

## CDT C-Environment

Das Fach CDT verwendet im C-Teil die üblichen Linux-Tools um Programme zu erstellen. Das sind folgende:

- gcc / g++ ( Compiler und Linker )
- gdb ( Debugger )
- make ( Build-Script interpreter )

Hier beschrieben ist eine Umgebung mit der Version 10.10 von Ubuntu, die Sie unter <http://www.ubuntu.org> erlangen können. Für die Praktikas ist es Ihnen jedoch freigestellt, auch eine andere Umgebung zu wählen. Nachdem Sie diese Version von Ubuntu installiert haben, sind die Tools für die Entwicklung bereits verfügbar. Verwenden Sie eine andere Plattform, muss die Software gegebenenfalls noch installiert werden. Ob alles vorhanden ist und soweit funktioniert, können Sie in jedem Fall mit folgendem Beispiel überprüfen.

### ***Beispiel ohne IDE***

Erstellen Sie ein kleines HelloWorld-Programm mit einem beliebigen Editor und speichern Sie die Datei ab unter dem Namen "hello.c".

```
#include <stdio.h>

int main(int argc, char **argv) {
    printf("HelloWorld\n");
    return 0;
}
```

Das Programm lässt sich nun wie folgt kompilieren und linken:

```
gcc -g -c -o hello.o hello.c
gcc -g -o hello hello.o
```

Mit einem Makefile lässt sich diese Abfolge von Befehlen kapseln, so dass das Programm mit dem einfachen Aufruf "make" erstellt und mit "make clean" wieder gelöscht werden kann. Erstellen Sie dafür eine neue Datei mit dem Name "makefile" und folgendem Inhalt:

```
default: hello

hello: hello.o
    gcc -g -o hello hello.c

hello.o: hello.c
    gcc -c -g -o hello.o hello.c

clean:
    rm -f hello.o
    rm -f hello
```

Starten Sie das Programm mit dem Debugger.

```
gdb hello
```

Da das Executable mit Debug-Symbols erstellt wurde (-g), können sie mit symbolischen Namen Breakpoints setzen z. B. In der Funktion „main“ mit „break main“. Eine Liste der vorhandenen Haltepunkte zeigt „info breakpoints“ an. Der Befehl „run“ startet das Programm. Sobald ein Breakpoint erreicht wird, haltet das Programm und der Debugger zeigt die aktuelle Zeile an. Mit „list“ kann man die 10 folgenden Zeilen aus dem Quellcode zusätzlich auflisten lassen. Drücken Sie „c“ und Enter, um das Programm weiter laufen zu lassen. Wenn es fertig ist, kann der Debugger mit „quit“ beendet werden.

Wenn dieses Beispiel auf Ihrem Rechner läuft, sind die notwendigen Tools für das Praktikum vorhanden, wenn auch noch etwas umständlich zu bedienen. Abhilfe schafft eine IDE für die C-Entwicklung.

## ***Installation von Eclipse CDT***

Für Ubuntu 10.10 ist bereits ein Paket verfügbar, um Eclipse inklusive einer Erweiterung für C zu installieren - Es soll das Plugin Eclipse CDT verwendet werden ( <http://www.eclipse.org/cdt> ). Sie können das Paket als Root mit Apt installieren.

