

Einstieg in die Programmierung mit C

Übersicht

- Warum C?
- Speicher
- Maschinencode und Compilierung
- Compiler installieren
- „hello, world“

Warum C?

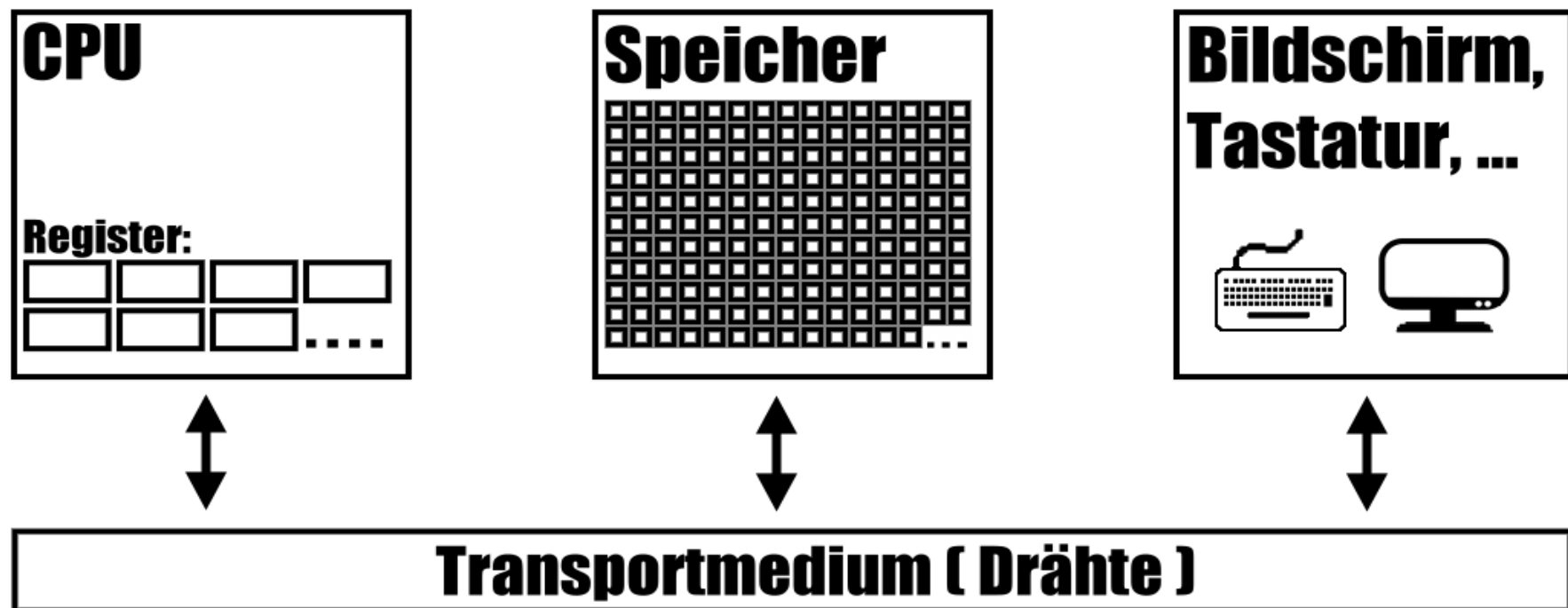
- Weit verbreitete, wichtige Programmiersprache
- Viele weitere, wichtige Sprachen wurden von C abgeleitet:
 - ◆ C++, C#, Java, ...
- Imperative Sprache
- Systemnahes Programmieren ist möglich
- Der „imperative Teil“ einer objektorientierten Sprache wie C++ aber auch Java benutzt die Syntax von C
 - ◆ Im Wesentlichen
- Es gibt Compiler für alle Betriebssysteme

Übersicht

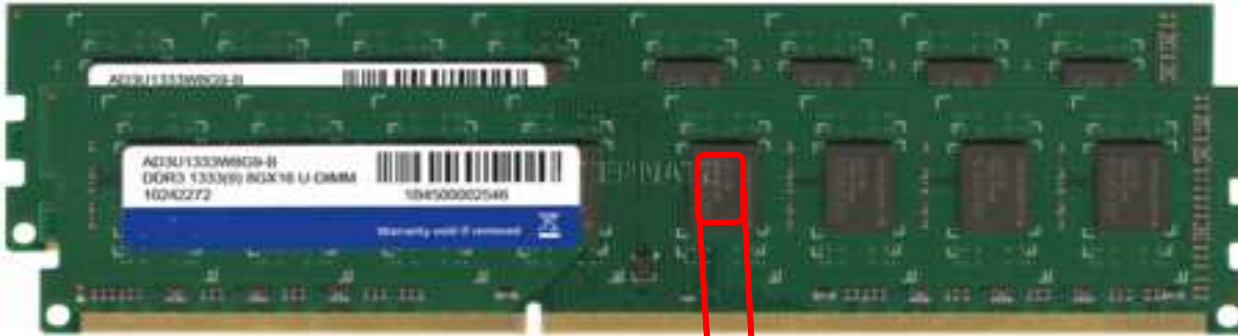
- Warum C?
- Speicher
- Maschinencode und Compilierung
- Compiler installieren
- „hello, world“

Speicher eines PC

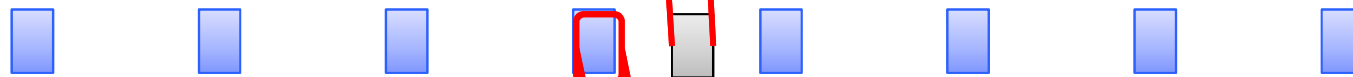
- Im Folgenden grober Überblick über den Aufbau
 - ◆ Soviel man braucht um ein Bild zu bekommen:



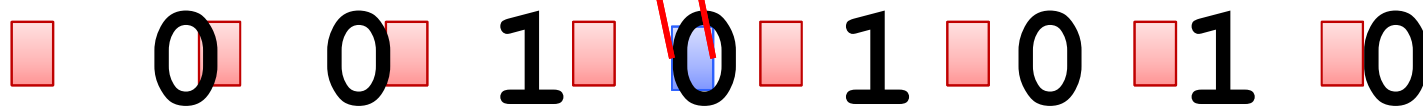
Speicher eines PC



Aktueller DDR3-Speicher




8 Byte
x 1024-kB
x 1024-MB
x 1024-GB



8 Bit

Speicher: Zahlenformate

- 1Byte = 8Bit = 2^8 = 256 Zustände
- Speichern von Zahlen:
 - ◆ Ganze Zahlen von 0 – 255
 - ◆ Ganze Zahlen von -127 – 127
 - ◆ Ganze Zahlen von -128 – 127 (Zweierkomplement)
- Unterschied Speicher und Interpretation:
 - ◆ Datentyp!
- Zusammenschließen von Bytes erlaubt größere Zahlen:
 - ◆ 2Byte = 16Bit = 65536 Zustände
 - ◆ 4Byte = 32Bit = 4294967296 Zustände
 - ◆ 8Byte = 64Bit = 18446744073709551616 Zustände

 Komplexere Datentypen:
float, double zur Darstellung
reeller Zahlen

Übersicht

- Warum C?
- Speicher
- Maschinencode und Compilierung
- Compiler installieren
- „hello, world“

Compiler: Vom Code zum Maschinencode

- Zusammenhänge: C und Assembler
 - ◆ Hochsprachen sind abstrakt und dadurch einfach zu gebrauchen
 - ◆ Code wird vom Compiler in Maschinsprache übersetzt
 - ⇒ Für Hardwarenahe Programmierung (in C) ist ein grobes Verständnis des Zusammenhangs mit dem Maschinencode wichtig.
 - ⇒ In diesem Kurs noch nicht.
 - ◆ Wie werden die folgenden Konzepte im Maschinencode realisiert?
 - ⇒ Variablen
 - ⇒ Datentypen
 - ⇒ Funktionen

