字符串和浮点

在VC6或者其他编译器上

- `strlen`做到了无分支求长度

```
lea edi, [str]
or ecx, -1      ; ecx 为正数最大，表循环次数
xor eax, eax    ; al = 0
repne scasb     ; 与al不相等继续
not ecx
dec ecx

; 数学推导 ecx = 0xffffffff - len - 1
;           ecx = -1 - len - 1
;           ecx = -2 - len
;           len = -2 - ecx
;           len = -2 + neg(ecx)
;           len = -2 + not(ecx) + 1
;           len = not(ecx) - 1
```

- `strcpy`先使用无分支求长度，而后拷贝

```
lea edi, [str]
or ecx, -1
xor eax, eax
repne scasb
not ecx
; ------------------- 以上求长度ecx = strlen(str) + 1
sub edi, ecx    ; edi 还原回开头
lea edx, [dst]
mov eax, ecx
mov esi, edi
mov edi, edx
shr ecx, 2      ; ecx / 4 求有几个4字节
rep movsd       ; 4字节拷贝
mov ecx, eax
and ecx, 3      ; ecx % 4 求最后剩余几个字节
rep movsb       ; 1字节拷贝
```

- `memcpy`使用串传输
- `strcmp`有两个`sbb`结尾特征

```
.text:00401070                 lea     esi, [esp+30h+str2]
.text:00401074                 lea     eax, [esp+30h+str1]
.text:00401078
.text:00401078 loc_401078:                             ; CODE XREF:
_main+9A•j
.text:00401078                 mov     dl, [eax]
.text:0040107A                 mov     bl, [esi]
.text:0040107C                 mov     cl, dl
```

```
.text:0040107E                  cmp     dl, bl
.text:00401080                  jnz     short loc_4010A0 ; if str1 > str2:
.text:00401080                                           ;     eax = eax - eax
- cf = 0
.text:00401080                                           ;     eax = eax - -1
- cf= 1
.text:00401080                                           ; else
.text:00401080                                           ;     eax = eax - eax
- cf = -1
.text:00401080                                           ;     eax = -1 - -1 -
cf = -1
.text:00401082                  test    cl, cl           ; 判断是否到末尾`\0`
.text:00401084                  jz      short loc_40109C ; 返回0
.text:00401086                  mov     dl, [eax+1]
.text:00401089                  mov     bl, [esi+1]
.text:0040108C                  mov     cl, dl
.text:0040108E                  cmp     dl, bl
.text:00401090                  jnz     short loc_4010A0 ; if str1 > str2:
.text:00401090                                           ;     eax = eax - eax
- cf = 0
.text:00401090                                           ;     eax = eax - -1
- cf= 1
.text:00401090                                           ; else
.text:00401090                                           ;     eax = eax - eax
- cf = -1
.text:00401090                                           ;     eax = -1 - -1 -
cf = -1
.text:00401092                  add     eax, 2
.text:00401095                  add     esi, 2
.text:00401098                  test    cl, cl           ; 判断是否到末尾'\0'
.text:0040109A                  jnz     short loc_401078
.text:0040109C
.text:0040109C loc_40109C:                               ; CODE XREF:
_main+84•j
.text:0040109C                  xor     eax, eax         ; str1 == str2, 返回0
.text:0040109E                  jmp     short loc_4010A5
.text:004010A0 ; ---------------------------------------------------------
---------------
.text:004010A0
.text:004010A0 loc_4010A0:                               ; CODE XREF:
_main+80•j
.text:004010A0                                           ; _main+90•j
.text:004010A0                  sbb     eax, eax         ; if str1 > str2:
.text:004010A0                                           ;     eax = eax - eax
- cf = 0
.text:004010A0                                           ;     eax = eax - -1
- cf= 1
.text:004010A0                                           ; else
.text:004010A0                                           ;     eax = eax - eax
- cf = -1
.text:004010A0                                           ;     eax = -1 - -1 -
cf = -1
.text:004010A2                  sbb     eax, 0FFFFFFFFh
```

- `float`和`double`使用浮点指令集FPU

- 单操作数隐含的另一个操作数是st0，浮点寄存器循环栈`st0-7`，64位
- `fld -> push, fst -> top, fstp -> pop`等带p后缀的都是有出栈操作

```
.text:0040102F                 fld     [esp+18h+var_C]  ; var_C压st0
.text:00401033                 fdiv    [esp+18h+var_8]  ; st0 / var_C -
> st0
.text:0040103A                 fstp    [esp+14h+var_14] ; 将st0压入栈中
```

