

# CS315-02 RISC-V Structs Linked Lists

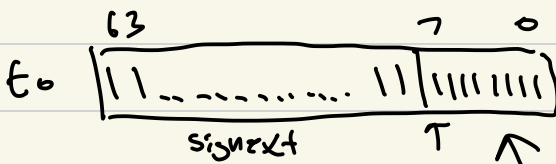
## RISC-V Memory Instructions Loads and Stores

lb load byte

lbu load byte unsigned

Registers

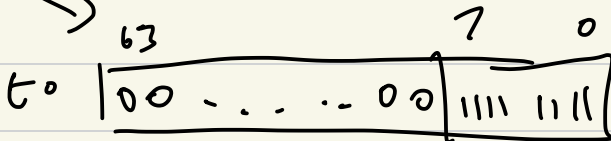
Memory



lb t0, (a0)

lbu t0, (a0)

-1 11111111



char  
unsigned char

int8\_t  
uint8\_t

```
struct foo_st {  
    char a;  
};
```

```
struct foo_st foo;  
struct foo_st *fp;
```

fp = &foo;

fp → a = 'z'.

C = fp → a;

---

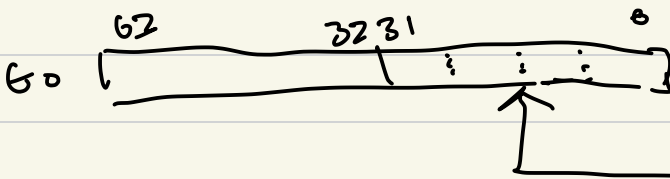
ASM

mov rax, fp

struct-s:

mov rax, [rax]

lw load word - signed  
lwu load word unsigned



lh half word  
lhu 16 bit int16\_t  
int16\_t  
short  
lhu 16 bit uint16\_t  
uint16\_t

---

ld

~~ldo~~ not needed on 64 bit machine

---

Stores

sb

sw

sd

no unaligned versions

---

structs

C  
struct foo {  
int a;  
int b;  
};

ASM  
# struct foo  
# 0 : int a  
# 4 : int b

```
struct foo {
```

```
    int64_t a;
```

```
    int64_t b;
```

```
}
```

```
# struct foo
```

```
# 0 : int64_t a
```

```
# 8 : int64_t b
```

```
struct foo {
```

```
    char c;
```

```
    int a;
```

```
    int64_t b;
```

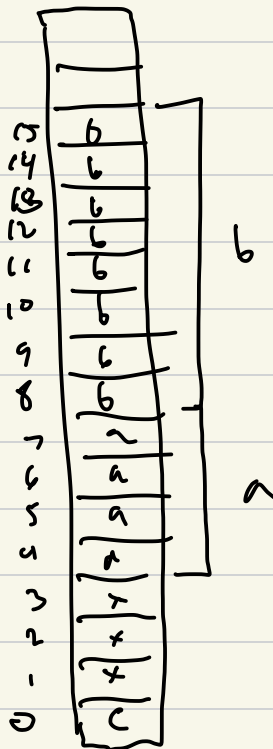
```
}
```

```
# struct foo
```

```
# 0 : char c
```

```
# 4 : int a
```

```
# 8 : int64_t b
```



```
struct foo {
```

```
    int a;
```

```
    char b;
```

```
    char c;
```

```
    int d;
```

```
    char e;
```

3

```
# struct foo
```

```
# 0 : int a;
```

```
# 4 : char b;
```

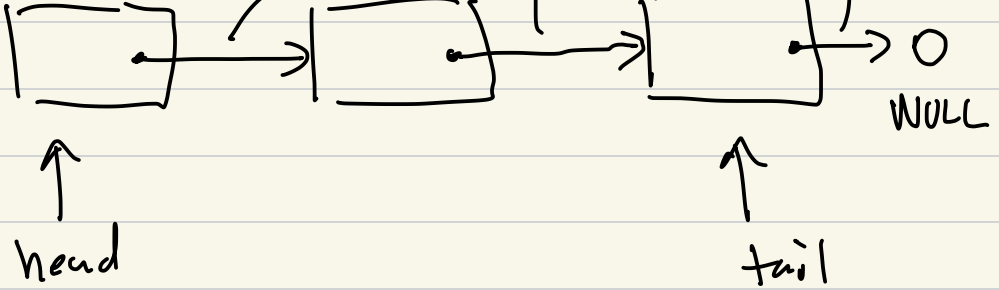
```
# 5 : char c;
```

```
# 8 : int d;
```

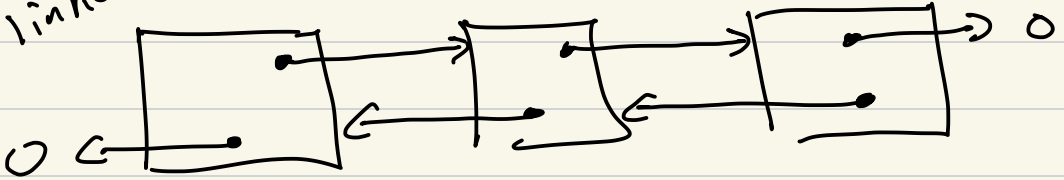
```
# 12 : char e;
```

# Linked Lists

Single linked



double linked



```
struct node_st {
```

```
    struct node_st *next_p;  
    int value;
```

```
};
```

```
# struct node_st
```

```
# 0 : struct node_st * next_p
```

```
# 8 : int value
```

int \* ip;

char \* cp;

struct foo \* fp;

64-bits

8 bytes

