



Leibniz
Universität
Hannover

ASSIGNMENT 04

Programmieren I

14.11.2024

NORWIN BERTRAM WIESECKE

10057811

TIMON WELLHAUSEN

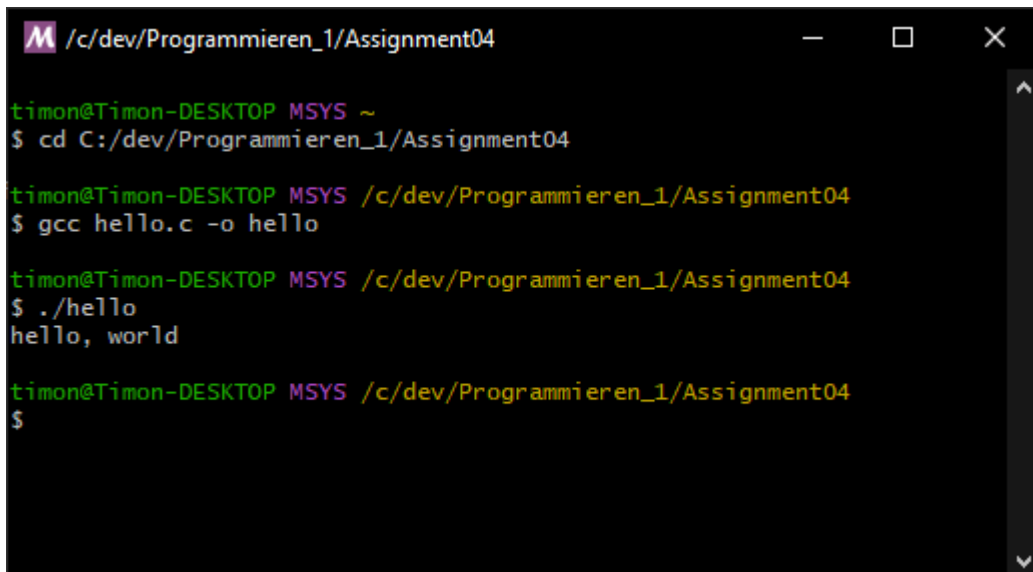
10041137

AUFGABE 1:

- a) Testen Sie die Installation indem Sie `gcc -v` auf der Kommandozeile aufrufen. Fügen Sie die Ausgabe die Sie erhalten in ihre Abgabe ein.

```
$ gcc -v
Using built-in specs.
COLLECT_GCC=gcc
COLLECT_LTO_WRAPPER=/usr/lib/gcc/x86_64-pc-msys/13.2.0/lto-wrapper.exe
Target: x86_64-pc-msys
Configured with: /c/S/B/src/gcc-13.2.0/configure --build=x86_64-pc-msys --
prefix=/usr --libexecdir=/usr/lib --enable-bootstrap --enable-shared --enable-
shared-libgcc --enable-static --enable-version-specific-runtime-libs --with-
arch=nocona --with-tune=generic --disable-multilib --enable-__cxa_atexit --
with-dwarf2 --enable-languages=c,c++,lto --enable-graphite --enable-
threads=posix --enable-libatomic --enable-libgomp --disable-libitm --enable-
libquadmath --enable-libquadmath-support --disable-libssp --disable-win32-
registry --disable-symvers --with-gnu-ld --with-gnu-as --disable-lsl-version-
check --enable-checking=release --without-libiconv-prefix --without-libintl-
prefix --with-system-zlib --enable-linker-build-id --enable-libstdcxx-
filesystem-ts
Thread model: posix
Supported LTO compression algorithms: zlib
gcc version 13.2.0 (GCC)
```

- b) Fügen Sie einen Screenshot der Ausführung Ihrer Abgabe bei.



```
/c/dev/Programmieren_1/Assignment04
timon@Timon-DESKTOP MSYS ~
$ cd C:/dev/Programmieren_1/Assignment04

timon@Timon-DESKTOP MSYS /c/dev/Programmieren_1/Assignment04
$ gcc hello.c -o hello

timon@Timon-DESKTOP MSYS /c/dev/Programmieren_1/Assignment04
$ ./hello
hello, world

timon@Timon-DESKTOP MSYS /c/dev/Programmieren_1/Assignment04
$
```

- c) Welche Funktion haben die folgenden Terminal Befehle? Geben Sie einen Beispielaufruf an und fügen Sie, falls es eine Ausgabe auf der Konsole gibt, diese der Abgabe bei:

