SYSTEM\_INFO结构体包含了当前计算机的信息。这个信息包括计算机的体系结构、中央处理器的类型、系统中中央处理器的数量、页面的大小以及其他信息。

结构原型

typedef struct \_SYSTEM\_INFO { // sinf

union {

DWORD dwOemId;

struct {

WORD wProcessorArchitecture;

WORD wReserved;

};

};

DWORD dwPageSize;

LPVOID lpMinimumApplicationAddress;

LPVOID lpMaximumApplicationAddress;

DWORD dwActiveProcessorMask;

DWORD dwNumberOfProcessors;

DWORD dwProcessorType;

DWORD dwAllocationGranularity;

WORD wProcessorLevel;

WORD wProcessorRevision;

} SYSTEM\_INFO;

## 结构成员

**dwOemId：**

已废弃的成员，保留这个成员是为了向以前版本的Windows NT保持兼容。从Windows NT 3.51和Windows 95的预发行版本开始，使用这个成员的子分支wProcessorArchitecture。

**Windows 95：**系统总是把这个成员的值设为0，也就是PROCESSOR\_ARCHITECTURE\_INTEL的预定义值。

**wProcessorArchitecture：** 指定系统中的[中央处理器](http://baike.baidu.com/view/14045.htm)的[体系结构](http://baike.baidu.com/view/1188494.htm)，可以是如下的值：

PROCESSOR\_ARCHITECTURE\_INTEL

**只适用于Windows NT:** PROCESSOR\_ARCHITECTURE\_MIPS

**只适用于Windows NT:**PROCESSOR\_ARCHITECTURE\_ALPHA

**只适用于Windows NT:** PROCESSOR\_ARCHITECTURE\_PPC

**只适用于Windows NT:** PROCESSOR\_ARCHITECTURE\_UNKNOWN

**wReserved：**

保留供将来使用。

**dwPageSize：**

指定页面的大小和页面保护和委托的颗粒。这是被 VirtualAlloc 函数使用的页大小。

**lpMinimumApplicationAddress：**

指向应用程序和[动态链接库](http://baike.baidu.com/view/887.htm)(DLL)可以访问的最低[内存地址](http://baike.baidu.com/view/404417.htm)。

**lpMaximumApplicationAddress：**

指向应用程序和动态链接库(DLL)可以访问的最高[内存地址](http://baike.baidu.com/view/404417.htm)。

**dwActiveProcessorMask：**

指定一个用来代表这个系统中装配了的[中央处理器](http://baike.baidu.com/view/14045.htm)的掩码。二进制0位是处理器0；31位是处理器31。

**dwNumberOfProcessors：**

指定系统中的处理器的数目。

**dwProcessorType：**

**Windows 95:**

指定系统中中央处理器的类型。

**Windows NT:**

这个成员已经不再有用了，但是为了向Windows 95和以前版本的Windows NT保持兼容，使用**wProcessorArchitecture**,**wProcessorLevel**, 和 **wProcessorRevision** 三个成员定义处理器的类型。

这个成员的值是下面中的一个：

PROCESSOR\_INTEL\_386

PROCESSOR\_INTEL\_486

PROCESSOR\_INTEL\_PENTIUM

**只适用于Windows NT:** PROCESSOR\_MIPS\_R4000

**只适用于Windows NT:** PROCESSOR\_ALPHA\_21064

**dwAllocationGranularity：**

指定已经被分配的[虚拟内存](http://baike.baidu.com/view/976.htm)空间的粒度。例如，如果使用**VirtualAlloc**函数请求分配1byte内存空间，那么将会保留由**dwAllocationGranularity**指定大小byte的[地址空间](http://baike.baidu.com/view/1507129.htm)。在过去，这个值被定为64K并固化在硬件中，但是其它的硬件体系结构可能需要另外的值。

**wProcessorLevel：**

**Windows 95:** 不使用这个成员。

**Windows NT:** 指定系统体系结构依赖的处理器级别。

如果 **wProcessorArchitecture**的值是 PROCESSOR\_ARCHITECTURE\_INTEL，那么 **wProcessorLevel**可以使如下值中的一个：

**值** **含义**

3 Intel 80386

4 Intel 80486

5 Pentium

如果 **wProcessorArchitecture**的值是 PROCESSOR\_ARCHITECTURE\_MIPS，那么 **wProcessorLevel**的值会是00xx的形式，xx是一个8位的立即数。这个成员可以使如下的值：

**值** **含义**

0004 MIPS R4000

如果 **wProcessorArchitecture**的值是 PROCESSOR\_ARCHITECTURE\_ALPHA, 那么 **wProcessorLevel**的值会是xxxx的形式，xxxx是一个16位的处理器版本数（固件中版本号的低16位），它的值可以是下表中的一个：

**十进制值** **含义**

21064 Alpha 21064

21066 Alpha 21066

21164 Alpha 21164

如果 **wProcessorArchitecture**的值是 PROCESSOR\_ARCHITECTURE\_PPC, 那么 **wProcessorLevel**的值会是xxxx的形式，xxxx是一个16位的处理器版本数（处理器版本寄存器的高16位），它的值可以使下表中的一个：

**十进制值** **含义**

1 PPC 601

3 PPC 603

4 PPC 604

6 PPC 603+

9 PPC 604+

20 PPC 620

**wProcessorRevision：**

**Windows 95:** 不使用这个成员。

**Windows NT:** 指定系统体系结构依赖的处理器修订版本号。下表显示了对于每一种处理器体系，处理器的修订版本号是如何构成的。

**处理器体系** **值**

Intel 80386 or 80486 类似xxyz的形式。如果xx等于0xFF，y-0xA 是型号，z 是步进表示符。例如，一个Intel 80486-D0 CPU 组成的系统返回0xFFD0。

如果xx不等于0xFF，'A'是步进表示符，yz 是次要步进表示符。

Intel Pentium, Cyrix, or NextGen 586 类似xxyy的形式。xx是型号，yy是步进值。例如，一个0x0201的值表明这是一个Model 2,Stepping 1的CPU。

MIPS 类似00xx的形式。xx是8位的处理器修订号（PRId寄存器的低8位）。

ALPHA 类似xxyy的形式。xxyy是处理器固件中的修订号的低16位。这个值被显示成这个样子：Model 'A'+xx, yy被忽略。

PPC 类似xxyy的形式。xxyy是处理器版本寄存器的低16位。这个值被显示为[定点小数](http://baike.baidu.com/view/587146.htm)：xx.yy

**注释：**

从Windows NT 3.51和Windows 95的预发行版本开始，SYSTEM\_INFO结构发生了如下变化：

·**dwOemId** 成员被更改成一个结构。这个结构提供了一条新的可以用来描述Windows NT处理器体系结构的路，同时保持了与Windows 95 以及Windows NT以前版本的兼容性。

·**dwProcessorType** 成员在Windows NT系统上已经被废弃，但是仍在Windows 95系统中使用。

·一个先前保留的位于结构末端的双字值被**wProcessorLevel** 和 **wProcessorRevision**两个成员替代。Windows 95 不使用这两个成员。

**快捷信息：**

头文件：winbase.h

**请参见GetSystemInfo**