# 1. Speichersegmente

1. Tragen Sie ein „X“ in das Kästchen neben den Aussagen ein, die Ihrer Meinung nach wahr sind.  
     
     
   **Ist dies bei der Programmierung für Embedded Systemen ebenfalls gebräuchlich?**

|  |  |
| --- | --- |
|  | **Ohne das Heap-Segment und dynamische Speicherallokation sind Programme nicht lauffähig!** |
|  | **Wenn möglich wird es vermieden, da dynamische Allokationen zu Speicherfragmentierungen von ohnehin wenig Arbeitsspeicher führen können** |
|  | **In Ausnahmefällen, in denen ausreichend Arbeitsspeicher zur Verfügung stehen und/oder von einem Betriebssystem verwaltet werden.** |
|  | **Ohne das Heap-Segment ist es nicht möglich, zur Laufzeit auftretende Speicheranforderungen nachzukommen** |

1. **In welchem Speichersegment werden die lokalen Variablen abgelegt?   
   Im \_\_\_\_\_\_\_\_\_\_\_\_\_ -Segment.  
     
   Wieviel Bytes benötigen die drei Variablen insgesamt anhand ihres Datentyps?   
   \_\_\_\_\_\_\_ Bytes.**

**Wieviel Bytes reserviert der Compiler für die drei Variablen?  
\_\_\_\_\_\_\_ Bytes.**

1. Tragen Sie ein „X“ in das Kästchen neben den Aussagen ein, die Ihrer Meinung nach wahr sind.  
     
   **Welche Probleme könnten bzgl. der Speichersegmente auftreten?**

|  |  |
| --- | --- |
|  | **Die Rücksprungadresse der Funktion könnte aus Speichermangel nicht mehr gespeichert werden, welches den Programmfluss durcheinander bringt** |
|  | **Es treten keine Probleme auf, da Rekursionen besonders speicherschonend sind** |
|  | **Das Stack-Segment könnte in das Heap/BSS-Segment überlaufen, wenn viele lokale Variablen genutzt werden** |
|  | **Es passiert nichts, wenn der Stack groß genug dimensioniert ist** |

1. **In welchem Speichersegment werden die initialisierten, globalen Variablen abgelegt?   
   Im \_\_\_\_\_\_\_\_\_\_\_\_\_-Segment.  
     
   In welchem Speichersegment werden die nicht-initialisierten, globalen Variablen abgelegt?   
   Im \_\_\_\_\_\_\_\_\_\_\_\_\_-Segment.**

**Nehmen Sie weiter an, Sie haben zusätzlich folgenden Codeschnipsel gegeben:** short int allowedPortNumbers[32] = {20, 21, 22, 80, 8080};  
**In welchem Speichersegment erwarten Sie diese Variable?  
Im \_\_\_\_\_\_\_\_\_\_\_\_\_-Segment, da die Variable initialisierte und nicht-initialisierte Daten enthält.**

1. **Im \_\_\_\_\_\_\_\_\_\_\_\_\_-Segment**
2. **Erläutern Sie die Symbole aus der Symboltabelle durch die Beantwortung folgender Fragen!  
     
     
   Welche Symbole in der ausführbaren Datei Aufgabe1\_d.elf stammen aus der Quelltextdatei Aufgabe1\_d.c?**

**Warum steht in der boot\_ice.o ein Verweis auf das Symbol main mit U?**

**Warum steht in der Aufgabe1\_d.o das Symbol mit T?**  
  
  
  
**Mit welchen Assemblerbefehlen wird in der boot\_ice.s in die Funktion main() gesprungen?  
  
  
  
Nennen Sie ein Symbol, welches in der Datei boot\_ice.s referenziert wird aber in der Datei swi.S enthalten ist.**

# 2. Optimierungsstufen

1. **Optimierungsstufe = \_\_\_\_\_**
2. **Optimierungsstufe = \_\_\_\_\_**
3. **Welche Optimierungsstufe führte zur kürzesten Ausführungszeit (real)?  
   Optimierungsstufe = \_\_\_\_\_  
     
   Erläutern Sie Ihre Annahme, warum die von Ihnen ermittelte Optimierungsstufe zur kürzesten Laufzeit führte:**
4. **Wie wirkt sich die Optimierungsstufe auf Ihre lokalen Variablen aus?  
     
   Erläutern Sie warum die Variable C komplett aus dem Code entfernt wurde:**

**Warum wurde der Zugriff der globalen Variable nicht von der Optimierung verändert? Erläutern Sie kurz Ihre Annahme:   
  
  
  
  
Mit welchem Mittel/Konzept der C- und C++- Programmierung können Sie die Beeinträchtigung von Variablen durch die Optimierung verhindern? Erläutern Sie dieses kurz!**

1. **Wie wurde das Laufzeitverhalten des Programms mit der Optimierungsstufe 2 im Vergleich zur Optimierungsstufe 1 verbessert? Erläutern Sie hierzu, wie der Funktionsaufruf aus der main()-Funktion verändert wurde:  
     
     
     
     
   Was ändert sich an der main()-Funktion, wenn man die Optimierungsstufe 's' verwendet (im Vergleich zu Optimierungsstufe 2)?   
     
     
     
     
   Wenn Sie die Laufzeit des Programms unter den Optimierungsstufen 2 und 's' vergleichen, würde welche Optimierungsstufe den potentiell schnelleren Code erzeugen und warum?**

**Hinweis für PDF-Erzeugung:**

**Achten Sie beim Ausfüllen von mehrzeiligen Textfeldern darauf, dass Ihr Text nicht mehr Zeilen aufweist, als das Textfeld hoch ist! Sämtlicher Text, der durch einen Scrollbalken dargestellt werden muss, wird beim Erzeugen der PDF-Datei abgeschnitten! Sollte dieses Problem auftauchen, versuchen Sie erst die Textgröße zu verkleinern.**