- 浮点比较
  - 通过`fnstsw`将状态位传入ax中，然后查看状态位判断比较类别

```
.text:0040105F                 fld     [esp+18h+var_C]  ; var_C压st0
.text:00401063                 fcomp   [esp+18h+var_8]  ; var_8 与 st0
比较
.text:0040106A                 fnstsw  ax               ; 将状态位传入ax
.text:0040106C                 test    ah, 41h          ; 比较状态位
.text:0040106F                 jnz     short loc_40107E
```

在高版本VS编译器上

- `strlen`依然使用循环
- `strcpy`依然使用循环
- `strcmp`有一个`sbb`和一个`or`结尾特征

```
.text:0040104B                 lea     ecx, [ebp+str1]
.text:0040109A                 lea     eax, [ebp+str2]
; ...
; ...

.text:004010A2 loc_4010A2:                             ; CODE XREF:
sub_401040+7C↓j
.text:004010A2                 mov     dl, [ecx]
.text:004010A4                 cmp     dl, [eax]
.text:004010A6                 jnz     short loc_4010C2 ; if str1 > str2:
.text:004010A6                                         ;     eax = eax - eax
- cf = 0
.text:004010A6                                         ;     eax = eax | 1 =
1
.text:004010A6                                         ; else
.text:004010A6                                         ;     eax = eax - eax
- cf = -1
.text:004010A6                                         ;     eax = eax | 1 =
-1
.text:004010A8                 test    dl, dl           ; 判断是否到末尾`\0`
.text:004010AA                 jz      short loc_4010BE ; str1 == str2,
return 0
.text:004010AC                 mov     dl, [ecx+1]
.text:004010AF                 cmp     dl, [eax+1]
```

```
.text:004010B2                 jnz     short loc_4010C2 ; if str1 > str2:
.text:004010B2                                          ;     eax = eax - eax
- cf = 0
.text:004010B2                                          ;     eax = eax | 1 =
1
.text:004010B2                                          ; else
.text:004010B2                                          ;     eax = eax - eax
- cf = -1
.text:004010B2                                          ;     eax = eax | 1 =
-1
.text:004010B4                 add     ecx, 2
.text:004010B7                 add     eax, 2
.text:004010BA                 test    dl, dl
.text:004010BC                 jnz     short loc_4010A2
.text:004010BE
.text:004010BE loc_4010BE:                              ; CODE XREF:
sub_401040+6A↑j
.text:004010BE                 xor     eax, eax        ; str1 == str2,
return 0
.text:004010C0                 jmp     short loc_4010C7
.text:004010C2 ; ---------------------------------------------------------
---------------
.text:004010C2
.text:004010C2 loc_4010C2:                              ; CODE XREF:
sub_401040+66↑j
.text:004010C2                                          ; sub_401040+72↑j
.text:004010C2                 sbb     eax, eax        ; if str1 > str2:
.text:004010C2                                          ;     eax = eax - eax
- cf = 0
.text:004010C2                                          ;     eax = eax | 1 =
1
.text:004010C2                                          ; else
.text:004010C2                                          ;     eax = eax - eax
- cf = -1
.text:004010C2                                          ;     eax = eax | 1 =
-1
.text:004010C4                 or      eax, 1
```

- `float`和`double`使用多媒体指令集

```
; 加法
.text:00401086                 lea     eax, [ebp+var_4]
.text:0040108A                 lea     eax, [ebp+var_C]

.text:00401098                 movss   xmm0, [ebp+var_4]    ; var_4 -> xmm0
.text:004010A0                 cvtps2pd xmm0, xmm0  ; 压缩单精度浮点值转换成压缩
双精度浮点值
.text:004010A3                 addsd   xmm0, [ebp+var_C]    ; xmm0 + var_C -
> xmm0
.text:004010A8                 movsd   [esp+14h+var_14], xmm0   ; -> 结果存放
在栈中
.text:004010AD                 push    offset asc_402108 ; "%f\n"
.text:004010B2                 call    _printf

; 比较
```

```
.text:004010F5                movss   xmm0, [ebp+var_4]
.text:004010FD                cvtps2pd xmm0, xmm0  ; 压缩单精度浮点值转换成压缩双精度浮点值
.text:00401100                comisd  xmm0, [ebp+var_C] ; xmm0 cmp var_C
.text:00401105                jbe     short loc_401114
```