Exercises in System Level Programming (SLP) – Summer Term 2024

Exercise 4

Maximilian Ott
Lehrstuhl für Informatik 4
Friedrich-Alexander-Universität Erlangen-Nürnberg

Presentation Assignment 2
In depth: Pointers

- Variable: `uint8_t x`
- Pointer: `uint8_t *y`
- Address-of operator: `&x`
- Indirection operator: `*y`

```c
uint8_t a = 23;
uint8_t b = 42;
uint8_t *p = &a;
*p = 66;
p = &b;
*p -= 40;
uint8_t c = *p;
```
Caution: The exact placement of the variable on the stack depends on the compiler and the chosen optimization level!

Caution: The exact placement of the variable on the stack depends on the compiler and the chosen optimization level!
In depth: Pointers

- Variable: uint8_t x
- Pointer: uint8_t *y
- Address-of operator: &x
- Indirection operator: *y

01 uint8_t a = 23;
02 uint8_t b = 42;
03 uint8_t * p = &a;
04 *p = 66;
05 p = &b;
06 *p -= 40;
07 uint8_t c = *p;

Caution: ATmega328PB has 8-bit registers and 16-bit addresses
In depth: Pointers

- Variable: `uint8_t x`
- Pointer: `uint8_t *y`
- Address-of operator: `&x`
- Indirection operator: `*y`

01 uint8_t a = 23;
02 uint8_t b = 42;
03 uint8_t *p = &a;
04 *p = 66;
05 p = &b;
06 *p -= 40;
07 uint8_t c = *p;

Caution: ATmega328PB has 8-bit registers and 16-bit addresses
In depth: Pointers

- Variable: uint8_t x
- Pointer: uint8_t *y
- Address-of operator: &x
- Indirection operator: *y

```c
uint8_t a = 23;
uint8_t b = 42;
uint8_t *p = &a;
*p = 66;
p = &b;
*p -= 40;
uint8_t c = *p;
```

Stack:
```
<table>
<thead>
<tr>
<th>Value</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>66</td>
<td>0x0910</td>
</tr>
<tr>
<td>2</td>
<td>0x090f</td>
</tr>
<tr>
<td>0x090f</td>
<td>0x090e</td>
</tr>
<tr>
<td>2</td>
<td>0x090c</td>
</tr>
<tr>
<td>2</td>
<td>0x090b</td>
</tr>
</tbody>
</table>
```

Caution: ATmega328PB has 8-bit registers and 16-bit addresses

In depth: Arrays

- Constant pointer: uint8_t a[
- Variable pointer: uint8_t *b
- Current element: *b
- x-th element: b[x]
- x-th element: *(b+x)

```c
uint8_t x[] = {2,4,8,16};
uint8_t *y = x;
uint8_t z = x[1];
z = *y;
y = y+2;
z = *y;
z = x[7];
```
In depth: Arrays

- Constant pointer: `uint8_t a[]`
- Variable pointer: `uint8_t *b`
- Current element: `*b`
- x-th element: `b[x]`
- x-th element: `*(b+x)`

```
08 uint8_t x[] = {2,4,8,16};
09 uint8_t *y = x;
10 uint8_t z = x[1];
11 z = *y;
12 y = y+2;
13 z = *y;
14 z = x[7];
```

Stack:

<p>| | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>x[3]</td>
<td>16</td>
<td>0x0909</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>x[2]</td>
<td>8</td>
<td>0x0908</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>x[1]</td>
<td>4</td>
<td>0x0907</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>x[0]</td>
<td>2</td>
<td>0x0906</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
In depth: Arrays

- Constant pointer: uint8_t a[]
- Variable pointer: uint8_t *b
- Current element: *b
- x-th element: b[x]
- x-th element: *(b+x)

```c
08 uint8_t x[] = {2,4,8,16};
09 uint8_t *y = x;
10 uint8_t z = x[1];
11 z = *y;
12 y = y+2;
13 z = *y;
14 z = x[7];
```

```
  .                 Stack ↓
  .                   0x09090a
x[3]       16  0x090909
x[2]       8  0x090908
x[1]       4  0x090907
x[0]       2  0x090906
|       y  0x090906  0x090905
|     y  0x090906  0x090904
|    z      4  0x090903
|                         0x090902
  .
```

In depth: Arrays

- Constant pointer: uint8_t a[]
- Variable pointer: uint8_t *b
- Current element: *b
- x-th element: b[x]
- x-th element: *(b+x)

```c
08 uint8_t x[] = {2,4,8,16};
09 uint8_t *y = x;
10 uint8_t z = x[1];
11 z = *y;
12 y = y+2;
13 z = *y;
14 z = x[7];
```

```
  .                 Stack ↓
  .                   0x09090a
x[3]       16  0x090909
x[2]       8  0x090908
x[1]       4  0x090907
x[0]       2  0x090906
|       y  0x090906  0x090905
|     y  0x090906  0x090904
|    z      4  0x090903
|                         0x090902
  .
```
Constant pointer: uint8_t a[]
Variable pointer: uint8_t *b
Current element: *b
x-th element: b[x]
x-th element: *(b+x)

uint8_t x[] = {2,4,8,16};
uint8_t *y = x;
uint8_t z = x[1];
y = y+2;
z = *y;
z = x[7];
In depth: Arrays

- Constant pointer: `uint8_t a[]`
- Variable pointer: `uint8_t *b`
- Current element: `*b`
- x-th element: `b[x]`
- x-th element: `*(b+x)`

```
08 uint8_t x[] = {2,4,8,16};
09 uint8_t *y = x;
10 uint8_t z = x[1];
11 z = *y;
12 y = y+2;
13 z = *y;
14 z = x[7]; // ***
```

Stack ↓

<table>
<thead>
<tr>
<th>Stack</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>⋮</td>
<td></td>
</tr>
<tr>
<td>x[3]</td>
<td>16</td>
</tr>
<tr>
<td>x[2]</td>
<td>8</td>
</tr>
<tr>
<td>x[1]</td>
<td>4</td>
</tr>
<tr>
<td>x[0]</td>
<td>2</td>
</tr>
<tr>
<td>y</td>
<td>0x0908</td>
</tr>
<tr>
<td>y</td>
<td>0x0908</td>
</tr>
<tr>
<td>z</td>
<td>0x0903</td>
</tr>
<tr>
<td></td>
<td>0x0902</td>
</tr>
</tbody>
</table>

Hands-on: Pointers

No Screencast
- Call-by-value vs. call-by-reference
- Pointer and arrays
- Pointer arithmetic
- `struct` for GPS coordinates
- Array of GPS coordinates
- Function pointers

Can be compiled for the SPICboard (serial console), the SPICSim or Linux

Source code:
https://sys.cs.fau.de/extern/lehre/ss24/slp/uebung/material/pointer.c