Code vs. Maschinencode

Assembler-Ausgabe erstellt mit: `gcc -S add.c`

C-Code

```
int add(int i,int j)
{
    int p = i + j;
    return p;
}
```

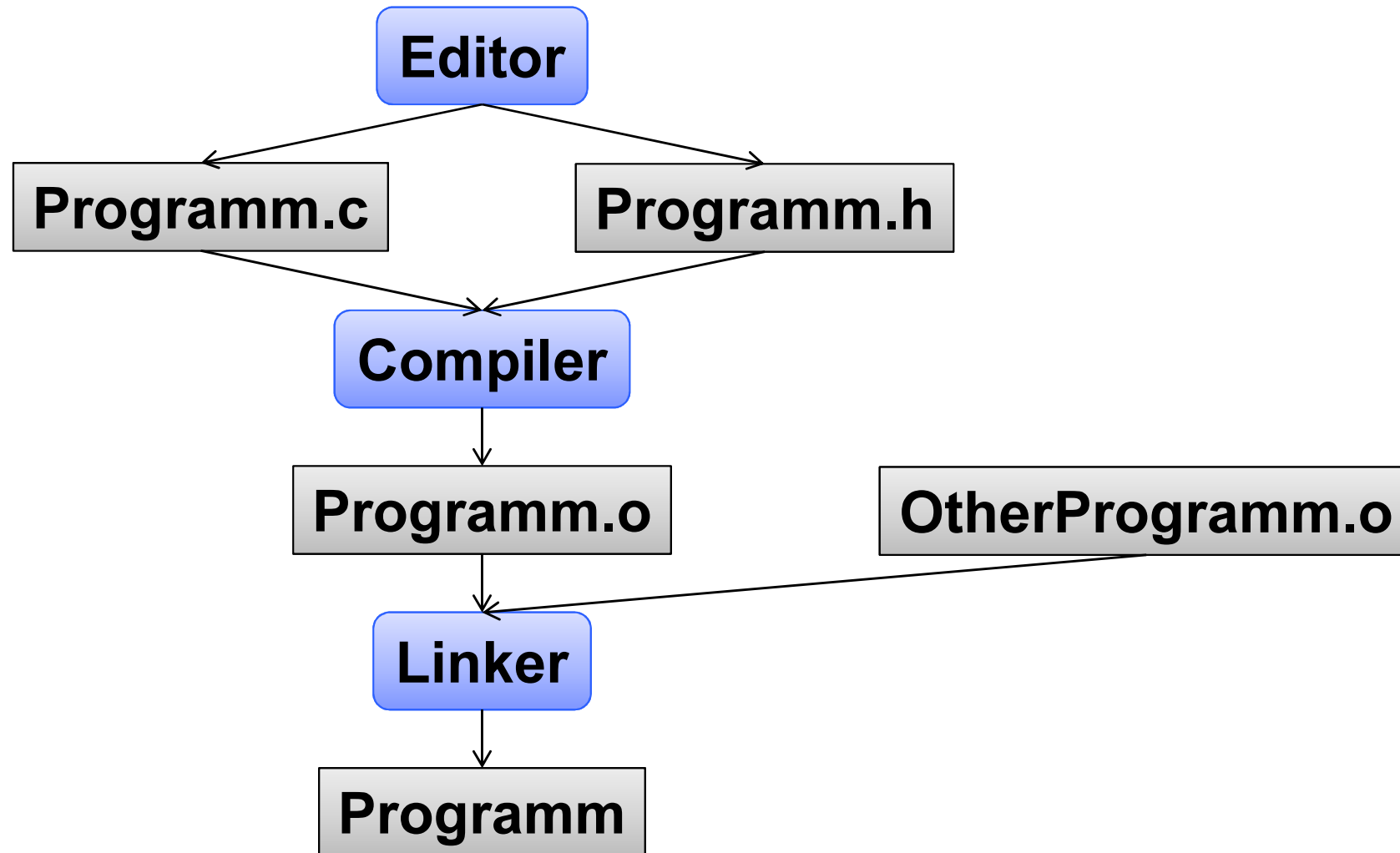
Assembler

```
add:
    pushl %ebp
    movl %esp, %ebp
    subl 4, %esp
    movl 8(%ebp), %edx
    addl 12(%ebp), %edx
    movl %edx, -4(%ebp)
    movl -4(%ebp), %eax
    leave
```

Variablen

- Inhalt von Variablen liegt im Speicher
- Während der Übersetzung verwaltet der Compiler intern eine Tabelle mit den Speicheradressen der Variablen
- Im Maschinen-Code gibt es keine Variablen

Übersetzung C nach Assembler



Übersicht

- Warum C?
- Speicher
- Maschinencode und Compilierung
- Compiler installieren
- „Hello, world“

How to install GCC at windows

- Aber wie programmiert man denn jetzt?
- Compiler für C:
 - ◆ Gcc unter <http://gcc.gnu.org/>
 - ◆ Für Linux-Benutzer
 - ◆ Bei den meisten Distributionen bereits installiert!
- Cygwin unter <http://www.cygwin.com/>
 - ◆ Für Windows-Benutzer.
 - ◆ Nun kann man die GCC auch unter Windows benutzen

How to install GCC at windows

- Hier gibt es die setup.exe:
 - ◆ www.cygwin.com/setup.exe
- Die muss nur noch ausgeführt werden.
 - ◆ Es sind nur wenig Veränderungen an den Standard-Einstellungen nötig, um auch den GCC mit zu installieren.
- Schritt 0: Runterladen und Ausführen

How to install GCC at windows

- Einige Tipps am Anfang:
 - ◆ Am besten werden alle Programme in einem Unterordner abgespeichert.
 - ◆ Leerzeichen in den Namen sollten vermieden werden
 - ◆ „ä“, „ö“ und „ü“ machen auch immer wieder Probleme, also am besten weglassen oder „ae“, „oe“ oder „ue“ schreiben

How to install GCC at windows

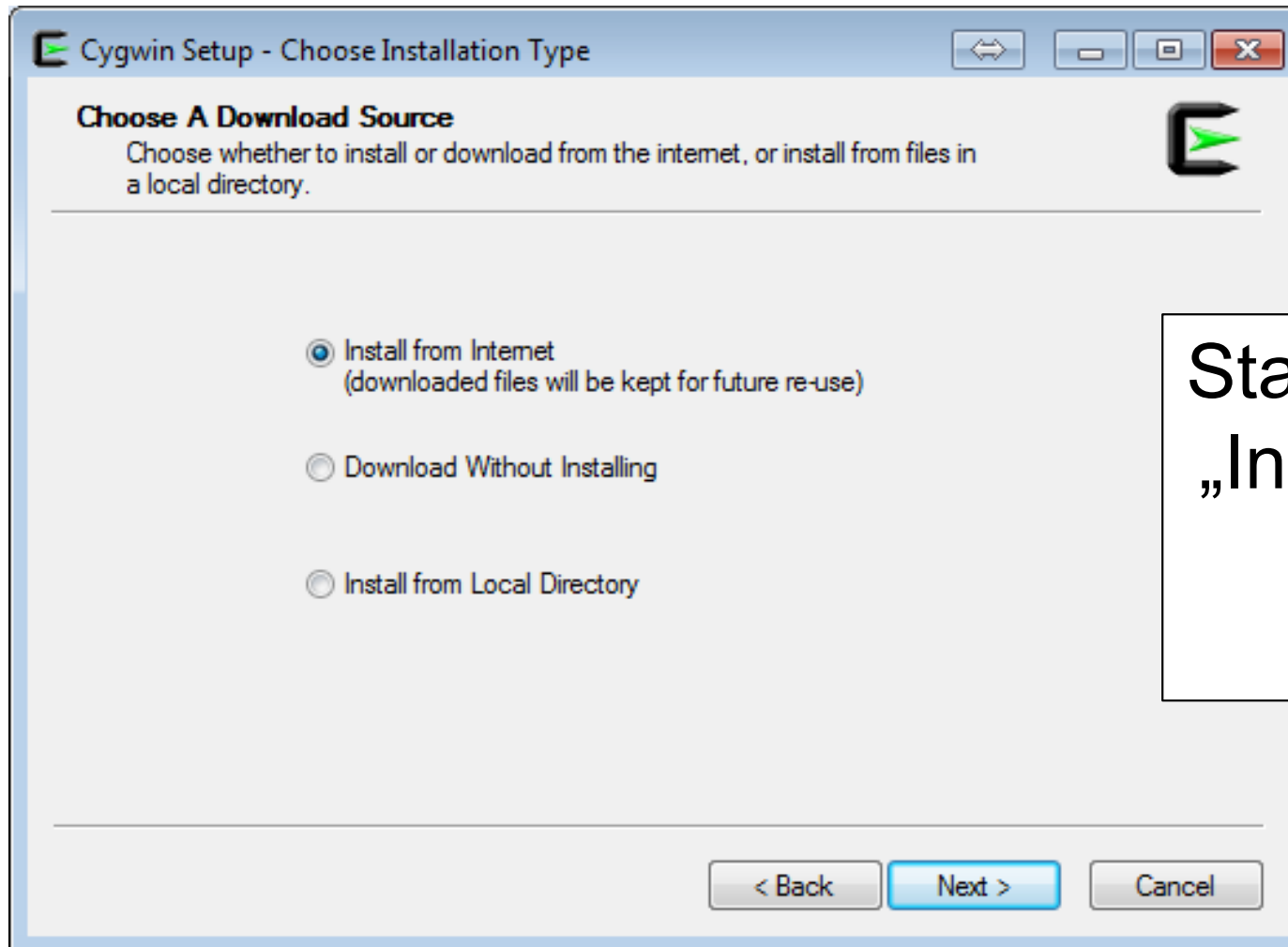
- Schritt 1: Wenn ihr die heruntergeladene Datei startet, findet ihr folgende Anzeige:



