

缩进、空格、换行、空行、对齐

1. 缩进应该是每个程序员都会做的，只要学过程序就应该知道这个，但是我仍然看过不缩进的程序，或是乱缩进，由网友vector_3d提醒改正
2. 空格。空格能给程序带来什么损失吗？没有，有效的利用空格可以让你的程序读进来更加赏心悦目。而不一

```
ha=(ha*128+*key++)%tabPtr->size;
```

```
ha = ( ha * 128 + *key++ ) % tabPtr->size;
```

有空格和没有空格的感觉不一样吧。一般来说，语句中要在各个操作符间加空格，函数调用时，要以各个参

```
if ((hProc=OpenProcess(PROCESS_ALL_ACCESS, FALSE, pid))==NULL) { }
```

```
if ( ( hProc = OpenProcess(PROCESS_ALL_ACCESS, FALSE, pid) ) == NULL ) { }
```

iii) 换行。不要把语句都写在一行上，这样很不好。如：

```
for(i=0;i<len;i++) if((a[i]< '0' ||a[i]> '9' )&&(a[i]< 'a' ||a[i]> 'z' ))  
break;
```

我拷，这种即无空格，又无换行的程序在写什么啊？加上空格和换行吧。

```
for ( i=0; i<len; i++) { if ( ( a[i] < '0' || a[i] > '9' ) &&  
    ( a[i] < 'a' || a[i] > 'z' ) ) {  
        break;  
    }  
}
```

```
}
```

好多了吧？有时候，函数参数多的时候，最好也换行，如：

```
CreateProcess( NULL, cmdbuf, NULL, NULL, bInH, dwCrtFlags, en-  
    vbuf, NULL, &siStartInfo, &prInfo  
  
    );
```

条件语句也应该在必要时换行：

```

if ( ch >= '0' || ch <= '9' || ch >= 'a' || ch <= 'z' ||
    ch >= 'A' || ch <= 'Z' )

```

iv) 空行。不要不加空行，空行可以区分不同的程序块，程序块间，最好加上空行。如：

```

HANDLE hProcess; PROCESS_T procInfo;

/* open the process handle */ if ((hProcess = OpenProcess (PROCESS_ALL_ACCESS,
FALSE, pid)) == NULL) { return LSE_MISC_SYS; }

memset(&procInfo, 0, sizeof(procInfo)); procInfo.idProc = pid;
procInfo.hdProc = hProcess; procInfo.misc |= MSCAVA_PROC;

return(0);

```

v) 对齐。用TAB键对齐你的一些变量的声明或注释，一样会让你的程序好看一些。如：

```

typedef struct _pt_man_t_ { int numProc; /* Number of processes / int max-
Proc; / Max Number of processes / int numEvt; / Number of events / int
maxEvt; / Max Number of events / HANDLE pHndEvt; /* Array of events
/ DWORD timeout; / Time out interval / HANDLE hPipe; / Namedpipe /
TCHAR usr[MAXUSR];/ User name of the process / int numMsg; / Number
of Message / int Msg[MAXMSG];/ Space for intro process communicate */

} PT_MAN_T;

```

怎么样？感觉不错吧。

这里主要讲述了如果写出让人赏心悦目的代码，好看的代码会让人的心情愉快，读起代码也就不累，工整、整洁