

Ergänzungen und Erläuterungen

Inhalt

- Compiler und Linker
- Makefiles
- Variablen und Zeiger



Ergänzungen und Erläuterungen

Compiler und Linker

[gcc bzw. g++]

- Compiler wie gcc und g++ übersetzen C bzw. C++ Programme, d.h. Source Code-Files, in Maschinencode. Dieser findet sich nach dem Compile-Prozess normalerweise in gleichnamigen Object-Files mit Endung *.o.
- Manchmal wird der Linker (ld) implizit mit aufgerufen und es wird direkt eine ausführbare Datei generiert. Ohne weitere Angaben heisst diese hier a.out.
- Bsp. 1: Der Befehl


```
gcc example.c
```

 generiert das Programm a.out, welches mit


```
./a.out
```

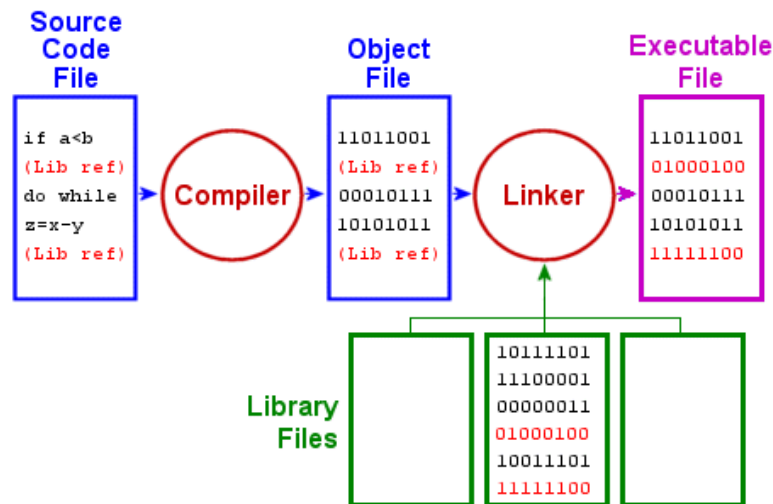
 ausgeführt wird.
- Bsp. 2 : Der Befehl


```
gcc -c example.c
```

 compiliert ohne zu Linken (wg. -c) und generiert das Object-File example.o. Diese Datei kann alleine nicht ausgeführt werden. Der Aufruf von


```
gcc example.o
```

 ruft intern den Linker auf und generiert dann wieder a.out.
- AboutDebian.Com:



- Das Object-File enthält auch schon Maschinencode, aber es sind noch nicht alle Verweise, z.B. auf Bibliotheken mit mathematischen Funktionen wie `sin`, aufgelöst. Das vollzieht der Linker und übernimmt z.B. den entsprechenden Code für `sin` aus einer Standardbibliothek mit in die ausführbare Datei.
- Es gibt 1000 Variationen vom Zusammenspiel Compiler-Linker ...

Ergänzungen und Erläuterungen

Makefiles

[]

- Das Programm `make` automatisiert den Erstellungsprozess, indem es, vereinfacht ausgedrückt, alles, was "veraltet" ist, neu generiert.
- Die Befehle und Abhängigkeiten dazu stehen in speziellen Makefiles, häufig namens `makefile`.
- Bsp. 1 für ein `makefile`: Die abhängigen Files sind durch ein Doppelpunkt getrennt, der Befehl steht darunter.

```
a.out : example.o
    gcc example.o
example.o : example.c
    gcc -c example.c
```

- Abhängigkeit bedeutet hier konkret: Wenn aus `example.c` das ausführbare Programm `a.out` erzeugt wird, so ist dieses normalerweise jünger als der Source Code, da es ja aus diesem generiert wird. Stellt man also `a.out` in Beziehung zu `example.c` (:) und ist `example.c` jünger als `a.out`, so ist an dem Source Code offenbar etwas geändert worden und `a.out` noch nicht erzeugt, sprich kompiliert worden.
Dazu gibt es im Makefile allgemeine Regeln bzw. Befehle, wie etwa aus `*.c` Files ausführbare Dateien erzeugt werden. `make` geht schliesslich alle erklärten Abhängigkeiten durch und führt diese Befehle genau dann aus, wenn abhängige Files nicht jünger sind als das, wovon sie abhängig sind.
- Es entstehen Abhängigkeitsketten, die von `make` der Reihe nach getestet und die zugehörigen Kommandos ausgeführt werden.

Ergänzungen und Erläuterungen

Variablen und Zeiger

[]

- Lokale Variablen liegen normalerweise auf dem Stack, globale häufig in einem speziellen Datenbereich.
- Folgendes Bild gibt die Situation schön wieder:
<http://www3.ntu.edu.sg/home/ehchua/programming/cpp/images/Memory-AddressContent.png>

Computer		Programmers		
Address	Content	Name	Type	Value
90000000	00	sum	int (4 bytes)	000000FF (255 ₁₀)
90000001	00			
90000002	00			
90000003	FF			
90000004	FF	age	short (2 bytes)	FFFF (-1 ₁₀)
90000005	FF			
90000006	1F	average	double (8 bytes)	1FFFFFFFFFFFFFFF (4.45015E-308 ₁₀)
90000007	FF			
90000008	FF			
90000009	FF			
9000000A	FF			
9000000B	FF			
9000000C	FF			
9000000D	FF			
9000000E	90	ptrSum	int* (4 bytes)	90000000
9000000F	00			
90000010	00			
90000011	00			

Note: All numbers in hexadecimal