Systemnahe Informatik

Übungsgruppe Xeon Phi

Dominik Walter

Sommersemester 2018

Memory Alignment

Memory:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
int16		int16		int16		int16		int16		int16		int16		int16	
int32				int32				int32				int32			
int64								int64							

Memory Alignment in C

```
struct A {
        char c1;
        int64_t i1;
        char c2;
        int32_t i2;
};
```

```
struct B {
        int64_t i1;
        int32_t i2;
        char c1;
        char c2;
};
```

```
struct A {
        char c1;
        char pad[7];
        int64_t i1;
        char c2;
        char pad[3];
        int32_t i2;
};
```

```
struct B {
        int64_t i1;
        int32_t i2;
        char c1;
        char c2;
        char pad[2];
};
```

Memory Alignment in RISCV

```
.data 0x100
char_1:          .space 1        #Adresse: 0x100
int64_t_1:       .space 8        #Adresse: 0x101
char2_2:         .space 1        #Adresse: 0x109
int32_t_2:       .space 4        #Adresse: 0x10A
```

Memory Alignment in RISCV

```
.data 0x100
char_1:          .space 1          #Adresse: 0x100
.align 3
int64_t_1:       .space 8          #Adresse: 0x108
char_2:          .space 1          #Adresse: 0x110
.align 2
int32_t_2:       .space 4          #Adresse: 0x114
```