The strangest way of detecting Windows NT

devblogs.microsoft.com/oldnewthing/20041026-00

October 26, 2004

Raymond Chen

A colleague of mine nominated this code for Function of the Year. (This is the same person who was the first to report that a Windows beta used a suspect URL.) I have to admit that this code is pretty impressive. Of all the ways to check the operating system, you have to agree that sniffing at an undocumented implementation detail of memory-mapped files is certainly creative!

```
// following the typographical convention that code
// in italics is wrong
int AreWeRunningOnWindowsNT()
{
      HANDLE hFile, hFileMapping;
      BYTE *pbFile, *pbFile2;
      char szFile[MAX_PATH];


      GetSystemDirectory(szFile, MAX_PATH);
      strcat(szFile, “\\MAIN.CPL”);
      hFile = CreateFile(szFile, GENERIC_READ | GENERIC_WRITE, 0,
            NULL, OPEN_ALWAYS, FILE_ATTRIBUTE_NORMAL, NULL);


      hFileMapping = CreateFileMapping(hFile, NULL, PAGE_READWRITE,
            0, 0, NULL);


      pbFile = (PBYTE) MapViewOfFile(hFileMapping, FILE_MAP_WRITE,
            0, 0, 0);


      pbFile2 = (PBYTE) MapViewOfFile(hFileMapping, FILE_MAP_WRITE,
            0, 65536, 0);


      if (pbFile + 65536 != pbFile2)
            return 1;


      return 0;
}
```

Nevermind that the function also leaves a file locked and leaks two handles and two views each time you call it!

What’s more, this function may erroneously report `FALSE` on a Windows NT machine if by an amazing coincidence the memory manager happens to assign the second file view to the very next 64K block of memory (which it is permitted to do since address space granularity is 64K).

It can also erroneously report `TRUE` on a Windows 95 machine if the `MAIN.CPL` file happens to be smaller than 64K, or if you don’t have write permission on the file. (Notice that the program requests read-write access to the `MAIN.CPL` file.)

This particular function is from a library that is used by many popular multimedia titles.

The quickest way to detect whether you are running on a Windows 95-series system or a Windows NT-series system is to use the hopefully-obviously-named function GetVersion.

```
int AreWeRunningOnWindowsNT()
{
    return (GetVersion() & 0x80000000) == 0;
}
```

[Raymond is currently on vacation; this message was pre-recorded.]

Raymond Chen

Follow