

E Vom C-Quellcode zum laufenden Programm

E.1 Übersetzen - Objektmodule

■ 1. Schritt: Präprozessor

◆ entfernt Kommentare, wertet Präprozessoranweisungen aus

- fügt include-Dateien ein
- expandiert Makros
- entfernt Makro-abhängige Code-Abschnitte (*conditional code*)

Beispiel:

```
#define DEBUG
...
#ifdef DEBUG
    printf("Zwischenergebnis = %d\n", wert);
#endif DEBUG
```

◆ Zwischenergebnis kann mit `cc -P datei.c` als `datei.i` erzeugt werden
oder mit `cc -E datei.c` ausgegeben werden

E.1 Übersetzen - Objektmodule (2)

- 2. Schritt: Compilieren
 - ◆ übersetzt C-Code in Assembler
 - ◆ Zwischenergebnis kann mit `cc -S datei.c` als `datei.s` erzeugt werden

- 3. Schritt: Assemblieren
 - ◆ assembliert Assembler-Code, erzeugt Maschinencode (Objekt-Datei)
 - ◆ standardisiertes Objekt-Dateiformat: ELF (Executable and Linking Format) (vereinfachte Darstellung) - in nicht-UNIX-Systemen andere Formate
 - Maschinencode
 - Informationen über Variablen mit Lebensdauer static (ggf. Initialisierungswerte)
 - Symboltabelle: wo stehen welche globale Variablen und Funktionen
 - Relokierungsinformation: wo werden welche "nicht gefundenen" globalen Variablen bzw. Funktionen referenziert
 - ◆ Zwischenergebnis kann mit `cc -c datei.c` als `datei.o` erzeugt werden

E.2 Binden und Bibliotheken

■ 4. Schritt: Binden

- ◆ Programm **ld** : (*linker*), erzeugt ausführbare Datei (*executable file*)
 - ebenfalls ELF-Format (früher a.out-Format oder COFF)
- ◆ Objekt-Dateien (.o-Dateien) werden zusammengebunden
 - noch nicht abgesättigte Referenzen auf globale Variablen und Funktionen in anderen Objekt-Dateien werden gebunden (Relokation)
- ◆ nach fehlenden Funktionen wird in Bibliotheken gesucht
- ◆ **statisch binden**
 - alle fehlenden Funktionen werden aus Bibliotheken genommen und in die ausführbare Datei einkopiert

■ Ergebnis: ein ausführbares Programm

E.3 Laden des ausführbaren Programms

■ Windows oder Linux

◆ Erzeugen einer Umgebung zur Ausführung des Programms:

↳ Prozess

◆ dynamisch binden

- Funktionen, die von mehreren Programmen gemeinsam genutzt werden (*shared libraries, dll*), werden erst beim Start des Programms hinzugeladen

■ Mikrocontroller

◆ Übertragen des statisch gebundenen Programms in den Programmspeicher des Mikrocontrollers

↳ Flashen