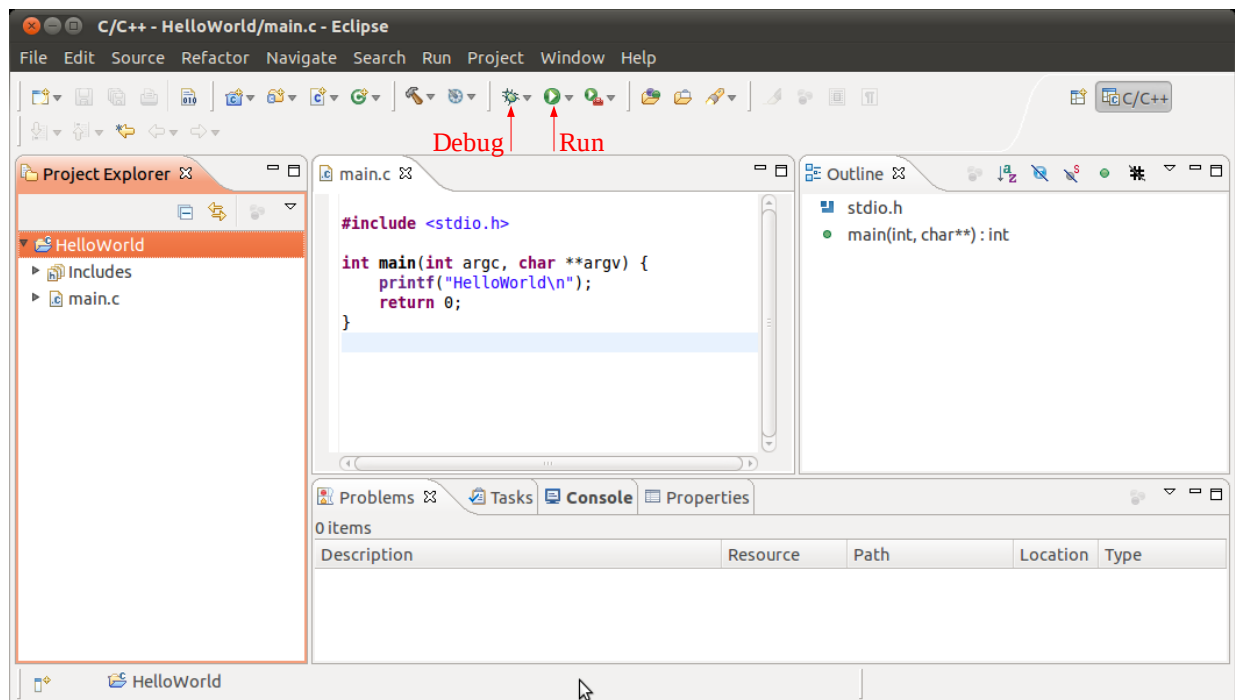
```
apt-get install eclipse-cdt
```

In dieser Dokumentation soll das Paket-System aber umgangen werden. Gehen Sie dafür auf die Seite <http://www.eclipse.org/cdt> und laden Sie die aktuelle von Version Eclipse CDT herunter. Das ist im Falle dieses Beispiels die Datei eclipse-cpp-helios-SR2-linux-gtk.tar.gz, die Sie mit „tar xpvBf Datei“ vom Terminal aus entpacken können.

Im neu erstellten Ordner „eclipse“ befindet sich ein Executable „eclipse“, mit dem Sie das Programm starten können. Oder Sie erstellen auf dem Desktop einen Launcher, der auf diese Datei verweist. In diesem Beispiel wurde Eclipse im Home-Verzeichnis des Benutzers „zhaw“ entpackt und ist damit nur für diesen Benutzer verfügbar. Möchten sie, dass alle Benutzer die Software verwenden können, verschieben Sie den Ordner z. B. in das Verzeichnis „/opt“. Dafür benötigen Sie jedoch Root-Rechte auf dem Rechner. Diese können Sie je nach Distribution mit „sudo su“ oder „su root“ erlangen.

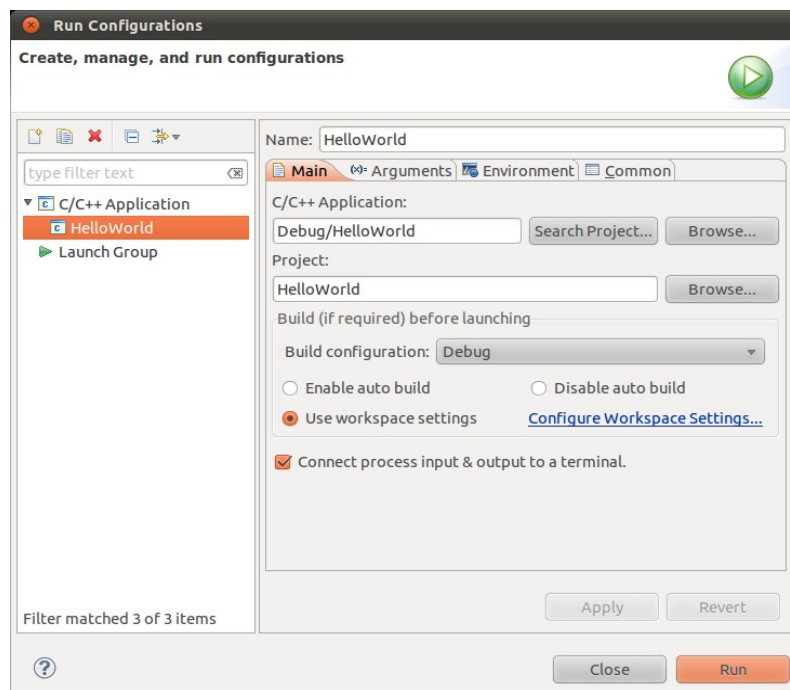
## ***Beispiel mit Eclipse CDT***

Wenn Sie Eclipse CDT starten, fragt Sie die Software nach dem Workspace, ein Dialog, den Sie im Normalfall einfach bestätigen können. Erstellen Sie ein neues Project mit „File->New->Other“ und dann „C Project“ aus dem Ordner „C/C++“. Es soll der Name „HelloWorld“ und der Typ „Empty Project“ gewählt und der Dialog beenden werden. Das Projekt ist links im Project-Explorer sichtbar und Sie können mit „New->File“ aus dem Kontext-Menü die Datei „main.c“ erstellen. Verwenden Sie den gleichen Source-Code, der bereits im Beispiel ohne IDE verwendet wurde. Das Programm soll also lediglich die Zeichenkette „HelloWorld“ ausgeben.



Mehr braucht es nicht für ein lauffähiges Programm. Wählen Sie „Project->Build All“ aus dem Hauptmenu von Eclipse. Die IDE kompiliert und linkt nun Ihr main.c und erstellt so das Executable „HelloWorld“.