Todo:
„Next“ klicken

How to install GCC at windows

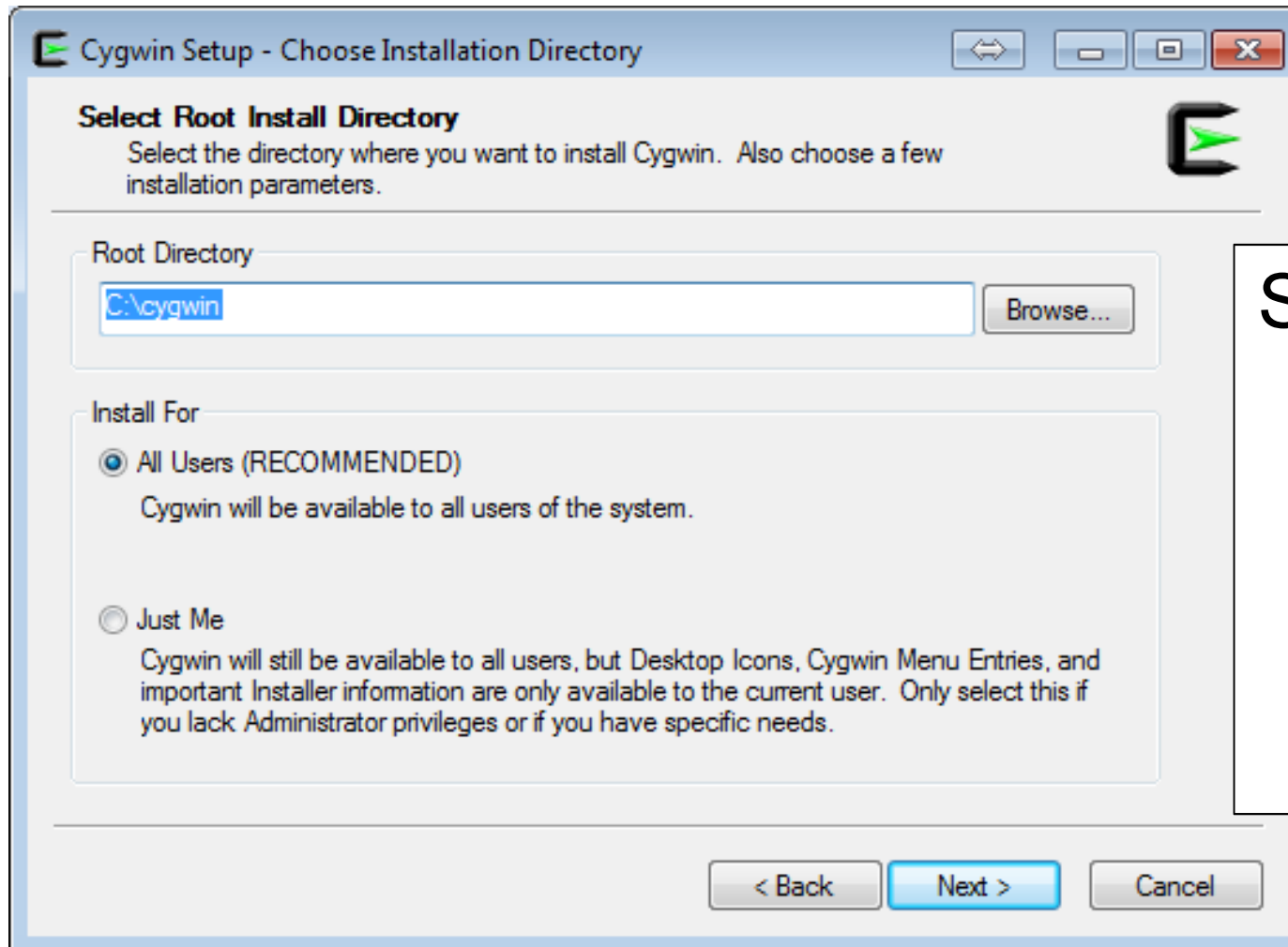
- Schritt 2: Von welcher Quelle soll Cygwin installiert werden?



Standard-Einstellung:
„Install from Internet“
Todo:
„Next“ klicken

How to install GCC at windows

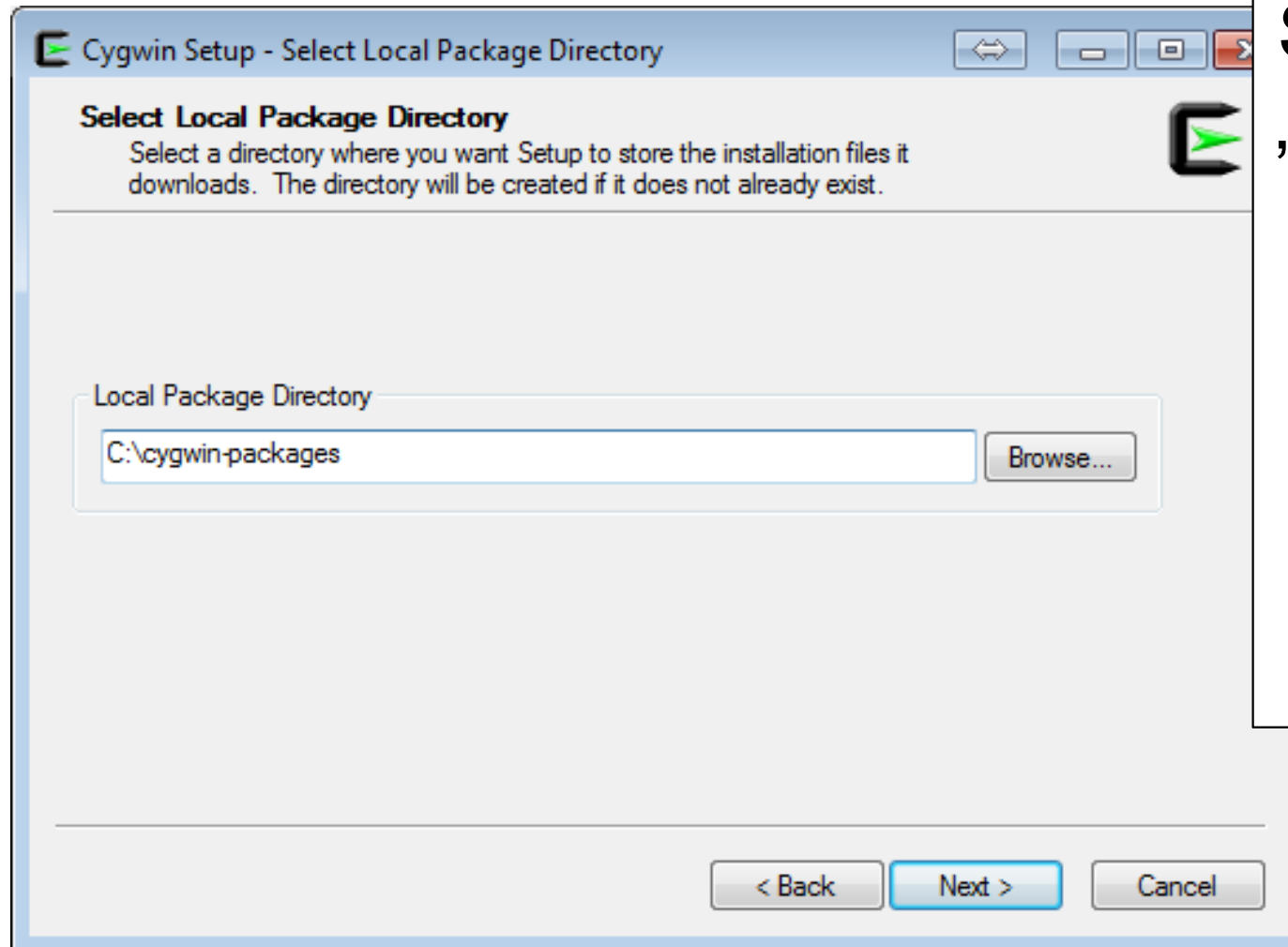
- Schritt 3: Ordner auswählen, in dem Cygwin installiert werden soll



Standard-Einstellung:
„C:\cygwin“
Todo:
Ordner erstellen
Pfad eingeben
„Next“ klicken

How to install GCC at windows

- Schritt 4: Cygwin benötigt einen temporären Ordner.



