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

Exercise 4

Maximilian Ott

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

Lehrstuhl für Verteilte Systeme
und Betriebssysteme

FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG
TECHNISCHE FAKULTÄT

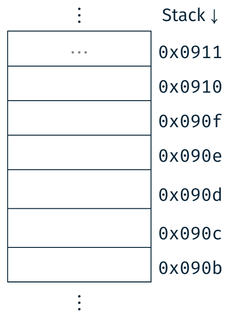
Presentation Assignment 2

Pointers & Arrays

In depth: Pointers

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

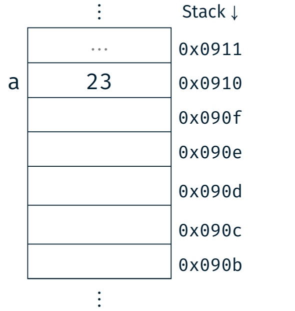
```
uint8_t a = 23;
uint8_t b = 42;
uint8_t * p = &a;
*p = 66;
p = &b;
*p -= 40;
uint8_t c = *p;
```



In depth: Pointers

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

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



Stack ↓

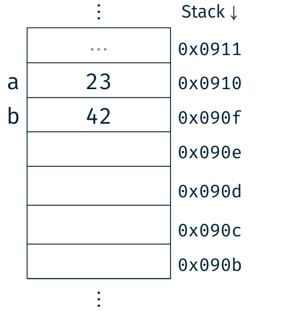
	⋮	
	…	0x0911
a	23	0x0910
		0x090f
		0x090e
		0x090d
		0x090c
		0x090b
	⋮	

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

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



Stack ↓

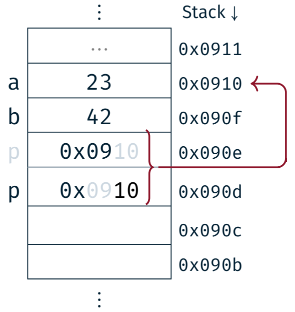
	⋮	
	…	0x0911
a	23	0x0910
b	42	0x090f
		0x090e
		0x090d
		0x090c
		0x090b
	⋮	

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

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



Stack ↓

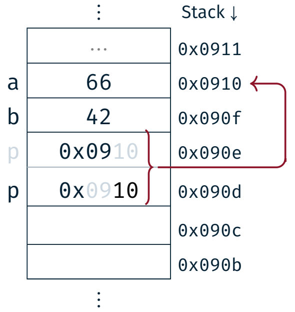
	⋮	
	…	0x0911
a	23	0x0910
b	42	0x090f
p	0x0910	0x090e
p	0x0910	0x090d
		0x090c
		0x090b
	⋮	

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

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



Stack ↓

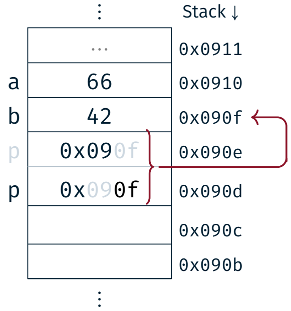
	⋮	
	…	0x0911
a	66	0x0910
b	42	0x090f
p	0x0910	0x090e
p	0x0910	0x090d
		0x090c
		0x090b
	⋮	

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

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

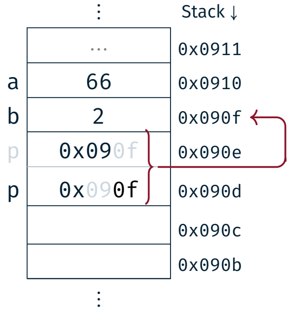


Caution: ATmega328PB has 8-bit registers and 16-bit addresses

In depth: Pointers

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

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

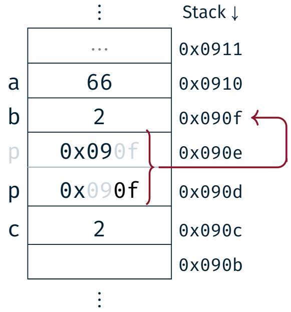


Caution: ATmega328PB has 8-bit registers and 16-bit addresses

In depth: Pointers

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

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

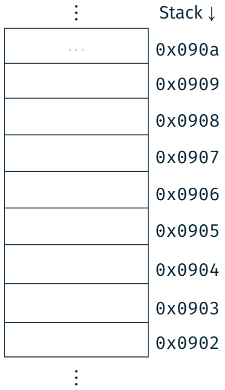


Caution: ATmega328PB has 8-bit registers and 16-bit addresses

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)

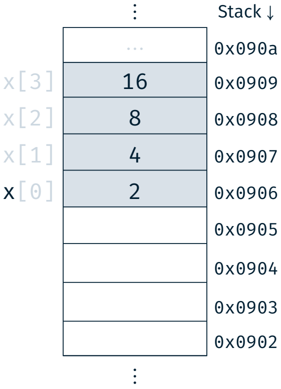
```
uint8_t x[] = {2,4,8,16};
uint8_t *y = x;
uint8_t z = x[1];
z = *y;
y = y+2;
z = *y;
z = x[7];
```