Drücken Sie den Knopf „Run“ und wählen Sie aus dem Menu den Eintrag „Run Configurations“. Im folgenden Dialog kann man mit dem Knopf oben Links (New launch configuration) eine neue Launch-Konfiguration vom Typ „C/C++ Application“ erstellen.



Wenn Sie in diesem Dialog „Run“ drücken, wird das Programm ausgeführt und der Text „HelloWorld“ in der Konsole von Eclipse ausgegeben.

Die gleiche Launch-Konfiguration kann man auch zum debuggen verwenden. Die IDE weist Sie darauf hin, dass das debuggen intern mit einer View verknüpft ist und fragt, ob diese geladen werden soll. Bestätigen Sie diese Frage, damit Eclipse in den Debug-Modus wechselt.

Wenn im Source-Code ein Breakpoint gesetzt ist, wird Ihr Programm an dieser Stelle angehalten und man kann mit Eclipse den weiteren Ablauf steuern, das ist durchaus etwas komfortabler als das debuggen in der Kommandozeile.

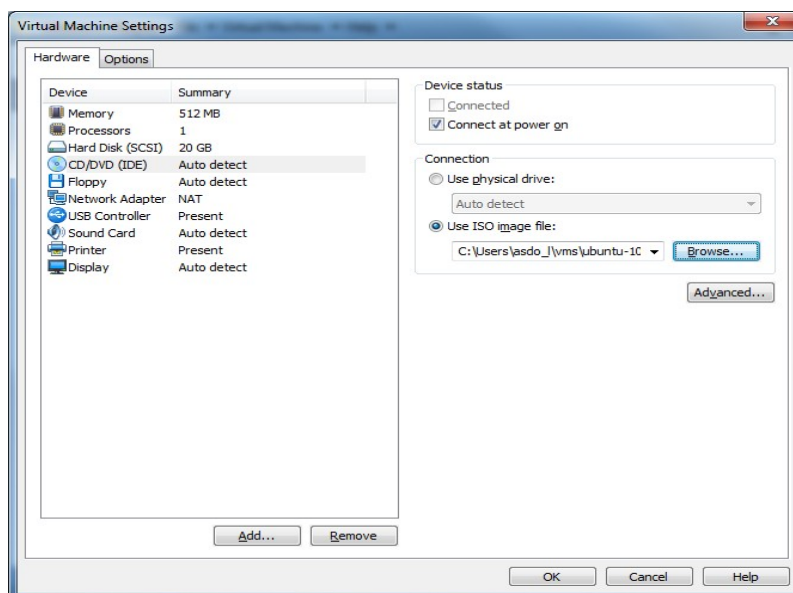
## ***Virtnelle Maschine***

Auch wenn man unter Windows mit Cygwin ( <http://www.cygwin.com> ) eine adäquate Umgebung aufbauen kann, wird empfohlen, auf einem Unix-Betriebssystem zu arbeiten. Es gibt in der Zwischenzeit diverse Anbieter von Software, die einen Computer virtuell emulieren. Dazu gehören VMware, deren Player frei verfügbar ist ( <http://www.vmware.com/products/player/overview.html> ) und VirtualBox von Oracle ( <http://www.virtualbox.org> ). Die kurze Beschreibung hier bezieht sich auf den VMware-Player.

Nachdem die Software installiert und gestartet wurde, können Sie entweder eine neue virtuelle Maschine erstellen oder eine Bestehende öffnen.

Es liegt auf [\\pools01.t.t-stud-cdt](http://pools01.t.t-stud-cdt) sowohl eine vorinstallierte VM, wie auch eine ISO-Image bereit, mit dem eine neu VM aufgesetzt werden kann.

Wenn eine neue VM gemacht werden soll, drücken Sie den entsprechende Knopf und wählen Sie im Dialog die Option „Betriebssystem später installieren“. In den folgenden Fenster muss das Betriebssystem der VM angegeben werden, ihr Name, der Ort im Filesystem und die maximale Grösse der Harddisk. Nun ist sie bereit für die Installation. Sie können in den Einstellungen der VM angeben, dass direkt ein ISO-File als CD-Laufwerk dienen soll.



Wenn die VM jetzt gestartet wird, wird vom ISO-Image gebootet und der Setup des OS erscheint. Die Maschine kann man wie üblich installieren.

Den kleinsten Aufwand hat man, wenn die bestehende VM vom Pool verwendet wird. Darin sind alle Tools bereits installiert. Sie müssen aber den Ordner „UbuntuCdt“ herunterladen und lokal abspeichern. Das sind knapp 5GB. Sie können dann das UbuntuCdt.vmx mit dem VMware-Player öffnen und die VM direkt starten. Wählen Sie die Option „I copied it“, wenn das Programm danach fragt. Das hat den Effekt, dass für das virtuelle Ethernet-Interface eine neue MAC-Adresse generiert wird. Da in den Einstellungen für das Netzwerk NAT eingestellt ist, spielt das jedoch keine grosse Rolle.

Benutzername: zhaw

Passwort: zhaw

Root-Passwort: zhaw

asdo