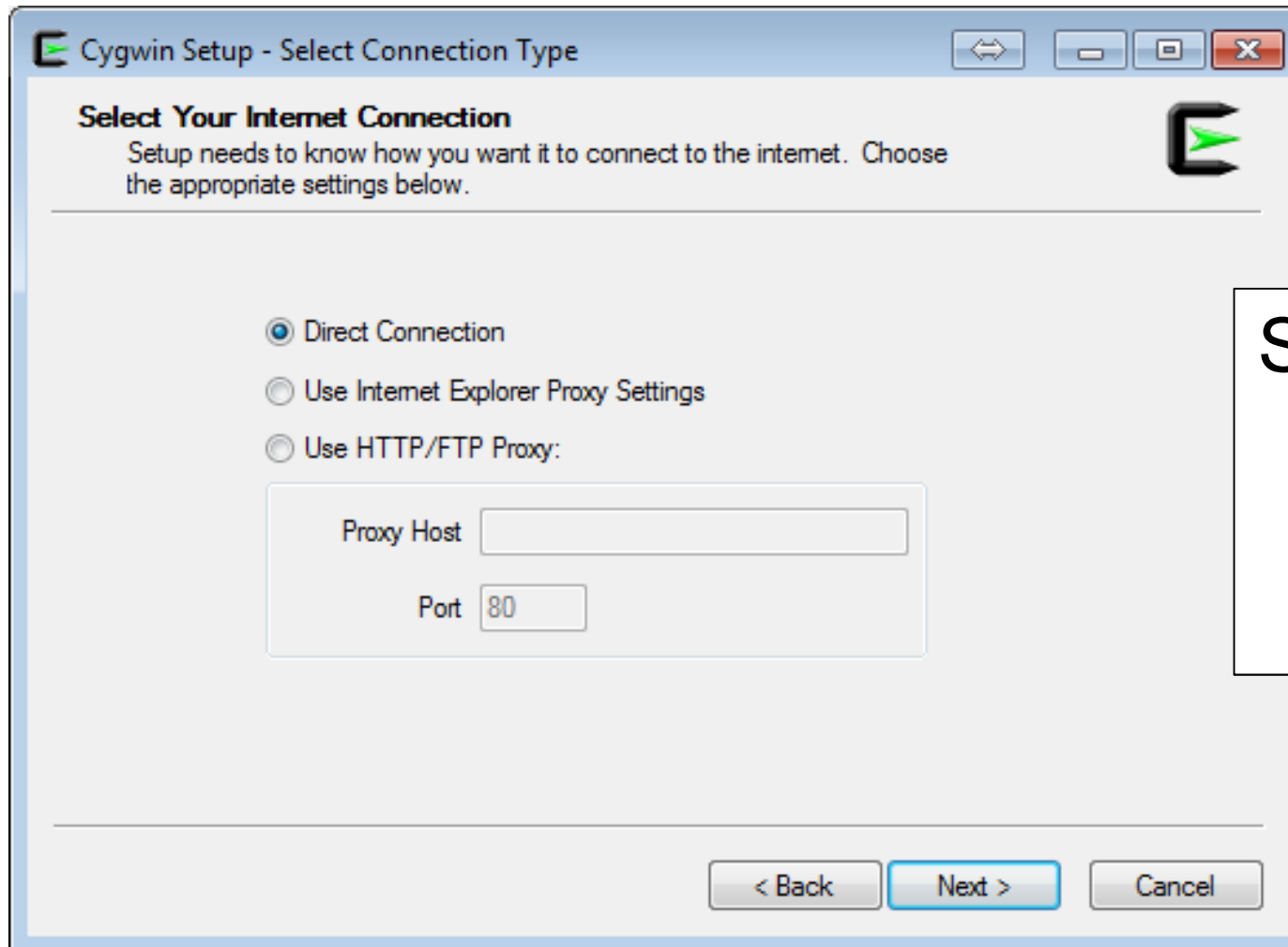
Standard-Einstellung:
„C:\cygwin-packages“

Todo:

„Next“ klicken
Nach der Installation
kann dieser Ordner
gefahrlos wieder
gelöscht werden

How to install GCC at windows

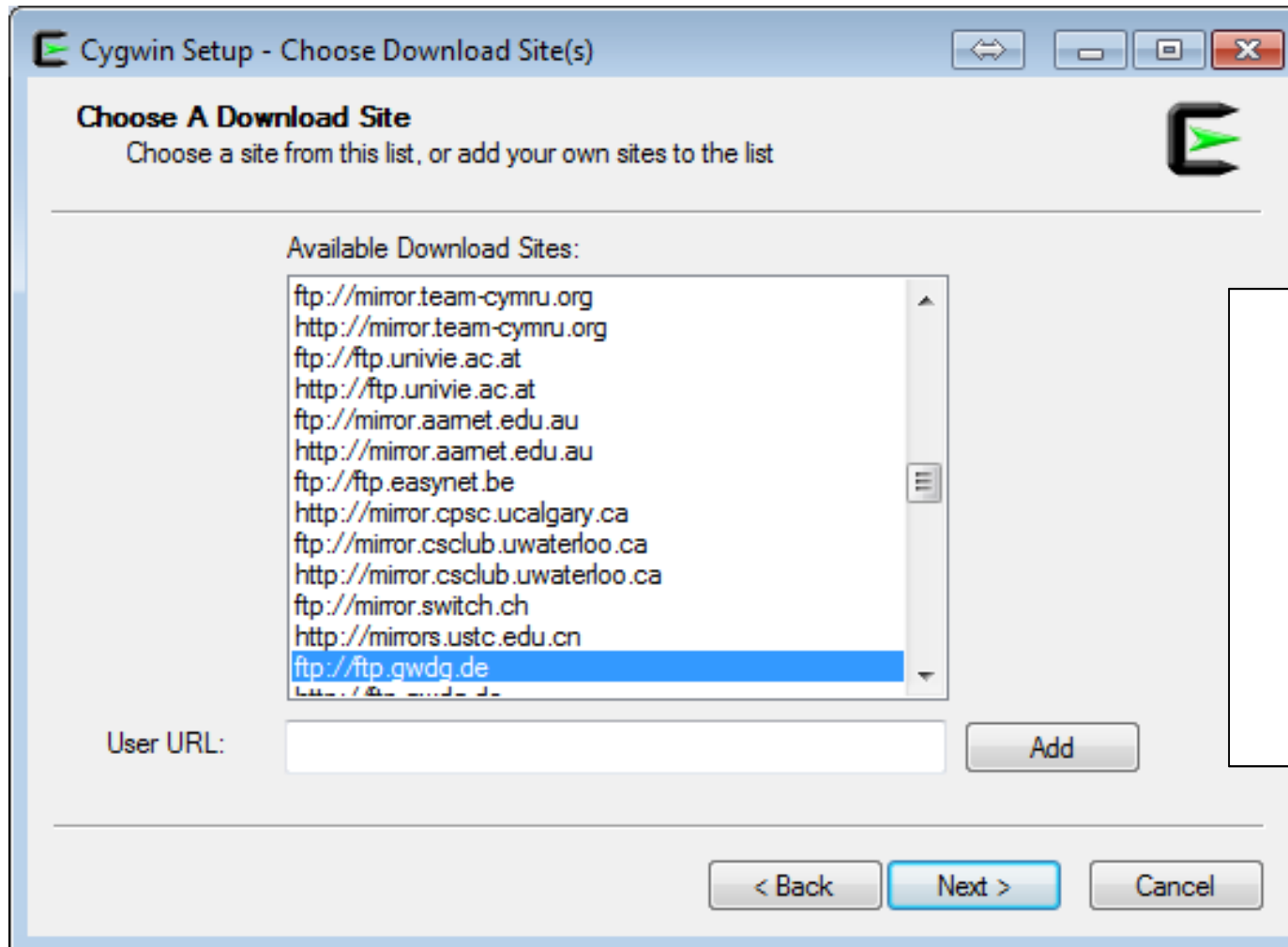
- Schritt 5: cygwin will eine Internetverbindung aufbauen



