

CS 631-02 Conversions & Evaluation

Lab01 & Lab02

Project 01 starter

321

Conversions

$$3 \times \underline{10^2} + 2 \times \underline{10^1} + 1 \times \underline{10^0}$$

Binary - base 2

0 6 0 1 0 1

0 6 1 1 0 1

64 32 16 8 4 2 1

$$(1) \underline{2^3} + (1) \underline{2^2} + (0) \underline{2^1} + 1(\underline{2^0})$$

$$8 + 4 + 0 + 1 = 13$$

Hexadecimal (Hex) base 16

0xF3B

0-9 A-F a-f
10 15

$$F \times \underline{16^2} + 3 \times 16^1 + B \times 16^0$$

$$15 \times 256 + 48 + 11 = 3879$$

Converting HEX to BIN BIN to HEX

HEX 0 x F3 B
 ↓ ↑
BIN 0 x 1111 0011 1011

Conversions

"214"
"0b1101" → uint32_t
"0xF3B"

parse_operand()

TK_INTLIT
TK_BINLIT
TK_HEXLIT

STR to VINT

"1011"

char *s = "1011"
 ↑ ↓
 s[0] s[1]

uint32_t tmp = 0;
uint32_t value = 0;
char d;

d = s[0];
tmp = d - '0';

'0' 48
'1' 49

value = tmp 1

v = 1

d = s[1];
tmp = d - '0'; 0

v = 0610

v = 2

value = (value * 2) + tmp 2
 ↑
 base

d = s[2]

tmp = d - '0' 1

value = (value * 2) + tmp

4 + 1 = 5

d = s[3]

tmp = d - '0' 1

value = (value * 2) + tmp

10 + 1 = 11

HEX

0 - 9

A - F

← to upper

a - f

d - '0'

(d - 'A') + 10

int long

int -b 10 "241"

241

Scan parsing code

↓
vint32_t

↓

str
print

uint32_t to str

base = 10 decimal

uint32_t value = 241;

char out[64];

out[0] = '2'

out[1] = '4'

out[2] = '1'

out[3] = '\0' 0

241

uint32_t tmp; base

$$241 / 10 = 24$$

$$241 \% 10 = 1$$

$$tmp = value \% \underbrace{10}_{\text{base}}$$

out[0] = '0' + tmp '1'

$$value = value / 10$$

$$tmp = value \% 10 4$$

out[1] = '0' + tmp '4'

done?
value = 0

width -w 4, 8, 16, 32

only -b 2 -b 16 -b 10 signed

nt -b 16 -w 4 -e "0xAC"

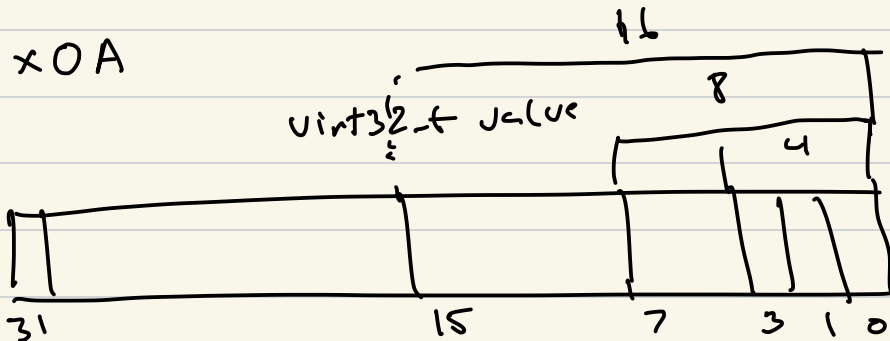
0xC

nt -b 16 -w 4 -e "0xAC >> 4"

0xA

nt -b 16 -w 8 -e "0xAC >> 4"

0x0A



- w 16

value = value & 0xFFFF

masking

- w 4 0xF 0b1111

- w 8 0xFF 0b1111 1111

- w 16 0xFFFF 0b1111 1111 1111 1111

Compute the mask from width

$$\boxed{\text{mask} = (0b1 \ll \text{width}) - 1}$$

0b1
⇐ 0b10000

hex	=	dec	bin
0x10	=	16	0b10000
- 1		- 1	- 1
0x0F		15	0b0 <u>1111</u>

signed output

nt -e "-3"
-3

nt -b2 -w4 -e "-3"
0b 1101

0011
int 1100
+1
1101

nt -b2 -w8 -e "-3"
0b 1111 1101

$\begin{array}{|c|c|c|c|} \hline 3' & & & 0 \\ \hline 0 & \dots & 0 & 11 \\ \hline \end{array}$
- 111 111 1101

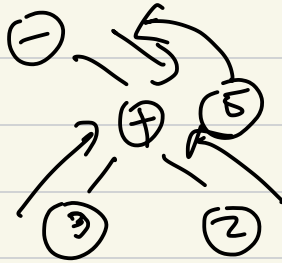
nt -b10 -w4 -e "0b 1101"
-3

[nt -b10 -w4 -e "0b 1101" -0
13

nt -b10 -e "0xFFFF FFFF"
-1

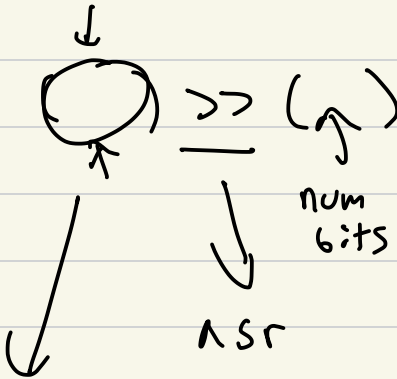
nt -b10 -e "0xFFFF FFFF" -0
42.....

Evaluation



bitwise

>> ASR



$(uint32_t)((int32_t)v1) >> v2$

eval_print()

width

- 1) constrain the bits we care about
 - 2) for %d (signed) determine if print negative (-)
 - 3) number of hex/bin digit to output
-

uint32_t x = -3;

if (x > -3) {

printf("%d", -3);