau so verwendet werden wie das externe Terminal
- ▶ Probiere eine Änderung in `test.c` vorzunehmen und die Datei unter Verwendung des internen Terminals zu kompilieren und auszuführen
- ▶ Achte vor dem Kompilieren darauf, dass *kein* Punkt neben dem Dateinamen steht. Immer erst speichern und dann kompilieren!

Verwendung des Terminals in VS Code

Tipps zum Umgang mit Terminals

- ▶ Mit den Pfeiltasten nach oben und unten kann man auf die zuvor ausgeführten Befehle zugreifen
- ▶ Mit der Tabulator-Taste kann man Eingaben automatisch vervollständigen (durch mehrmaliges Drücken kann man verschiedene Vorschläge ansehen)
- ▶ Durch Drücken von `Esc` wird die Eingabezeile geleert
- ▶ Mit dem Befehl `clear` oder `cls` kann man den sichtbaren Terminalinhalt entfernen

Verwendung von GCC im Terminal

Verwendung des C-Compilers

- ▶ GCC (der Compiler) erstellt standardmäßig immer eine ausführbare Datei (auch Executable genannt) mit dem Namen `a.exe` bzw. `a.out`
- ▶ Dabei wird eine ggf. vorhandene gleichnamige Datei überschrieben
- ▶ Durch anhängen der Option `-o <Dateiname>` kann man den Compiler anweisen der ausgegebenen Datei einen bestimmten Namen zu geben, z.B. `gcc test.c -o hallo_rostock` (keine Leerzeichen im Dateinamen verwenden)
- ▶ Die Datei mit dem Quellcode muss in dem Verzeichnis liegen, das aktuell im Terminal geöffnet ist. Überprüfe den aktuellen Pfad, der links von der Eingabe im Terminal angezeigt wird.
- ▶ Nutze `cd <Verzeichnisname>` oder `cd ..`, um das Verzeichnis zu wechseln und `ls`, um dir die Inhalte des aktuellen Verzeichnisses anzusehen.

Merke

- ▶ Nach jeder Änderung am Quellcode müssen die folgenden Schritte erneut durchgeführt werden, um die Änderungen im Programm zu sehen:
 1. Speichern
 2. Kompilieren
 3. Programm erneut ausführen
- ▶ Ihr werdet euch oft genug wundern, warum eure Änderungen am Quellcode nichts bewirken, weil ihr sie nicht gespeichert habt :)