Standard-Einstellung:
„Direct Connection“
Todo:
„Next“ klicken

How to install GCC at windows

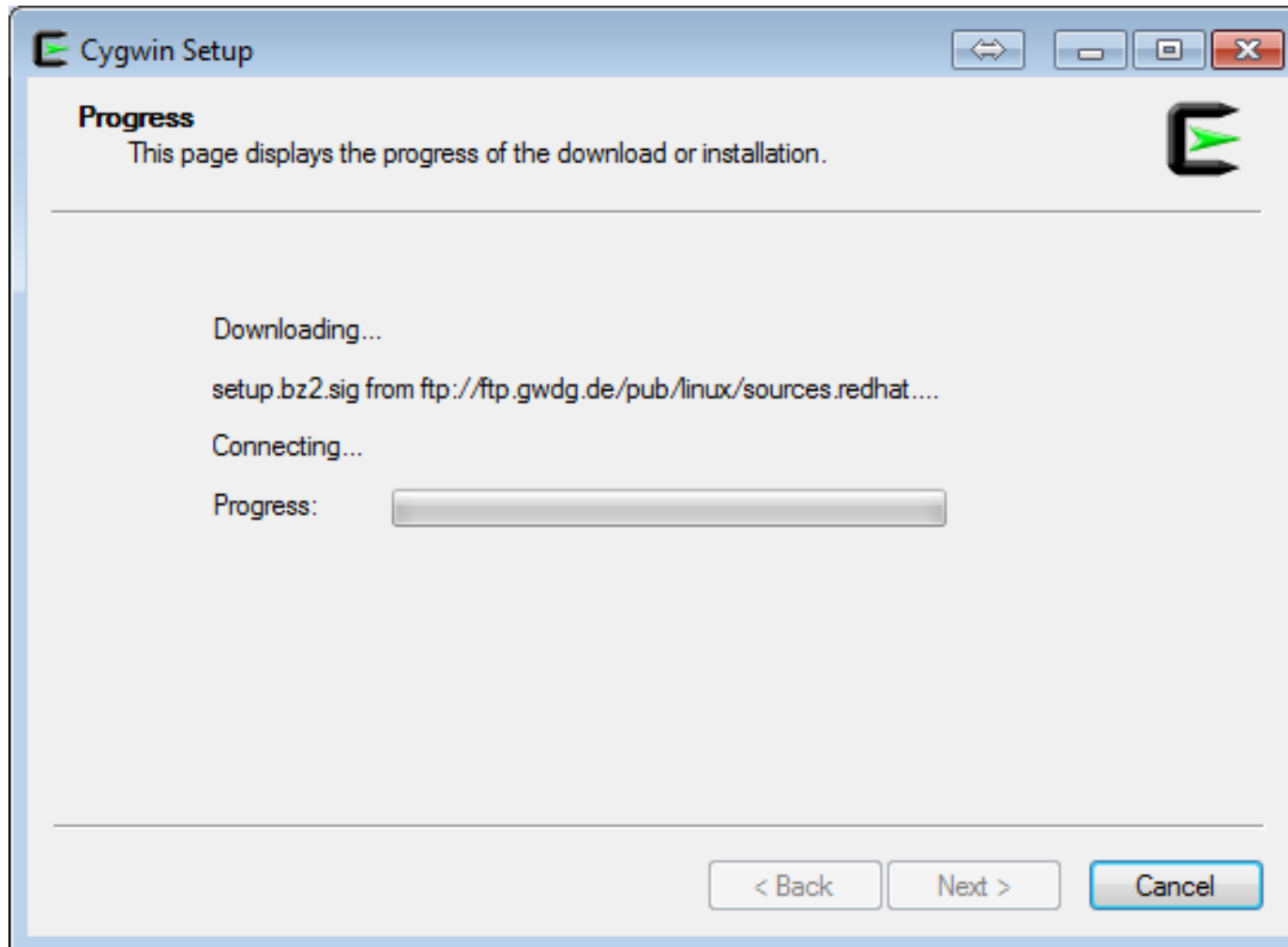
- Schritt 6: Mirror auswählen



Todo:
Suche einen Mirror
aus Deutschland
„.de“ am Ende
„Next“ klicken

How to install GCC at windows

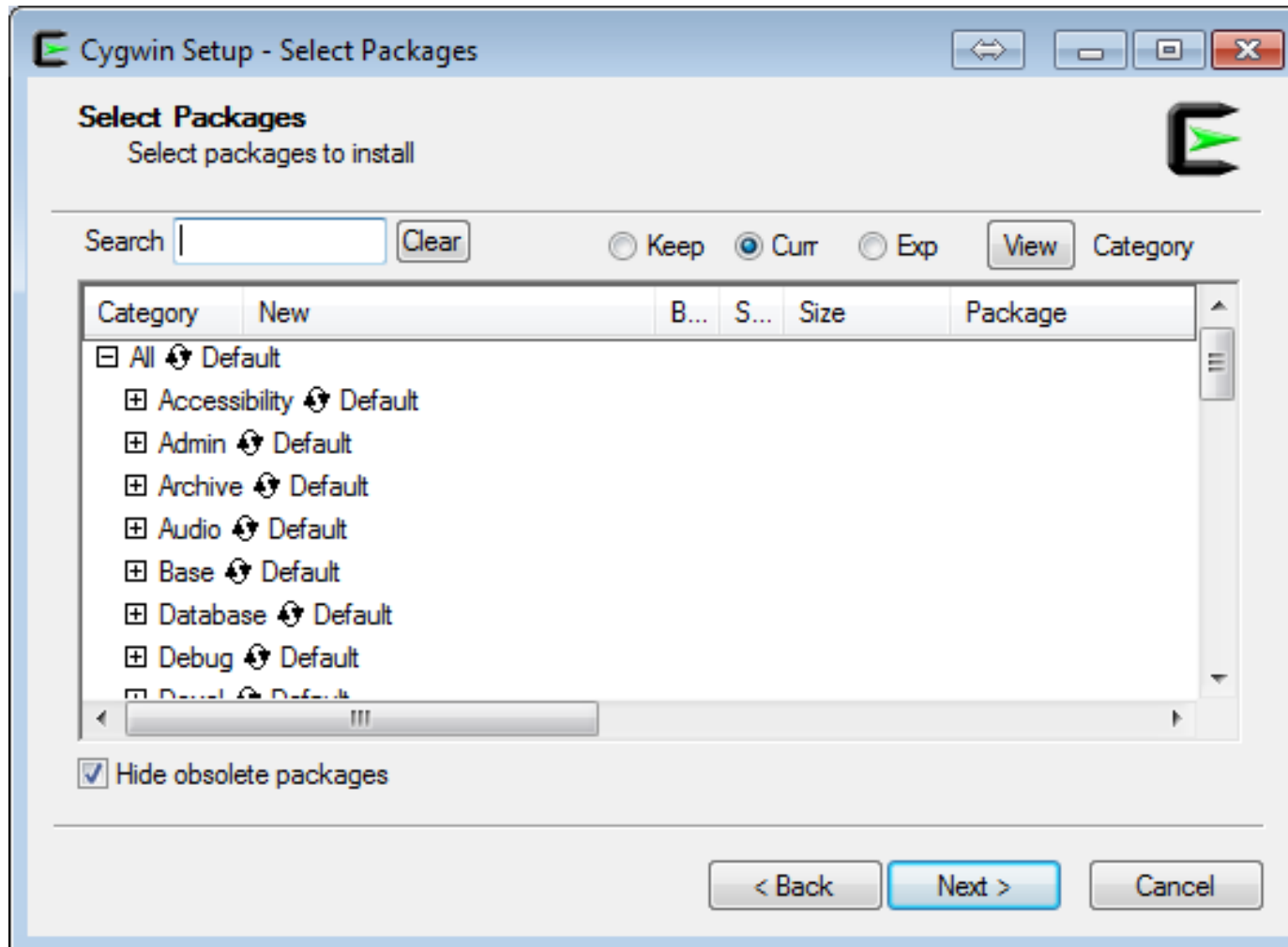
- Schritt 6.5: cygwin läd Infos aus dem Internet



