Lab03 Questions

MakeFiles

maketest → runs on raspberry Pi

Lab03 requirements → no extra files

binaries lab03 X

Big Picture: How do computers represent information?

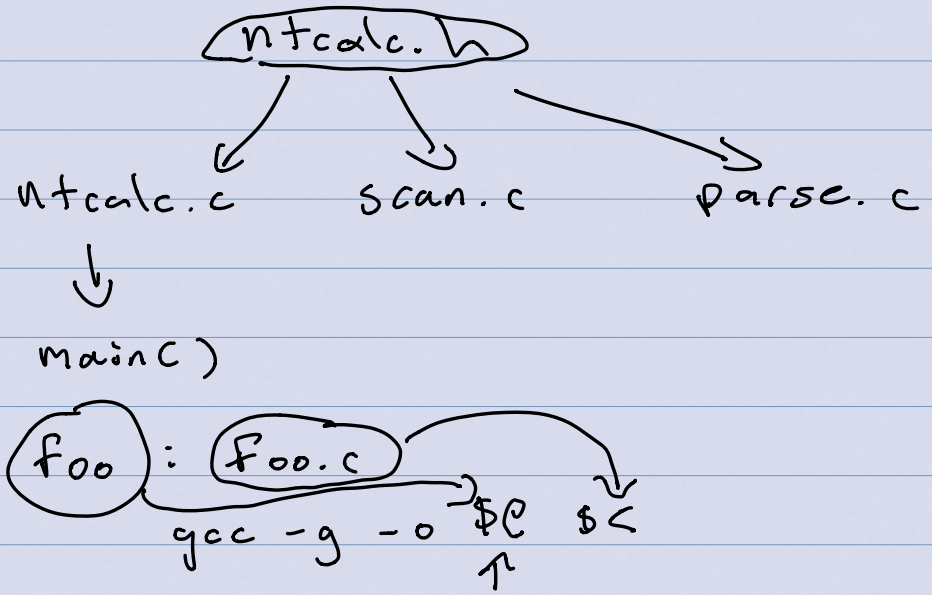
Project02 ntlang working with different number representations

decimal hexadecimal binary

bit manipulation

subset of the C language

---



program ::= expression EOT

expression ::= operand (operator operand)*

operand ::= integer
| '-' operand
| '(' expression ')'

accept  parse_expression  accept

1

1+2

1+2+3

---

Big Picture

How do computers represent and process data?

Data is all binary 01010110

Decimal 0 1 2 .. 9

Binary 0 1

→ Hexadecimal 0 .. 9 A .. F

Project 02

| | | | Output |
|---|---|---|---|
| 1+2 | dec | → | 3 |
| 0x1 + 0x2 | hex | → | 0x3 |
| 0b1 + 0b10 | bin | → | 0b11 |

Number systems

413 base 10 decimal $0, 1, \ldots, 9$

$400 + 10 + 3 = 413$

$(4 \times 10^2) + (1 \times 10^1) + (3 \times 10^0) = 413$

$10^2 = 100$ $10^1 = 10$ $10^0 = 1$

Binary base 2

| Dec | |
|---|---|
| 0 | 000 |
| 1 | 001 |
| 2 | 010 |
| 3 | 011 |
| 4 | 100 |
| 5 | 101 |
| 6 | 110 |
| 7 | 111 |
| 8 | 1000 |
| 9 | 1001 |

Binary

32.0

1011

$(1 \times 2^3) + (0 \times 2^2) + (1 \times 2^1) + (1 \times 2^0)$

$1 \times 8 + 0 \times 4 + 1 \times 2 + 1 \times 1$

$8 + 0 + 2 + 1$

$= 11_{10}$ (base 10)

$2^0 = 1$

$2^1 = 2$

$2^2 = 4$

$2^3 = 8$

$2^4 = 16$

$2^5 = 32$

$\rightarrow 2^6 = 64$

$2^7 = 128$

$2^8 = 256$

$2^9 = 512$

$2^{10} = 1024$

$2^{11} = 2048$

16 8 2 1

$1\ 1011 = 11_{10}$

16 4 2

$10110 = 22_{10}$

11

int x = 0b1011

word size

int x;

4 bytes (32 bits)

32

0b0000 ... 1101

0b1101 ==

0b0000|0000|0000|0000|0000|0000|0000|1101

Hexadecimal base 16

| Dec | Hex | Bin |
|---|---|---|
| 0 | 0 | 0000 |
| 1 | 1 | 0001 |
| 2 | 2 | 0010 |
| 3 | 3 | 0011 |
| 4 | 4 | 0100 |
| 5 | 5 | 0101 |
| 6 | 6 | 0110 |
| 7 | 7 | 0111 |
| 8 | 8 | 1000 |
| 9 | 9 | 1001 |
| 10 | A | 1010 |
| 11 | B | 1011 |
| 12 | C | 1100 |
| 13 | D | 1101 |
| 14 | E | 1110 |
| 15 | F | 1111 |

0x2B4

$(2 \times 16^2) + (B \times 16^1) + (4 \times 16^0)$

$(2 \times 256) + (11 \times 16) + (4 \times 1)$

512 + 176 + 4

512 + 180

= 692

Octal base 8 0, 1, ..., 7

Base conversion

- bin → dec
- hex → dec
- dec → hex
- dec → bin

easy

- hex → bin
- bin → hex

0xA7 ⇒ (A×16¹) + (7×16⁰)

(10×16) + (7×1) = 167

bin

0b 1010 0111 ⇒ (1×2⁷) + (1×2⁵) + (1×2²) + (1×2¹) + (1×2⁰)

128 + 32 + 4 + 2 + 1

= 160 + 7

= 167

0x12CE

?

8 bits

0b 1001 1000

hex

0x 9 8

0b 10 1101 1100

0b 0010 1101 1100

0xFF, 0x12 ← 8 bits = 1 byte

0x F2

* 4 bits nibble

---

Dec → Bin

$100_{10} = ?_2$

~~128~~ ~~64~~ ~~32~~ 16 8 (4) 2 1 0

$64 + 32 + 4 = 100$

$100 - 64 = 36$ 64

$36 - 32 = 4$ 32

4

Dec → Bin Algorithm

| | | | | Remainder |
|---|---|---|---|---|
| 100 | / 2 | = 50 | lsb | 0 |
| 50 | / 2 | = 25 | | 0 |
| 25 | / 2 | = 12 | | 1 |
| 12 | / 2 | = 6 | | 0 |
| 6 | / 2 | = 3 | | 0 |
| 3 | / 2 | = 1 | | 1 |
| 1 | / 2 | = 0 | msb | 1 |

$25 - (2 \times 12)$

64 32 4
1 1 0 0 1 0 0

25 / 2 = 12
25 % 2 = 1

64 + 32 + 4 = 100

Dec ⇒ Hex

100 / 16 =

Remainder
hex

Bitwise operators
~ & | ^
>> <<

Project 02
ntlang