

Einführung in die Programmierung

WS 2015/2016

Blatt 8

Simon Hangl, Sebastian Stabinger, Benedikt Hupfaut, Johannes Kessler

2015-12-02

- Abgabe bis spätestens Dienstag 21:59:59 über OLAT (<https://lms.uibk.ac.at/olat/dmz/>).
- Bereiten Sie jede Aufgabe so vor, dass Sie Ihre Lösung im Proseminar präsentieren können!
- Benennen Sie Ihre Abgabe nach folgendem Schema:
Gruppennummer-Nachname-blattÜbungsblattnummer.tar.gz Wenn Sie also Max Mustermann heißen und Gruppe 1 besuchen, heißt die Datei von Übung 8: *1-mustermann-blatt8.tar.gz*
- Compilieren Sie alle Programme mit den Optionen `-Wall -Werror -std=c99`

Feedback

Nutzen Sie die angebotenen Möglichkeiten, uns Feedback zu geben (eMail, Tutorium, Proseminar). Hier können Sie uns auf Probleme, notwendige Stoffwiederholungen, Unklarheiten, aber auch positive Dinge, die beibehalten werden sollten, hinweisen.

Testen und Dokumentation

Stellen Sie sicher, dass alle Lösungen fehlerfrei kompilieren. Testen Sie Ihre Lösungen ausführlich (z.B. auf falsche Eingaben, falsche Berechnungen, Sonderfälle) und dokumentieren Sie sie. Dies hilft Ihnen bei der Präsentation und uns beim Nachvollziehen Ihrer Entscheidungen. Lesen Sie die Aufgaben *vollständig* durch.

Aufgabe 1 (5 Punkte)

Gegeben ist folgender Ausschnitt des Speichers. Links steht die Adresse, rechts der Inhalt des Bytes an dieser Adresse in als Binärzahl. Gehen Sie davon aus, dass der `int`-Typ, sowie Zeiger **8 Bit** lang sind. Der `long int` hat in unserem Fall **16 Bit**.

- Die Variable `int a` liegt auf Adresse 12. Was gibt also `printf("%d", a)` aus?
- Gegeben sei die Variable `int d`. Der Befehl `printf("%d", d)` gibt 113 aus. Was wird also `printf("%p", &d)` ausgeben?
- Der Pointer `int* p` liegt auf Adresse 22. Auf welche Adresse verweist dieser? Was wird bei `printf("%p", p)` und bei `printf("%d", *p)` ausgegeben?
- Gegeben sei der Pointer `int* pc`. Der Befehl `printf("%d", *pc)` gibt 120 auf dem Bildschirm aus. Auf welche Adresse verweist der Zeiger? An welcher Adresse liegt der Zeiger?
- Wir haben eine Variable `int b`. Der Befehl `printf("%d", b)` gibt 57 aus. Was wird die Ausgabe des Befehls `printf("%p", &b)` sein?
- Der Pointer `long int* lp` liegt auf Adresse 13. Welche Länge in Bytes hat dieser Pointer. Welche Länge hat der Speicherbereich, auf den er verweist. Was wird der Befehl `printf("%ld", *lp)` ausgeben.
- Wir erzeugen eine neue Variable `int c = 64` und der Compiler legt diese auf Adresse 16. Anschließend erzeugen wir einen Zeiger `int* pc = &c`. Spielen Sie Compiler und überlegen Sie sich eine gültige Adresse für den Zeiger und füllen Sie den festgelegten Speicher mit dem korrekten Wert. Gehen Sie dabei davon aus, dass alle Speicherbereiche mit dem Wert 0, die bisher nicht verwendet wurden, frei sind.

Hinweis: Für ein besseres Verständnis notieren sie links von jeder Speicheradresse den Typ des hier gespeicherten Objekts falls er nach den obigen Angaben bekannt ist. Zeichnen Sie für jeden Zeiger einen Pfeil zu der Speicheradresse auf die er verweist.

Hinweis: Auf realen Systemen schreibt der C-Standard vor, dass Integer einen Wertebereich abdecken müssen, der über 8 Bit hinausgeht. Die Annahme, dass Integer 8 Bit lang sind wird hier wegen der besseren Lesbarkeit getroffen.

Adresse	Inhalt (Binär)
...	...
08	00000000
09	00111001
10	00000000
11	00000000
12	00000100
13	00001101
14	00000001
15	00000000
16	00000000
17	00000000
18	01111000
19	00000000
20	00100110
21	00000000
22	00011011
23	00000000
24	00000000
25	00000000
26	00000000
27	00010010
28	00000000
29	01110001
30	00000000
...	...

Hinweis: Abgabe: 1-mustermann-a1.txt

Aufgabe 2 (5 Punkte)

Betrachten Sie das Programm in der Datei `pointer.c` und beantworten Sie die folgenden Fragen. Bitte beachten Sie, dass der Code dieser Aufgabe fehlerhaft ist und ein Negativbeispiel darstellt. Damit das Programm compiliert müssen Sie das Compilerflag `-Werror` weglassen.

- Warum sind die Variablen `one` und `two` nach einem Aufruf der Funktion `swap_arguments` nicht vertauscht? (Siehe Ausgabe Zeile 1+2)
- Erklären Sie Schritt für Schritt die Funktion `swap_pointer`. Wie funktioniert der Variablentausch? (Siehe Ausgabe Zeile 1+3)
- Warum ergibt die Ausgabe in Zeile 4 ungleich, obwohl die beiden Pointer ursprünglich auf die selbe Speicheradresse zeigen und um eins erhöht werden? Ist dies immer der Fall, oder gibt es eine Situation in der dieser Test 'gleich' ergibt?
- Wie erklären Sie sich die Ausgabe in Zeile 5? Was passiert hier? Ist die Ausgabe auf allen Systemen gleich?
- Welche Ausgabe erhalten Sie in Zeile 6? Warum entspricht dieser Wert nicht dem erwarteten Wert 17? Wie können Sie diesen Wert beeinflussen?

Hinweis: Abgabe: 1-mustermann-a2.txt