Todo:
Warten

How to install GCC at windows

- Schritt 7: Packages installieren



How to install GCC at windows

- Schritt 7: Packages installieren

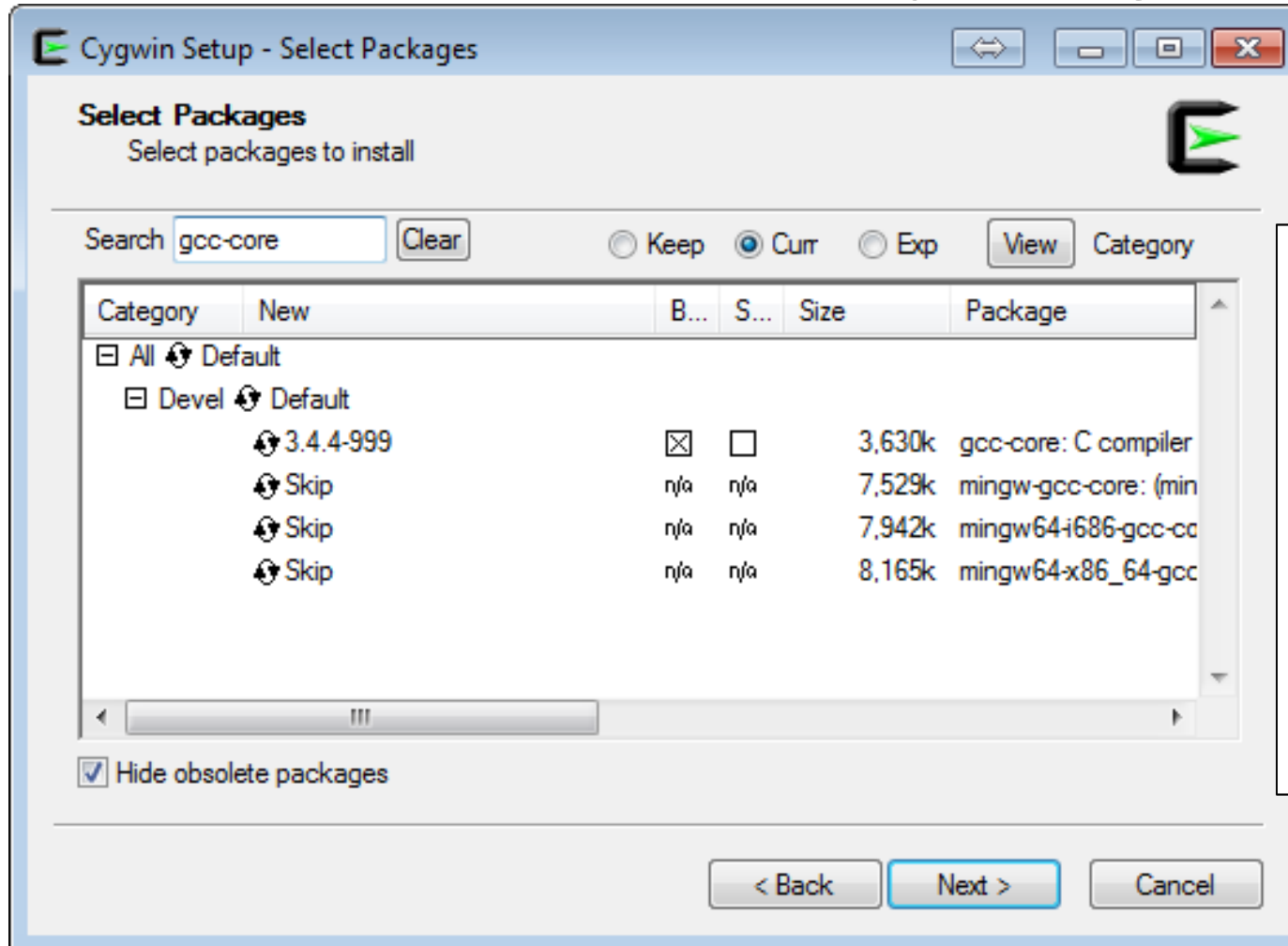
Wir brauchen hier folgende Pakete:

- ◆ gcc-core
 - ◆ gdb
 - ◆ make
 - ◆ Automake
- jeweils unter „search“ eingeben dann Spalte „Package“ angucken

How to install GCC at windows

- Schritt 7: Packages installieren

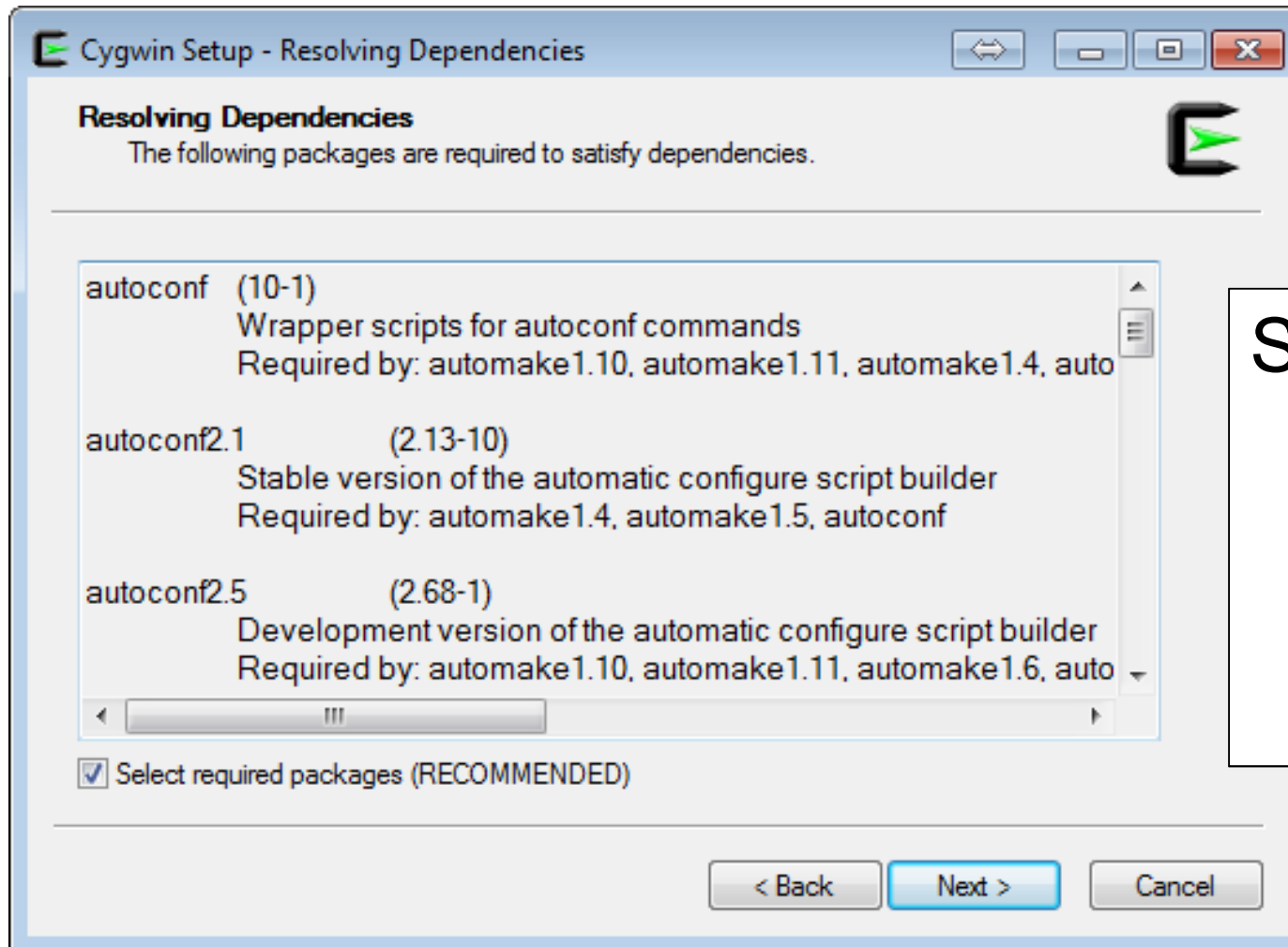
durch „Skip“ wird die jeweilige Version gewählt



