

Grundlagen - Betriebssysteme und Systemsoftware

IN0009, WiSe 2023/24

Übungsblatt 2

30. Oktober 2023 – 3. November 2023

Hinweis: Mit * gekennzeichnete Teilaufgaben sind ohne Lösung vorhergehender Teilaufgaben lösbar.

Aufgabe 1 Operatorpräzedenz

In C gelten die folgenden Präzedenzregeln (nicht vollständig):

[] (Array) und () (Funktion) werden von links nach rechts abgearbeitet und haben Vorrang vor * (Pointer/Dereferenzierung) und & (Adresse), welche von rechts nach links abgearbeitet werden.

Bestimmen Sie davon ausgehend, wie die folgenden Ausdrücke gelesen werden:

- a)* long **foo[7];
- b)* unsigned long int **x();
- c)* char *(*(**foo [][][8])())[];
- d)* int (*(*foo)(void))[];

Aufgabe 2 Hexdump

Es sei folgender Hexdump der Speicheradressen 0x00000000-0x000000FF gegeben:

```
0x0000 89 50 4e 47 0d 0a 1a 0a ff 00 00 00 49 48 44 52
0x0010 00 00 05 00 00 00 02 82 08 06 00 00 00 8e 3b 74
0x0020 aa 00 00 00 a7 73 42 49 54 08 08 08 08 7c 08 64
0x0030 47 42 53 00 88 00 00 00 4d 00 00 0f 61 00 00 0f
0x0040 61 01 a8 3f a7 69 00 00 00 38 74 45 58 74 53 6f
0x0050 66 74 77 61 72 65 00 6d 61 74 70 6c 6f 74 6c 69
0x0060 f1 20 f5 65 72 73 69 6f 6e 33 2e 31 2e 31 2c 20
0x0070 68 74 74 70 3a 2f 73 74 72 69 6e 67 00 74 6c 69
0x0080 62 2e 6f 72 67 2f 10 66 17 19 00 00 20 00 49 44
0x0090 41 54 78 9c ec dd 77 9c 54 e5 a1 ff f1 cf 6c 85
0x00a0 5d 76 97 be 85 de 62 69 6e 67 6f 00 a4 05 93 dc
0x00b0 18 63 bb 37 a2 49 ae 29 e6 17 d4 28 22 45 16 05
0x00c0 c4 44 62 12 4d 31 d7 80 b1 5f 5b 12 51 41 20 8a
0x00d0 08 00 00 00 39 05 00 00 b2 d4 2d 2c db cf ef 8f
0x00e0 49 f6 c6 58 00 dd e5 ec ce 7e de af d7 bc 5e 3c
0x00f0 87 39 33 df 59 75 9d f9 ce 73 9e 27 12 04 41 80
```

Wir befinden uns auf einer 32-bit little-endian Architektur, eine Speicheradresse sei somit 32 bit breit. Ein Integer sei 32 bit breit.

Das folgende Programm basiert darauf, dass Pointer und Integer gleich breit sind. Seien Sie sich bewusst, dass dies in der Realität nicht unbedingt der Fall ist!

Für die Umrechnung von hex zu ascii nehmen Sie `man 7 ascii` zur Hilfe.

- a)* Wie viele Hex-Zeichen umfasst eine Speicheradresse im obigen Hexdump?
- b) Nehmen wir an, an der Adresse 0x8c ist ein Pointer gespeichert. Wie lautet die Adresse, auf die der Pointer zeigt? Beachten Sie die Endianness der Architektur!
- c) Bestimmen Sie die Ausgaben des folgenden Programms, ausgeführt im Kontext des obigen Speicherausschnitts:

```
1 char *x = (char*) 0x30;
2 int* i = (int*) 0xd0;
3
4 printf("Some_string:_%s\n", x);
5 printf("Some_other_string:_%s\n", x+0x46);
6
7 int a = i[1];
8 int b = *(int*)i;
9 printf("a:_%d, _b:_%d\n", a, b);
```

Aufgabe 3 Sichere Programmierung und typische Fehler

Erklären Sie wieso jeder der folgenden Programmschnipsel auf *keinen* Fall verwendet werden sollte. Gehen Sie dabei auf Programmierfehler oder schlechte Programmierpraxis ein. Geben Sie auch jeweils eine Lösung an.

a)*

```
1 char userinput[256];  
2 gets(userinput);
```

b)*

```
1 char userinput[256] = {0};  
2 int ret = scanf("%256s", userinput);
```

c)*

```
1 #define MUL(x,y) x*y  
2 int y = MUL(4+1, 3+6);
```

d)*

```
1 int *p1, p2;
```

e)*

```
1 int *p = malloc(sizeof *p);  
2 scanf("%d", p);  
3 free(p);  
4 printf("*p_is_%d", *p);
```

f)*

```
1 int* list;  
2 if (list == NULL) {  
3     list = malloc(LIST_SIZE);  
4 }
```

Aufgabe 4 Zusätzliche Übungsaufgaben (Optional)

Hier finden Sie einige Aufgaben zum Thema **C-Programmierung** und **Hexdumps**, falls Sie noch mehr Übungsbedarf haben.

C-Programmierung:

- Endterm 2018: Aufgabe 3
- Endterm 2017: Aufgabe 2
- Retake 2019: Aufgabe 1
- Retake 2019: Aufgabe 2
- Retake 2018: Aufgabe 2
- Retake 2017: Aufgabe 2

Hexdump:

- Endterm 2020: Aufgabe 1
- Retake 2019: Aufgabe 6