

CT Übungsaufgaben

Modulare Programmierung und Linker

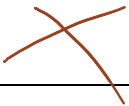


Aufgabe 1

Nennen Sie mindestens vier Punkte bei welchen die modulare Programmierung besonderen Nutzen bringt.

| Thema | Nutzen |
|----------------------------------------|--------------------------------------------------------------------------|
| Arbeiten im Team | Arbeitsteilung, effizienteres Arbeiten |
| Sinnvolle Strukturierung des Programms | bessere Wartbarkeit durch klare Strukturierung |
| Stellt Libraries zur Verfügung | bereits gut implementierter Code kann mittels Libraries abgerufen werden |
| Indiv. Verifikation pro Modul | Kapselierung pro Modul |

Aufgabe 2

Welche C Linkage haben die unterstrichenen Namen der gegebenen C Definitionen?

| Code | External Linkage | Internal Linkage | No Linkage |
|------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
| <pre>int <u>square</u>(int v) { return v * v; }</pre> |  | | |
| <pre>int square(int v) { int <u>res</u> = v * v; return res; }</pre> | | |  |
| <pre>static int <u>square</u>(int v) { return v * v; }</pre> | |  | |

Aufgabe 3

Ordnen Sie den Linker-Tasks 1 – 4 die entsprechenden Situationen a – d zu.

| Linker-Task | Situation |
|-------------------------------|---------------------------------------------------------------------------------------|
| 1) Merge code sections | a) Modul A ruft eine Funktion aus Module B auf |
| 2) Merge data sections | b) Modul A und Modul B enthalten Instruktionen |
| 3) Resolve referenced symbols | c) Modul A und Modul B enthalten globale Daten |
| 4) Relocate addresses | d) Verwendete Referenzen im Code müssen an die neue Lage der Symbole angepasst werden |

1 b

2 c

3 d

4 a

Aufgabe 4

Beim Linken werden Object Files zu einem Executable File zusammengebunden. Damit dieses Binden erfolgreich ausgeführt werden kann, muss jedes Objekt File entsprechende Informationen enthalten.

Nennen Sie die Object File Sections welche die entsprechenden Informationen enthalten.

| Object File Section Name | Enthaltene Information |
|--------------------------|-------------------------------------------------------------------------------------------------------------|
| Code Section | Instruktionen |
| Data Section | Globale Daten |
| symbol table | Liste der internen, importierten und exportierten Symbole |
| relocation table | Adressen der Instruktionen und Daten deren Zugriff auf Symbole im Laufe des Linkens angepasst werden müssen |

Aufgabe 5

Vom C Programm zum Executable: Nennen und beschreiben Sie die vier Schritte (Tools), welche bei der Übersetzung eines C Programmes in ein ausführbares Executable File notwendig sind.

| Name des Schrittes (Tools) | Was macht der Schritt? |
|----------------------------|---------------------------------------------------------------------------------------------------------------------------------------|
| Preprocessor | verarbeitet statements (#define, #include) Textprocessing: Ersetzt A gegen v. Inhalt -> Textfile mit modifiziertem C Sourcecode |
| Compiler | Übersetzt C Sourcecode in Assemblerbefehle -> Textfile mit Assemblercode |
| Assembler | Übersetzt Assemblercode in Maschinbefehle -> Binäres Objektfile |
| Linker | Fügt mehrere Objektfiles zu einem ausführbaren zusammen -> Ausführbares Objektfile (executable) |

Aufgabe 6

Folgender Ausschnitt einer Object File Symbol Table Section ist gegeben.

```
...
** Section #6 '.symtab' (SHT_SYMTAB)
# Symbol Name Value Bind Sec Type Vis Size
=====
7 a 0x00000000 Lc 4 Data De 0x4
8 b 0x00000004 Lc 4 Data De 0x4
11 main 0x00000001 Gb 1 Code Hi 0x14
12 square 0x00000000 Gb Ref Code Hi
...
```

Beantworten Sie folgende Fragen:

- 1) Welche Symbole stehen für Daten Adressen?

a, b

- 2) Welche Symbole stehen für Code Adressen?

main, square

- 3) Welche Symbole werden von diesem Modul importiert bzw. verwendet ohne dass sie in diesem Modul definiert sind?

square

- 4) Welche Symbole werden in diesem Modul definiert?

a, b, main

Aufgabe 7

Was ist eine Native-Compiler Tool Chain, was eine Cross-Compiler Tool Chain?

- 1) Native-Compiler Tool Chain:

Erzeugt Programme für die gleiche Architektur/
umgebung in welcher die Tool-chain ausgeführt wird

- 2) Cross-Compiler Tool Chain:

Erzeugt Programme für eine andere Architektur
als jene der Tool-chain