Todo:
Angegebene
Packages suchen
jeweils „Skip“
klicken
„Next“ klicken

How to install GCC at windows

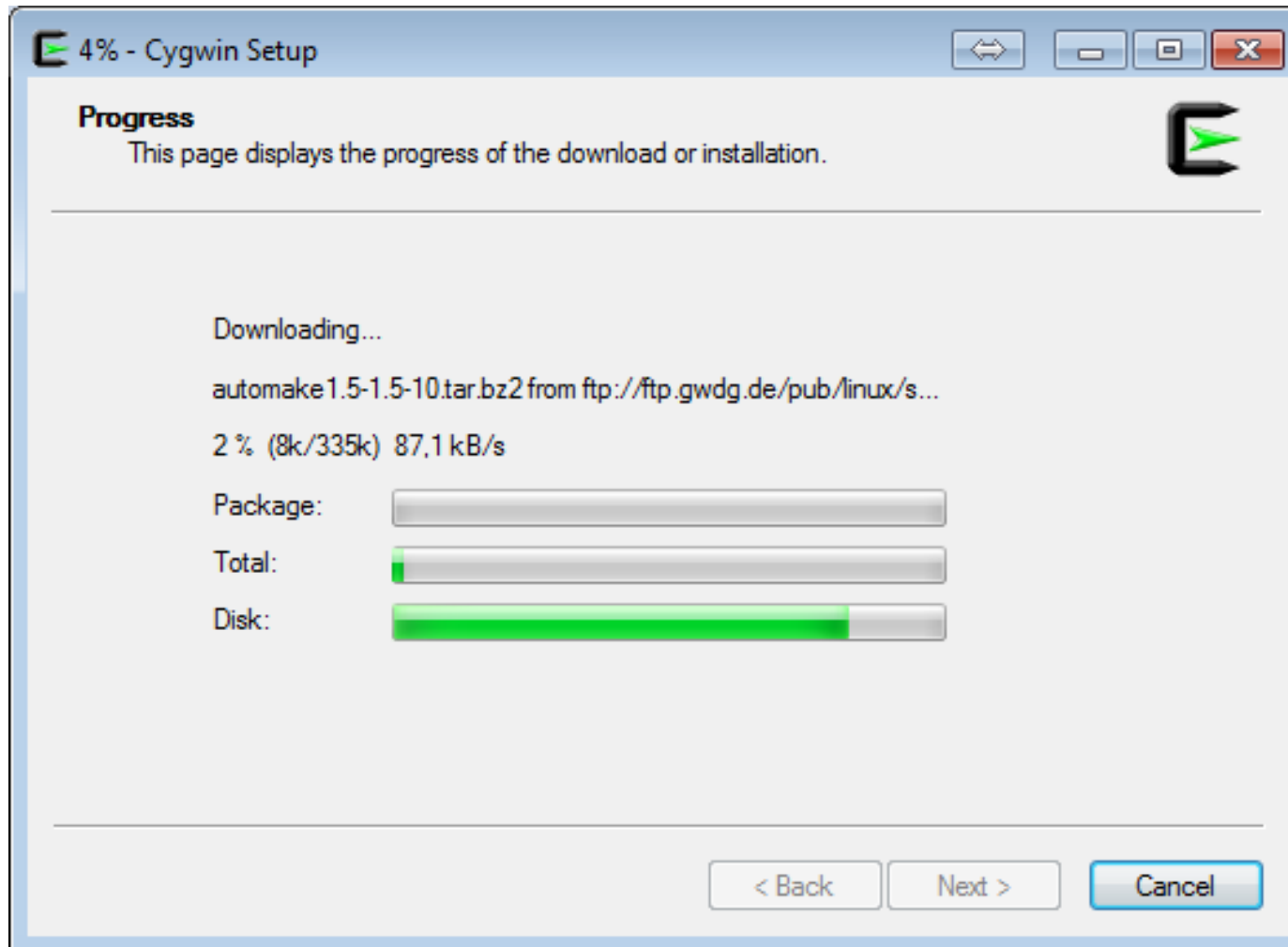
- Schritt 8: weitere Abhängigkeiten werden automatisch hinzugefügt



Standard-Einstellung:
Haken bei „Select
required packages“
Todo:
„Next“ klicken

How to install GCC at windows

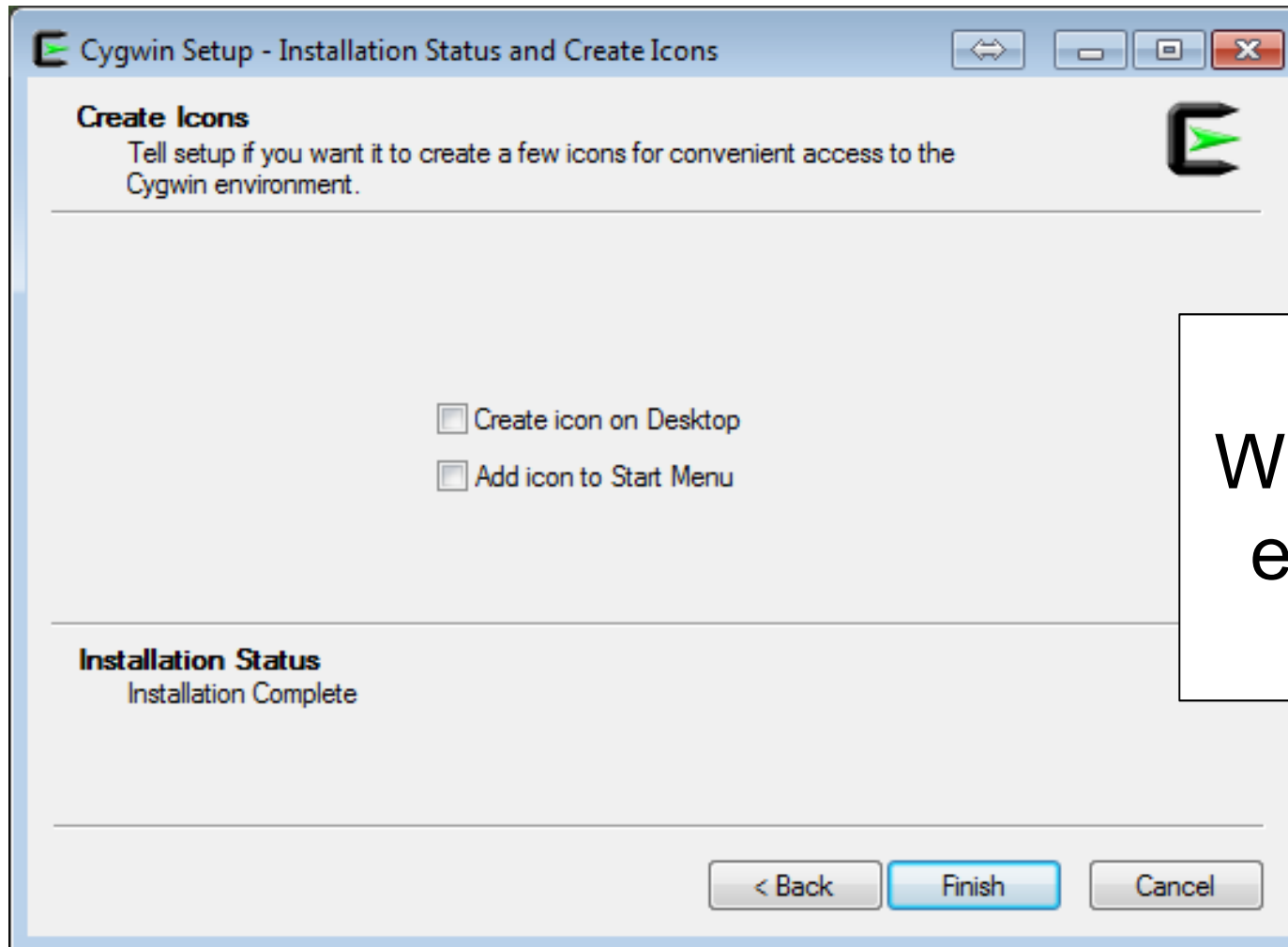
- Schritt 8.5: cygwin installiert



Todo:
Warten

How to install GCC at windows

- Schritt 9: Letzte Einstellungen



Todo:
Wählen, was davon
erwünscht ist und
„Finish“ klicken

How to install GCC at windows

- Schritt 10:

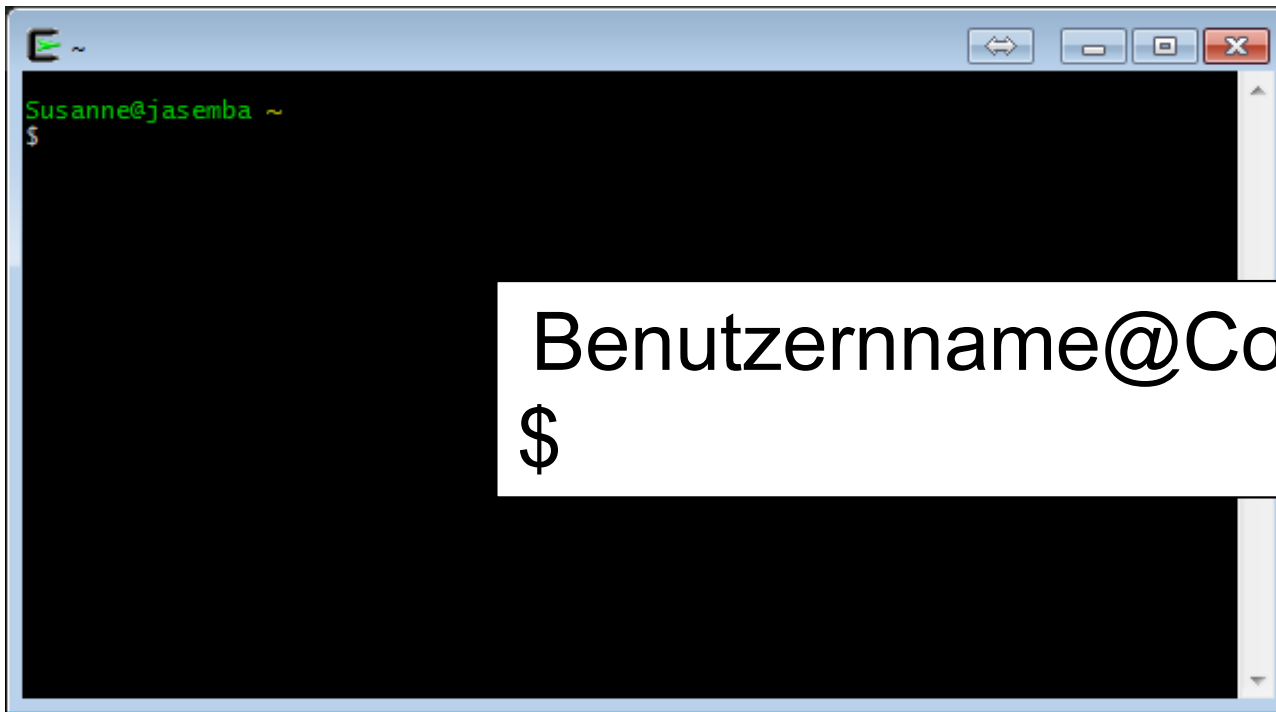
Sich gratulieren, weil man cygwin installiert hat

How to use cygwin

- Nach der Installation kann man cygwin nun starten.
- Jetzt ist der Augenblick, wo man sich ärgert, weil man kein Desktop-Item hat! Selber Schuld!

