Project02 loose ends

✓ Grading

✓ Rubric

→ SCAN_TOKEN_LEN ←

Deltas from Lab04
- w handling
- v unsigned output

Review

uint32_t to hexstr

Project02 Grading

Due Mon Sep 21 11:59pm
In GitHub Repo
Individual work

Rubric:

60% Passing tests in make test
40% Interactive grading and code quality
- 10% given question Q1
- 10% Q2
- 10% Q3
- 10% Code quality

Code quality:

- -2 inconsistent spacing or indentation
- -2 inconsistent naming or commenting
- -2 commented out code
- -2 redundant code
- -2 overly complicated code of messy repo

Source
Makefile
project02 ← maketest will use

No other files
X .o
X project02
X ntlang
X scan_3.c

Readme → brief
What?
How to build?
How to use?

Tuesday Sep 22 → InteractiveGrading
by Mon Google signup sheet

1) Run maketest

2) Q1 Q2, Q3
3) Code quality } notes

1 point extra credit

$ ./project02 -e "0xA1234ABCD"
4 bit 32 bits
36 bits

Literal to big: 0xA1234ABCD

$ ./project02 -e "0b11[0110 ... 1100]
32

large dec int
$ ./project02 -e "12...45"
uint32_t
v > 4294967298
64 bit
long long llvj

X $ ./project02 -e "-(423...91)"
conj

eval

$-2^{(n-1)}$ to $2^{(n-1)}-1$

-2^{31}

Deltas

Lab04 → Project02

Scanner (scan.c)

hexlit

ops: >>, >-, &, |, ^

LSR, ASR, AND, OR, XOR

Parser (parse.c)

parse_operand

conv.c

hexlit ← conv_hexstr_to_uint32()

intlit ← conv_decstr_to_uint32()

TODO

parse_expression

Add support for >>, >-, &, |, ^

eval.c

additional ops

-w -v -b

ntcalc.c

result = eval_tree(parse_tree)

uint32_t

-w, -u, -b

⟹ ntcalc_print_result(&config, result)

Binary (-b 2)

-b 2 binary (don't care about -u)

"0b0000...0110"

-w 32

-w 4 ← lowest 4 bits

"0b0110" [31, 30, ..., 1, 0]

(width-1) ... 0

3 2 1 0

4, 8, 16, 32

Hex (-b 16)

-b 16 hexadecimal output

-w 4, 8, 16, 32

multiples of 4

result uint32_t

0001 0010 0011 0100 1111 1110 1101 1100

0x 1 2 3 4 F E D C

# of hex digits

w/4

-w 32 ndigits = 32/4 = 8

-w 16 ndigits = 16/4 = 4

-w 8 ndigits = 8/4 = 2

-w 4 ndigits = 4/4 = 1

Hint: Litecnls → uint32_t

base ↑

- dec to uint32_t 10
- hex to uint32_t 16
- bin to uint32_t 2

1 func

- base 2 '0', '1'
- base 10 '0', ... '9'
- base 16 '0'..'9', 'A'..'F', 'a'..'f'

Decimal Output

-b 10 -w -u

-u unsigned output

result

0001 0010 0011 0100 | 1111 1110 1101 1100

w 32 0

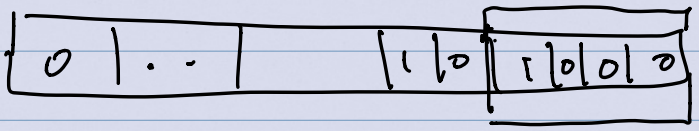
conv_uint32_to_decstr (width, result, v)

if (width == 16) {
    v = v & 0xD000FFFF
}

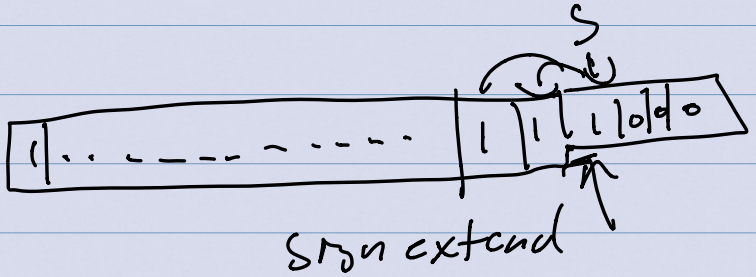
v

for unsigned int
    apply the width
    print the value

for signed
    apply the width_w4



↑ 2's comp



Conversion also used

on a 32 bit signed value

signed SV

int sv

if (sv < 0) {

neg = true

}

sv = sv * -1 ; pos

pos →

unsigned → Conversion from Unsigned to str

print

if (neg)

str[0] = '-' ; ←

i = 1

}

str[i] ← conv str

uint32_t to hexstr

v =

31 30 29 28

0001 0010 0011 0100 1111 1110 1101 1100

binstr

```
bool   bit = v >> 31 & 0b1
       ch = '0' + bit
```

hexstr

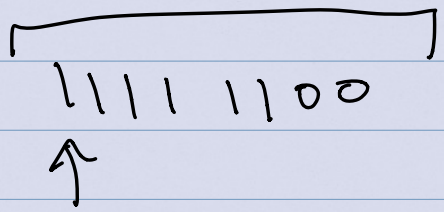
```
bool   nibble = (v >> 28) & 0b1111
       if (nibble >= 0 && nibble <= 9)
           ch = '0' + nibble
       else (nibble >= 10 && nibble <= 15)
           ch = 'A' + (nibble - 10)
   }
```

result — w 16

0001 0010 0011 0100 1111 1110 1101 1100

-v

0 000 0000 000c 0400 1111 1110 1101 1100


1111 1100