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)

```
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];
```

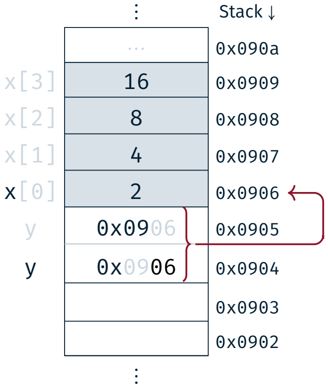


	⋮	Stack ↓
	...	0x090a
x[3]	16	0x0909
x[2]	8	0x0908
x[1]	4	0x0907
x[0]	2	0x0906
		0x0905
		0x0904
		0x0903
		0x0902
	⋮	

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)

```
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];
```

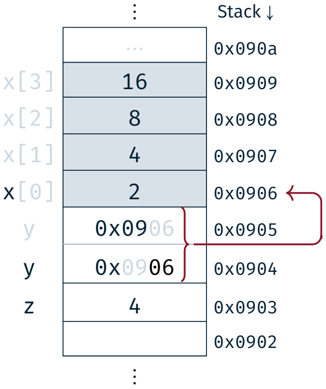


	⋮	Stack ↓
	...	0x090a
x[3]	16	0x0909
x[2]	8	0x0908
x[1]	4	0x0907
x[0]	2	0x0906
y	0x0906	0x0905
y	0x0906	0x0904
		0x0903
		0x0902
	⋮	

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)

```
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];
```

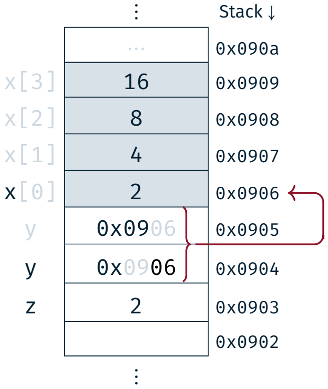


	⋮	Stack ↓
	...	0x090a
x[3]	16	0x0909
x[2]	8	0x0908
x[1]	4	0x0907
x[0]	2	0x0906
y	0x0906	0x0905
y	0x0906	0x0904
z	4	0x0903
		0x0902
	⋮	

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)

```
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];
```

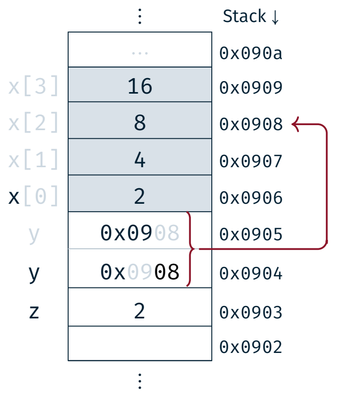


	⋮	Stack ↓
	...	0x090a
x[3]	16	0x0909
x[2]	8	0x0908
x[1]	4	0x0907
x[0]	2	0x0906
y	0x0906	0x0905
y	0x0906	0x0904
z	2	0x0903
		0x0902
	⋮	

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)

```
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];
```

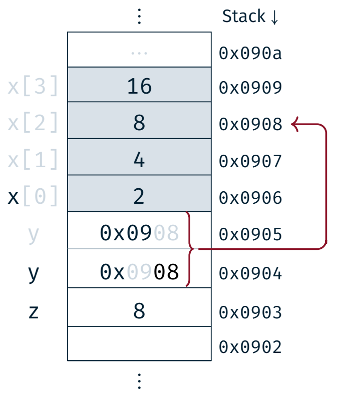


	Stack ↓	
	⋮	
	…	0x090a
x[3]	16	0x0909
x[2]	8	0x0908
x[1]	4	0x0907
x[0]	2	0x0906
y	0x0908	0x0905
y	0x0908	0x0904
z	2	0x0903
		0x0902
	⋮	

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)

```
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];
```

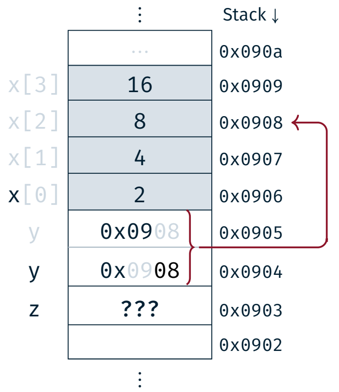


	Stack ↓	
	⋮	
	…	0x090a
x[3]	16	0x0909
x[2]	8	0x0908
x[1]	4	0x0907
x[0]	2	0x0906
y	0x0908	0x0905
y	0x0908	0x0904
z	8	0x0903
		0x0902
	⋮	

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)

```
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]; // ⚡⚡⚡
```



	Stack ↓	
	⋮	
	…	0x090a
x[3]	16	0x0909
x[2]	8	0x0908
x[1]	4	0x0907
x[0]	2	0x0906
y	0x0908	0x0905
y	0x0908	0x0904
z	???	0x0903
		0x0902
	⋮	

Hands-on: Pointers

No Screencast

Hands-on: Pointers, Arrays & Structures

- 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`