How to use cygwin

- Cygwin präsentiert sich als Console, schon direkt im richtigen Ordner.
- Hier: C:\cygwin\home\Susanne



Benutzernname@Computername ~
\$

How to use cygwin

- Hier sind ein paar Befehle für cygwin:

Befehl	Effekt
ls	Listet den Inhalt des aktuellen Verzeichnisses auf
mkdir <ordner>	Erstellt den Ordner „ <i>ordner</i> “
cd <ordner>	Wechselt in den Ordner „ <i>ordner</i> “
cp <quelle> <ziel>	Kopiert die Datei „ <i>quelle</i> “ nach „ <i>ziel</i> “
mv <quelle> <ziel>	Verschiebt die Datei „ <i>quelle</i> “ nach „ <i>ziel</i> “
rm <datei>	Löscht die Datei „ <i>datei</i> “

How to use cygwin

- Bevor wir ein Programm schreiben, wollen wir erstmal einen Ordner dafür erstellen:
- In cygwin eingeben:
mkdir ckurs
- //erstellt den gewünschten Ordner

How to use cygwin

- Nun in diesen Unterordner wechseln.
- In cygwin eingeben:
 cd ckurs
- //wechselt in den angegebenen Ordner

How to use cygwin

- Hier wollen wir jetzt die .c-Datei erstellen, in der wir unser erstes Programm schreiben:
- In cygwin eingeben
notepad helloworld.c
- /* falls die Datei noch nicht existiert, fragt Dich cygwin, ob diese Datei erstellt werden soll und öffnet sie direkt in notepad */

How to use cygwin

- Nun können wir in Notepad das gewünschte Programm schreiben:

```
#include <stdio.h>
int main() {
    printf("Hello world");
    return 0;
}
```

- Leerzeile am Ende des Programms nicht vergessen!
- Dann speichern („strg“ + „s“) und notepad schließen. Die cygwin-console ist wieder aktiv.

How to use cygwin

- Nun wollen wir unser Programm compilieren:

- In cygwin eingeben:

```
gcc -Wall -o helloworld.exe helloworld.c
```

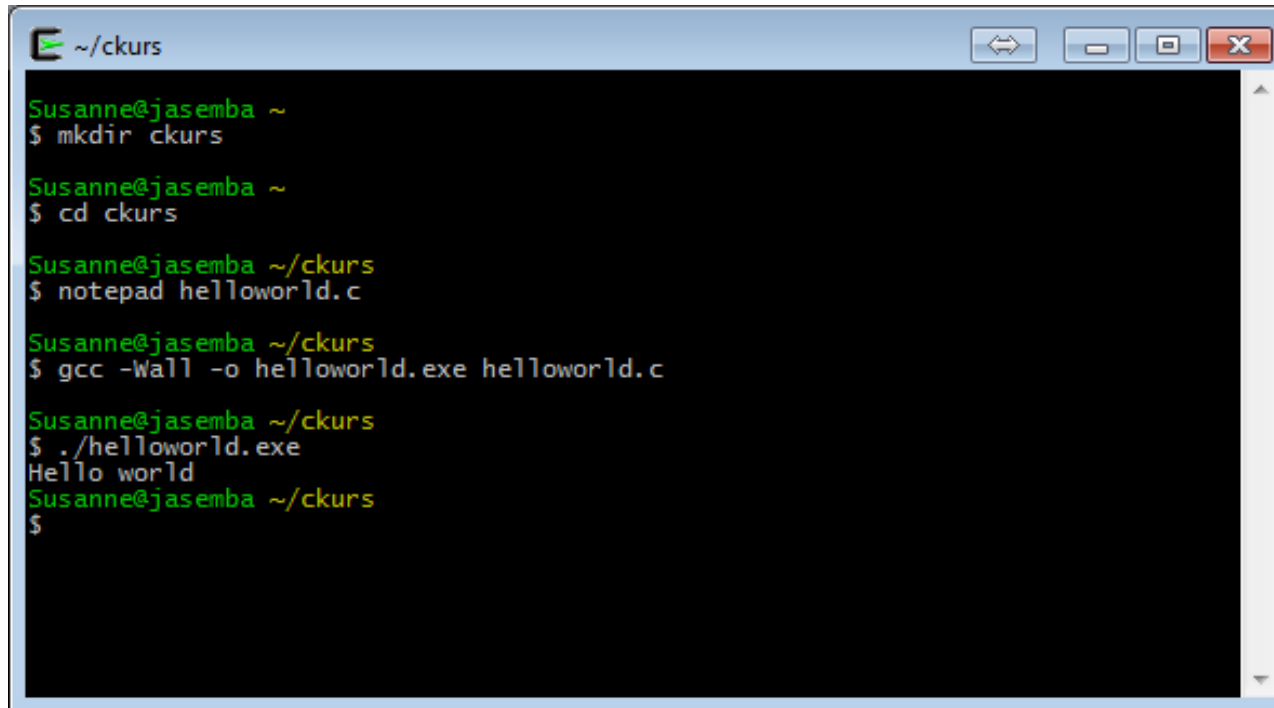
- `/* -Wall` ist die abkürzung für „warning: all“, es werden also alle Warnungen angezeigt. `helloworld.c` ist die .c-Datei, die wir eben geschrieben haben, aus dieser soll der Code entnommen werden. `helloworld.exe` gibt an, wie die ausführbare Datei heißen soll, die der Compiler herstellt. `*/`

How to use cygwin

- Und als letztes wird noch die ausführbare Datei gestartet:
- In cygwin eingeben:
./helloworld.exe
- /* “./“ gibt an, dass die executable aus dem aktuellen Ordner genommen werden soll */

How to use cygwin

- Eure cygwin-Console sollte inzwischen ungefähr folgende Einträge haben:



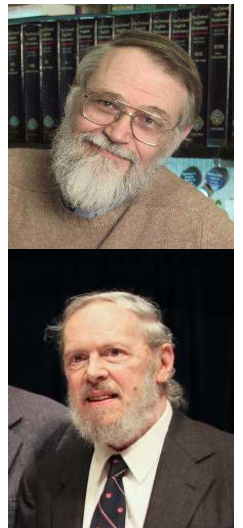
```
~/ckurs
Susanne@jasemba ~
$ mkdir ckurs
Susanne@jasemba ~
$ cd ckurs
Susanne@jasemba ~/ckurs
$ notepad helloworld.c
Susanne@jasemba ~/ckurs
$ gcc -Wall -o helloworld.exe helloworld.c
Susanne@jasemba ~/ckurs
$ ./helloworld.exe
Hello world
Susanne@jasemba ~/ckurs
$
```

Übersicht

- Warum C?
- Speicher
- Maschinencode und Compilierung
- „Hello, world“

Ein Beispiel: „hello, world“

- „Hello, world“-Programme gibt es für praktisch alle Programmiersprachen.
 - ◆ Es zeigt ein einfaches aber vollständiges, lauffähiges Programm.
 - ◆ Es gibt einen ersten Einblick in die Syntax der Sprache.
 - ◆ Erste Hürde: Code zu Lauffähigem Programm genommen!
 - ◆ Wurde zum ersten Mal in einem Buch über C verwendet:
 - ⇒ The C Programming Language
 - ⇒ Brian Kernighan, Dennis Ritchie
 - ⇒ Entwickler der Programmiersprache C



Ein Beispiel: „hello, world“

```
#include <stdio.h>
```

```
/* Hello */
```

```
int main(void)
```

```
{  
    printf("hello, world");  
    return 0;  
}
```

Livedemo: „hello, world“

- Schreiben,
- Kompilieren,
- Ausführen!