

Exercises in System Level Programming (SLP) – Summer Term 2025

Exercise 4

Maxim Ritter von Onciul
Eva Dengler

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



Lehrstuhl für Informatik 4
Systemsoftware



Friedrich-Alexander-Universität
Faculty of Engineering

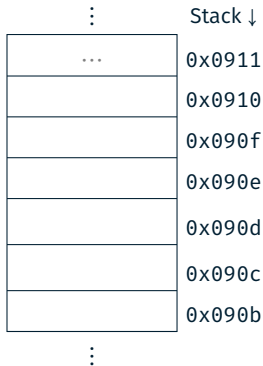
Presentation Assignment 2

Pointers & Arrays



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

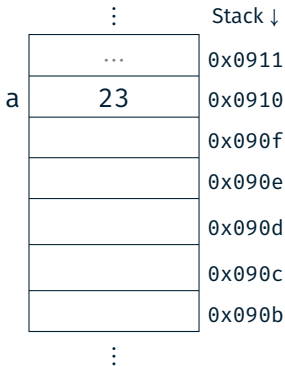
```
01 uint8_t a = 23;  
02 uint8_t b = 42;  
03 uint8_t * p = &a;  
04 *p = 66;  
05 p = &b;  
06 *p -= 40;  
07 uint8_t c = *p;
```





- 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;
```

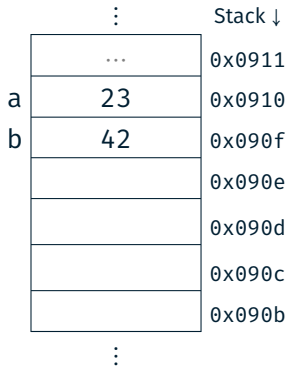


Caution: The exact placement of the variable on the stack depends on the compiler and the chosen optimization level!



- 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;
```

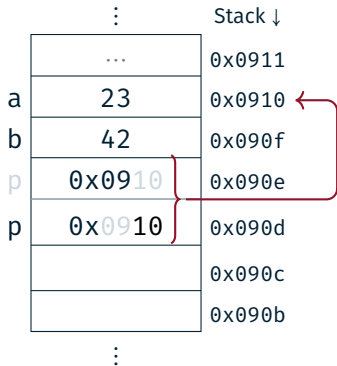


Caution: The exact placement of the variable on the stack depends on the compiler and the chosen optimization level!



- 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;
```

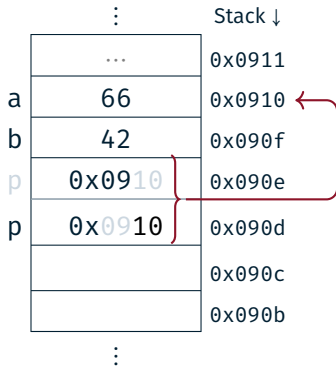


Caution: ATmega328PB has 8-bit registers and 16-bit addresses



- 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;
```

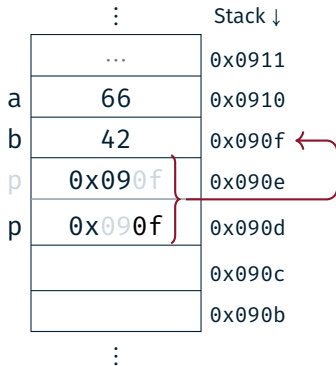


Caution: ATmega328PB has 8-bit registers and 16-bit addresses



- 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;
```

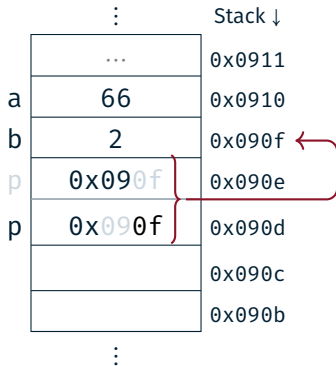


Caution: ATmega328PB has 8-bit registers and 16-bit addresses



- 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;
```

