

Shift and Copy One Byte at a Time

```
int32_t ntohs (int32_t arg)
{
    int32_t res;
    res = (arg << 24);
    res |= ((arg & 0xFF00) << 8);
    res |= ((arg >> 8) & 0xFF00);
    res |= (arg >> 24);
    return res;
}
res: 0x25242000
```

So far,
so good!

ECE 220: Computer Systems & Programming

© 2018 Steven S. Lumetta. All rights reserved.

slide 29

29

Shift and Copy One Byte at a Time

```
int32_t ntohs (int32_t arg)
{
    int32_t res;
    res = (arg << 24);
    res |= ((arg & 0xFF00) << 8);
    res |= ((arg >> 8) & 0xFF00);
    res |= (arg >> 24);
    return res;
}
res: 0xFFFFFFF80
```

Er ... is
that right?

ECE 220: Computer Systems & Programming

© 2018-2019 Steven S. Lumetta. All rights reserved.

slide 30

30

Shift and Copy One Byte at a Time

```
int32_t ntohs (int32_t arg)
{
    int32_t res;
    res = (arg << 24);
    res |= ((arg & 0xFF00) << 8);
    res |= ((arg >> 8) & 0xFF00);
    res |= (arg >> 24);
    return res;
}
```

We told the compiler that we
wanted an arithmetic right shift.

ECE 220: Computer Systems & Programming

© 2018 Steven S. Lumetta. All rights reserved.

slide 31

31

A Correct Variant of ntohs

```
uint32_t ntohs (uint32_t arg)
{
    uint32_t res;
    res = (arg << 24);
    res |= ((arg & 0xFF00) << 8);
    res |= ((arg >> 8) & 0xFF00);
    res |= (arg >> 24);
    return res;
}
```

Note use of
uint32_t.

ECE 220: Computer Systems & Programming

© 2018 Steven S. Lumetta. All rights reserved.

slide 32

32