1. mkdir: Erlaubt Interaktionen mit Ordnern oder das Erstellen von neuen Ordnern

```
$ mkdir --version
```

```
mkdir (GNU coreutils) 8.32
```

```
Copyright (C) 2020 Free Software Foundation, Inc.
```

```
License GPLv3+: GNU GPL version 3 or later  
<https://gnu.org/licenses/gpl.html>.
```

```
This is free software: you are free to change and redistribute it.
```

```
There is NO WARRANTY, to the extent permitted by law.
```

```
written by David Mackenzie.
```

2. ls: Listet alle Dateien und Ordner im Verzeichnis, in dem man sich befindet.

```
$ ls
```

```
Assignment_04_Norwin_Bertram_wiesecke_10057811_Timon_wellhausen_10041137.doc  
x
```

```
build
```

```
hello.c
```

```
hello.exe
```

```
puzzle.pf
```

```
task4
```

```
turn-signal.pf
```

```
volume.pf
```

```
water_sim_graphics.pf
```

```
'~$signment_04_Norwin_Bertram_wiesecke_10057811_Timon_wellhausen_10041137.do  
cx'
```

3. Cd: Wechselt den Dateipfad zum angegebenen Pfad

- d) 1. Geben Sie zwei verschiedene Wege an wie Sie mit `cd` in das Verzeichnis `folder/A01` gelangen können.

```
$ cd folder/A01
```

```
$ cd folder; cd A01
```

2. Geben Sie zwei verschiedene Wege an wie Sie mit dem `cd` vom Verzeichnis `A01` in das Verzeichnis `A02` gelangen können.

```
$ cd ../A02
```

```
$ cd folder; cd A02
```

3. Was passiert, wenn Sie in Ihrem ursprünglichem Verzeichnis `cd folder/../folder/A01` eingeben?

`folder` wird aufgerufen, ein Verzeichnis wird zurückgegangen, `folder` wird erneut aufgerufen und `A01` wird aufgerufen.

4. Was passiert, wenn Sie statt `ls` „`ls -al`“ aufrufen?

`ls` wird mit den Parametern `-a` und `-l` ausgeführt.

`-a` listet alle Dateien und Ordner auf, auch die, die mit „`.`“ beginnen. Dazu zählen auch „`.`“ und „`..`“, wobei „`.`“ das aktuelle Verzeichnis und „`..`“ das darüberliegende Verzeichnis bezeichnet.

`-l` gibt ein langes Listenformat aus. Dabei werden zu der jeweiligen Datei oder dem jeweiligen Ordner verschiedene weitere Informationen angegeben.

AUFGABE 2:

In einem PostFix-Programm sollen verschiedene Formen von geometrischen Körpern, nämlich Zylinder, Kugel und Quader, repräsentiert werden. Entwickeln Sie eine Funktion, die diese geometrischen Körper verarbeiten kann (Parametertyp: `Shape`) und das zugehörige Volumen berechnet.

Führen Sie die im Skript unter Recipe for Variant Data beschriebenen Schritte durch. Verwenden Sie die Template-Datei `volume.pf`. Bearbeiten Sie die mit `todo` markierten Stellen. Geben Sie je ein Beispiel (Eingaben und erwartete Resultate) für jede Variante an.

Siehe „*Assignment_04_Norwin_Bertram_Wiesecke_10057811_Timon_Wellhausen_10041137_volume.pf*“.

AUFGABE 3:

Entwickeln Sie eine Blinkersteuerung für Autos. Es gibt ein linkes und ein rechtes Blinklicht. Das linke Blinklicht wird mit der l-Taste, das rechte Blinklicht mit der r-Taste aktiviert bzw. deaktiviert. Ein aktives Blinklicht soll mit einer Frequenz von 1Hz blinken (500 ms an, 500 ms aus). Es dürfen nicht beide Blinklichter zur gleichen Zeit aktiv sein. Verwenden Sie für diese Aufgabe turn-signal.pf.

Siehe „Assignment_04_Norwin_Bertram_Wiesecke_10057811_
Timon_Wellhausen_10041137_turn-signal.pf“.

AUFGABE 4:

In dieser Aufgabe soll für die Wassersimulation von letzter Woche eine Visualisierung erstellt werden. Sie können dafür entweder Ihre Lösung von letzter Woche oder die auf StudIP bereitgestellt Musterlösung verwenden.

Siehe „Assignment_04_Norwin_Bertram_Wiesecke_10057811_
Timon_Wellhausen_10041137_water_sim_graphics.pf“.

AUFGABE 5:

Implementieren Sie die fehlenden Funktionen des 15-Puzzles. Eine detaillierte Beschreibung des Spiels findet sich hier: <https://de.wikipedia.org/wiki/15-Puzzle>. Die Template-Datei ist puzzle.pf.

Siehe „Assignment_04_Norwin_Bertram_Wiesecke_10057811_
Timon_Wellhausen_10041137_puzzle.pf“.