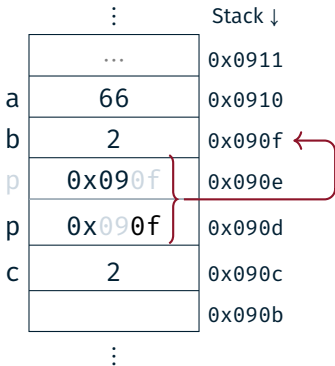


Caution: ATmega328PB has 8-bit registers and 16-bit addresses



- 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;
```

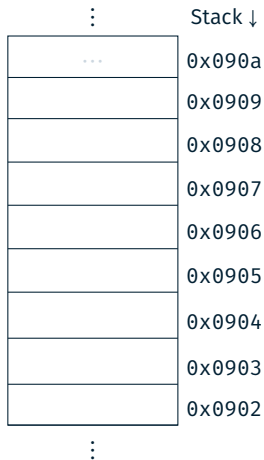


Caution: ATmega328PB has 8-bit registers and 16-bit addresses



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

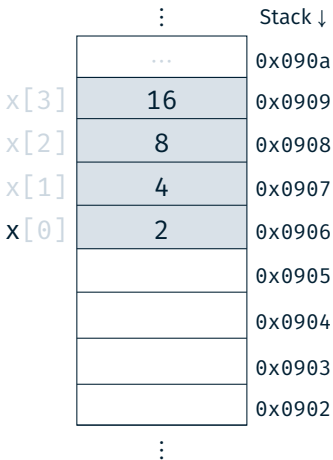
```
08 uint8_t x[] = {2,4,8,16};
09 uint8_t *y = x;
10 uint8_t z = x[1];
11 z = *y;
12 y = y+2;
13 z = *y;
14 z = x[7];
```





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

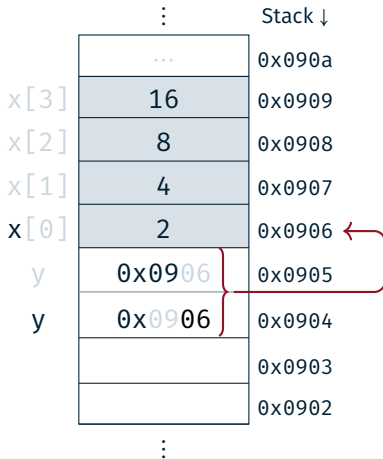
```
08 uint8_t x[] = {2,4,8,16};
09 uint8_t *y = x;
10 uint8_t z = x[1];
11 z = *y;
12 y = y+2;
13 z = *y;
14 z = x[7];
```





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

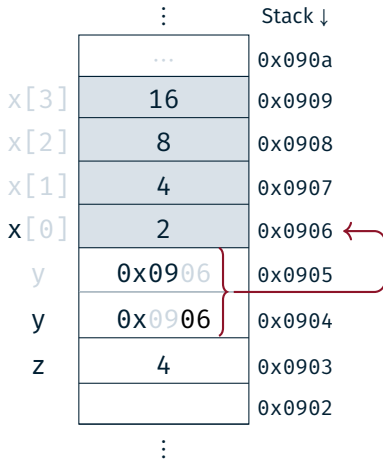
```
08 uint8_t x[] = {2,4,8,16};  
09 uint8_t *y = x;  
10 uint8_t z = x[1];  
11 z = *y;  
12 y = y+2;  
13 z = *y;  
14 z = x[7];
```





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

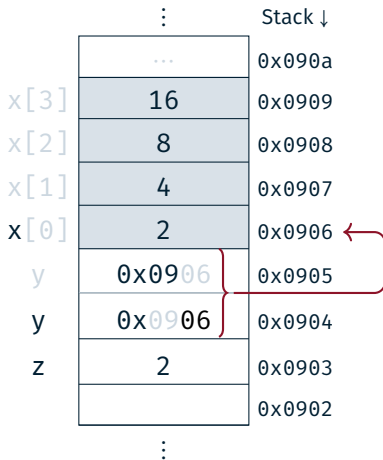
```
08 uint8_t x[] = {2,4,8,16};  
09 uint8_t *y = x;  
10 uint8_t z = x[1];  
11 z = *y;  
12 y = y+2;  
13 z = *y;  
14 z = x[7];
```





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

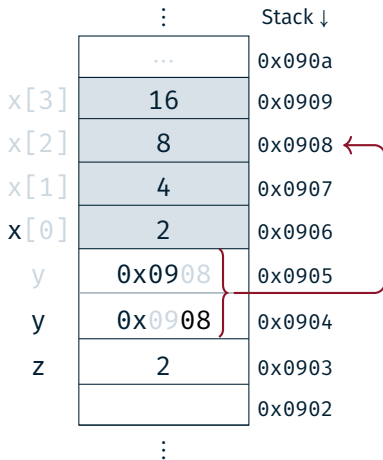
```
08 uint8_t x[] = {2,4,8,16};
09 uint8_t *y = x;
10 uint8_t z = x[1];
11 z = *y;
12 y = y+2;
13 z = *y;
14 z = x[7];
```





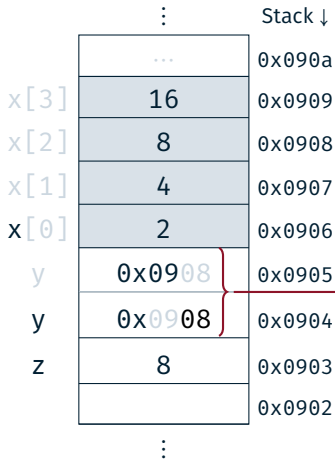
- Constant pointer: `uint8_t a[]`
- Variable pointer: `uint8_t *b`
- Current element: `*b`
- x-th element: `b[x]`
- x-th element: `*(b+x)`

```
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];
```



- 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];
```

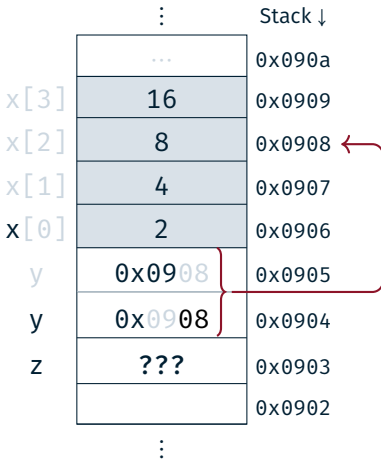


- Constant pointer: `uint8_t a[]`
- Variable pointer: `uint8_t *b`
- Current element: `*b`
- x-th element: `b[x]`
- x-th element: `*(b+x)`

```
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]; // ???
```

Undefined behaviour!

Access to an array outside of the defined area



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/ss25/slp/uebung/material/pointer.c>