

Einführung in die Informatik
Ausarbeitung Übung 2

Julian Bertol

November 13, 2023

1 compile and debug

1.1 Problem

Ich muss einen euklerischen Algorithmus erstellen. Diesen dann mit gcc oder g++ compielieren.

Danach muss ich erklären was gcc, g++ und gdb ist.

1.2 Lösungskonzept

Ich erstelle das Programm und teste dann was gcc, g++ und gdb machen. Wenn ich nicht weiter weis benutze ich das internet.

1.3 konkrete Lösung

euklerischer Algortihmus:

```
#include <stdio.h>
int main() {
    int number1;
    int number2;
    int ergebnis;
    int ergebnis_rest;
    printf("Geben_Sie_die_erste_Zahl_ein:_\n");
    scanf("%i", &number1);
    printf("Geben_Sie_die_zweite_Zahl_ein:_\n");
    scanf("%i", &number2);
    do {
        ergebnis = number1 / number2;
        ergebnis_rest = number1 % number2;
        number1 = number2;
        number2 = ergebnis_rest;
    } while (ergebnis_rest > 0);
    printf("Das_Ergebnis_lautet:_%i\n", number1);
    return 0;
}
```

Zum compielieren müssen wir mit dem command cd in das Verzeichniss gehen wo die Datei liegt. Danach compielieren wird die c++ datei entweder mit dem Befehl:

```
g++ "dateiname" -o "neuer Dateiname"
```

oder

```
gcc "dateiname" -o "neuer Dateiname"
```

mit dem befehl chmod +x "dateiname"

muss man die Zugrifssrechte auf das asuführen der Datei bearbeiten.

Nun kann man mit dem Befehl ;

```
./"dateiname"
```

die compelierte Datei ausführen.

gdb ist ein Debugger. Man kann den Debugger mit dem Befehl:
gdb "dateiname" aufrufen.

1.4 Tests

Ausführen des programmes und nachrechnen.

2 Aufgabe 2 Build

2.1 2.1 make

Make ist ein build management tool. Ein Build-Management-Tool ist ein Softwarewerkzeug, das den Prozess der Kompilierung, des Testens und der Bereitstellung von Quellcode automatisiert, um eine ausführbare Anwendung oder ein Softwarepaket zu erstellen. Dieser Prozess wird als "Build" bezeichnet, und das Build-Management ist entscheidend für die Entwicklung von Softwareprojekten. Einige der Hauptaufgaben eines Build-Management-Tools umfassen

erstellen einer makefile

Zuerst erstellt man eine .make datei. In dieser werden dann Infortmationen zu make gespeichert.

```

# Variable f r Compiler-Flags. Das W steht f r Warnungen aktivieren
# und das g f r Debug-Informationen
Coption = -W -g

# Regel f r das Standardziel "all"
all: euklerisch

# Regel f r das Ziel "euklerisch"
euklerisch: euklerisch.cpp
            g++ euklerisch.cpp $(Coption) -o euklerisch

# Regel f r das Ziel "ex" (Ausf hren)
ex: euklerisch
    ./euklerisch

# Regel f r das Ziel "db" (Debugging)
db: euklerisch
    gdb euklerisch

# Regel f r das Ziel "rm" (L s chen)
rm: euklerisch
    rm euklerisch
    rm euklerisch.o

```

2.2 2.2 cmake

Cmake ist an sich das selbe wie make, nur dass cmake auf allen betriebsystemen anwendbar ist und make nur auf Unix systemen.

```

cmake_minimum_required(VERSION 3.22.1) #Version angeben ist hilfreich
#fuer komplexe Projekte, die spezifische Funktionen oder Befehle
#beim Build besitzen
project(euklerisch) #Name des Projekts
add_executable(euklerisch euklerisch.cpp) #Erstellt passende
#build-Dateien f"ur das Projekt euklerisch.cpp

```

- **Worin unterscheidet sich das Ziel von make und cmake? Beschreiben Sie in eigenen Worten die Aufgabe von make.**

Make ist eine automatisierende anwendung die den compiler automatisiert.

Außerdem kann man sein Programm debuggen ohne IDEA.

Der unterschied zwischen make und cmake besteht darin, dass cmake auf mehreren Plattformen ausführbar ist und make nur auf Unix.

- **Was sollte das PHONY-Target clean machen?**
 .phony sorgt dafür, dass der Befehl clean ausgeführt wird und make nicht nach einer Datei namens clean sucht und diese versucht ausführbar zu machen. Das Phony target clean sollte die Makefile2-Datei im CMakeFiles Ordner bereinigen.
- **Wie wurde die Funktionserfüllung geprüft?**
 Die Funktionserfüllung wurde überprüft, indem die Makefile Datei mit dem Befehl make ausgeführt wurde. Es entsteht ein ausführbares Programm, welches man mit dem Terminalbefehl ./"Dateiname" ausführen kann. Durch einsetzen von Testwerten kann man erkennen, dass das Programm das macht, was es soll.

3 Resümee zur dieser "Übungsaufgabe"

Dauer für

- Durchführung 1.5 Stunden
- Dokumentation 0.5 Stunden

Welche großen Probleme waren zu lösen? Die make-file war mein größtes Problem, da ich mich nicht mit diesem System auskannte.