

CS315-02 Optional Args & Base Conversion

ls -a
└─

ls -a -l

ls -l -a

X ls -al

gcc -o <filename> <source>

loop:

check for options

consume option

also get required arg

echorepeat -h -f -n <count> <string>

echorepeat -h -f foo

foo

Numbers

245

quantity

"245"

String

byte byte byte

'2' '4' '5'

50 52 53

binary

245

integer
machine

1111 0101

↓
add r1, r0, r1

Bases

10 Decimal

2 Binary

16 Hexadecimal

Decimal Base 10

245
↑ ↑ ↑
100s place 10s place 1s place

$$(2 \times \underline{10^2}) + (4 \times \underline{10^1}) + (5 \times \underline{10^0})$$
$$2 \times 200 + 4 \times 10 + 5 \times 1$$
$$200 + 40 + 5$$
$$= 245$$

Binary Base 2

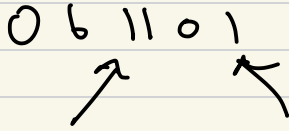
061101
8421

int x = 13;
int x = 0b1101;
int x = 0xD;

$$(1 \times \underline{2^3}) + (1 \times \underline{2^2}) + (0 \times \underline{2^1}) + (1 \times \underline{2^0})$$
$$(1 \times 8) + (1 \times 4) + (0 \times 2) + (1 \times 1)$$
$$8 + 4 + 0 + 1$$
$$= 13$$

4 bit binary number

0 1 1 0 1



most
significant
bit

least
significant
bit

n-bit binary number

2^n possible values

$$2^4 = 16$$

$$2^4 = 16$$

0 to 15

range 0 to $(2^n) - 1$

0000 1111

Hexadecimal Base 16

Dec (10) Bin (2) Hex (16)

0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A a
11	1011	B b
12	1100	C c
13	1101	D d
14	1110	E e
15	1111	F f

0x2AF

$$(2 \times 16^2) + (A \times 16^1) + (F \times 16^0)$$

$$\begin{array}{r} 2 \times 256 + 10 \times 16 + 15 \times 1 \\ 512 + 160 + 15 \\ = 687 \end{array}$$

Project 1

```
$ numconv 687 -o 16  
0x2AF
```

```
$ numconv 0x2AF -o 10  
687
```

numstr "1245" → Machine int → numstr (base)
uint32_t

decstr
binstr
hexstr

→ uint32_t

→ decstr
→ binstr
→ hexstr

input base

```
if prefix(str) == "0b" → bin (base 2)  
if prefix(str) == "0x" → hex (base 16)  
else dec (10)
```

char *s = "245";

ASCII

s[0] = '2'

s[1] = '4'

s[2] = '5'

s[3] = '\0'

'0' = 48 ←

'1' = 49

'2' = 50 ←

'3' = 51

'4' = 52

'5' = 53

int v0 = s[0] - '0';

= 50 - 48

= 2

int v1 = s[1] - '0';

= 52 - 48

= 4

int v2 = s[2] - '0';

= 53 - 48

= 5

int v = (v0 × 100) + (v1 × 10) + (v2 × 1)

200 + 40 + 5

= 245

base

```
uint32_t decstr_to_uint32 (char *s) ,
```

```
uint32_t v = 0 ;
```

```
int d ;
```

```
int i = 0 ;
```

```
while ( s[i] != '\0' ) {
```

```
    v = v * 10 ;
```

```
    d = s[i] - '0' ;
```

next digits

```
    v = v + d ;
```

```
    i = i + 1
```

```
}
```

```
return v ;
```

```
}
```

uint32_t $\rightarrow$ num str

$$245 / 10 = 24$$
$$\text{remainder} = 5$$

int v = 245;

$$245 \% 10 = 5$$

int d0, d1, d2 base

$$d_0 = v \% 10$$

$$= 245 \% 10$$

$$= 5 \rightarrow \text{char } c_0 = '0' + d_0 = '5'$$

$$v = v / 10$$

$$= 24$$

$$d_1 = v \% 10$$

$$= 24 \% 10$$

$$= 4$$

$$c_1 = '0' + d_1 = '4'$$

$$v = v / 10$$

$$= 2$$

$$d_2 = v \% 10$$

$$= 2 \% 10$$

$$= 2$$

$$c_2 = '0' + d_2 = '2'$$