

6/17/2019

learn part and let learn

地址: 12位: 4 byte  
64位: 8 byte

有符号	运算	类型转换
$uint: \sum_{i=0}^{31} x_i \cdot 2^i (0-1) 2^{31-1}$ $int: (-x_{31} \cdot 2^{31}) + \sum_{i=0}^{30} x_i \cdot 2^i$ (补码有偏, 符号位左移一位) 补码: $(-1)^{x_{31}} \cdot 2^{31} + \sum_{i=0}^{30} x_i \cdot 2^i$	①: extension: unsigned to signed, 高位 0 to 1 truncate: 2's complement ②: 加, 减: 均为 mod, 均为 2 <sup>32</sup> ③: 乘: $-x \cdot -y = x \cdot y$ ④: 移位: 左移: $x \ll k$ , 右移: $x \gg k$ (signed: 符号位补)	using sign - 一起转换: 符号位补 易产生问题 $\Rightarrow$ 用 using 有符号相同表示不同
double like float $s: 1 \text{ bit}, e: 11 \text{ bits}, m: 52 \text{ bits}$ $s: \text{sign}, e: \text{exponent}, m: \text{mantissa}$ $e: \text{biased}$ $m: 0: \text{normalized}, (1 \leq x_i \leq 2)$ $e_{20}: (0 \leq x_i \leq 2)$ $e_{21}: \begin{cases} \infty: \infty \\ \neq 0: NaN \end{cases}$	①: $x: s_1 s_2, e_4 e_5, m_1 m_2$ + 通变率 ②: found: too overflow 指数 (四舍五入) 与 found to even	$int \rightarrow \begin{cases} float: \text{可转换} \\ double: \text{转换} \end{cases}$ $float / double \rightarrow int$ (向 0 舍入) $double \rightarrow float$ (舍入)

little: val+1, val-1  
 (val+1) (补码 + 4 位)

地址:  $A[i:j]$   
 $A + (i - c) \cdot k$

地址: 对齐  
 char: 1 byte  
 short: 2 bytes  
 int/float: 4 bytes  
 double: 8 bytes  
 char\*: 32 bytes, 64 bytes

hierarchy: cache  $\Rightarrow$  内存 (RAM)  $\Rightarrow$  硬盘 (Disk)  
 s, e, b

2<sup>e</sup>(E) 个 block  
 block:  $\sqrt{\text{tag}} \cdot 2^b$

address:  $\text{tag} | \text{set} | \text{block offset}$

② 地址对 set, 比较 block  
 tag-致命中 (hit)  
 命中: 命中 (hit)  
 未命中: (miss, evict)  
 LRU (least recent used)  
 random

hit: write through, write-back  
 miss: write allocate, no-write allocate, write-back, write allocate





3X  
[11]  
各組  
輸入  
106

山  
東四地(周)

调用: ①: 重设  $ebp$ :  $mov\ esp, ebp$   
 $ebp = esp$   
 ② 压入寄存器 ( $ebx, esi, edi$ )  
 ③ 操作

① 回值: %car  
(① 回值, caller & callee 都有资格成为传参)

⑥  $PC := \text{val}(P(\text{fix}))$   
 $\text{val}(C / \text{val}(M(\text{call\_jump\_let})))$   
 $\overline{\text{fix}} PC$





只操作

① 取符号位  $(x \gg 31) \& 1$

② 取前n位全1的mask  $(1 \ll n) - 1$

③ 取n-1位全1的mask  $(1 \ll n) - 1$

④ 不用减法求  $x-y$ :  $x + (ny+1)$

⑤ 不用if else:

```
if (a==1) return x
if (b==1) return y
if (c==1) return z
```

①  $m-a = a+1$  ( $=1$  全1)  
②  $m-b = b+1$  ( $=0$  0)  
③  $m-c = c+1$

④  $\text{return } (m-a \& x) | (m-b \& y) | (m-c \& z)$

⑥ 统计1的个数/奇偶性

⑦ 统计1的个数/奇偶性

⑧ 统计1的个数/奇偶性

⑨ 统计1的个数/奇偶性

⑩ 统计1的个数/奇偶性

⑪ 统计1的个数/奇偶性

⑫ 统计1的个数/奇偶性

⑬ 统计1的个数/奇偶性

⑭ 统计1的个数/奇偶性

⑮ 统计1的个数/奇偶性

⑯ 统计1的个数/奇偶性

⑰ 统计1的个数/奇偶性

⑱ 统计1的个数/奇偶性

⑲ 统计1的个数/奇偶性

⑳ 统计1的个数/奇偶性

㉑ 统计1的个数/奇偶性

㉒ 统计1的个数/奇偶性

6/19/2019  
Programming

C语言

① 统计1的个数/奇偶性

① 统计1的个数/奇偶性

② 统计1的个数/奇偶性

③ 统计1的个数/奇偶性

④ 统计1的个数/奇偶性

⑤ 统计1的个数/奇偶性

⑥ 统计1的个数/奇偶性

⑦ 统计1的个数/奇偶性

⑧ 统计1的个数/奇偶性

⑨ 统计1的个数/奇偶性

⑩ 统计1的个数/奇偶性

⑪ 统计1的个数/奇偶性

⑫ 统计1的个数/奇偶性

⑬ 统计1的个数/奇偶性

⑭ 统计1的个数/奇偶性

⑮ 统计1的个数/奇偶性

⑯ 统计1的个数/奇偶性

⑰ 统计1的个数/奇偶性

⑱ 统计1的个数/奇偶性

㉑ 统计1的个数/奇偶性

㉒ 统计1的个数/奇偶性

求师得教育实验室



由 扫描全能王 扫描创建