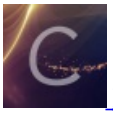
分享一个在内存里搜索QQ号码的源码，源自看雪论坛

原创

Witch_Soya 于 2012-06-13 21:19:49 发布 3700 收藏
分类专栏： LOVE QQ 文章标签： qq basic system query null c



源码出于看雪论坛。源址已佚。先谢过

看结果

耗时47微秒

```
#include <windows.h>
#include <tlhelp32.h>
#include <tchar.h>
#include <stdlib.h>
#include <stdio.h>

int SearchStr(PTSTR pszString, int iStrLen, PTSTR pszSearchStr)
{
    int i = 0;
    int iSearchStrlen = _tcslen(pszSearchStr);

    while ((i + iSearchStrlen) <= iStrLen )
    {
        int n = 0;
        //先匹配两个字串的第一个字符
        if (pszSearchStr[0] == pszString[i])
        {
            //若相等，则开始逐字符匹配
            for (int j = 0; j <iSearchStrlen; j++)
            {
                //相同位置字符匹配
                if (pszSearchStr[j] == pszString[i + j])
                {
                    //若相同位置字符匹配成功，则计数器加1
                    n++;
                }
                else  //相同位置字符匹配失败
                {
                    //源字符串位置跳过匹配相同的n个字符
                    i = i + n;
```

```
                    //跳出当前匹配循环，开始新位置的匹配
                    break;
                }
            }
            //若匹配成功，计数和目标字符串长度相等，则找到目标
            if (iSearchStrlen == n)
            {
                //i为找到的目标字符串在源字符串中的起点位置，
                //此处return，即找到的目标字符串首次出现位置
                return i;
                //找到一个目标，后移一位继续找，
                //如要继续找，请注意最后的return，代码需做点小修改
                //i++;
            }
        }
        else  //若两字符串的第一个字符不同
        {
            //开始反向找源字符串相对目标字符串的后一个字符是否在目标字符串内
            for (int j = iSearchStrlen - 1; j >= 0; j--)
            {
                //找到存在紧跟其后的那个字符
                if (pszSearchStr[j] == pszString[i + iSearchStrlen])
                {
                    //该字符出现在目标字符串中的位置
                    n = j;
                    //只需知道排在倒数第一那个位置，跳出循环开始移动位置
                    break;
                }
            }
            //移动到位置为(一个目标串长度减去出现目标串中匹配字符出现的位置)，
            //即如果出现该字符，则使相同的两个字符对齐，若未出现，直接移动一个目标串长度
            i = i + iSearchStrlen - n;
        }
    }
    return 0;
}

int ReadMem(DWORD dwPid)
{
    //要搜索的特征码
    TCHAR szSub[] = TEXT("index?uin=");
    //特征码出现的位置
    int iPos = 0;
    HANDLE hProcess = OpenProcess(PROCESS_QUERY_INFORMATION | PROCESS_VM_READ, 0, dwPid);
    if (hProcess == NULL)
    {
        return 0;
    }
    SYSTEM_INFO siSysInfo;
    GetSystemInfo(&siSysInfo);

    MEMORY_BASIC_INFORMATION mbi;
    DWORD pAddress = (DWORD)siSysInfo.lpMinimumApplicationAddress;

    int Count = GetTickCount();
    while (pAddress < (DWORD)siSysInfo.lpMaximumApplicationAddress)
    {
        if (VirtualQueryEx(hProcess, (LPVOID)pAddress, &mbi, sizeof(mbi)) != sizeof(mbi))
        {
            return 0;
```

```
            return 0;
        }

        if ((mbi.State == MEM_COMMIT) && (mbi.Protect == PAGE_READWRITE))
        {
            DWORD Base = pAddress;
            int ReadSize = mbi.RegionSize;

            if (ReadSize >= 1024)
            {
                DWORD dwBytes = 0;
                TCHAR *MemBuf = (TCHAR *)malloc(ReadSize * sizeof(TCHAR));

                if (ReadProcessMemory(hProcess, (LPCVOID)Base, MemBuf, ReadSize, &dwBytes))
                {
                    //开始搜索特征码
                    iPos = SearchStr(MemBuf, dwBytes, szSub);
                    if (iPos)
                    {
                        _tprintf(TEXT(">>>   Address: 0x%.8X\n"), Base + iPos * sizeof(TCHAR));
                        //指向QQ号码的第一个字符
                        TCHAR *ptsQQ = &MemBuf[iPos + _tcslen(szSub)];
                        _tprintf(TEXT(">>>        QQ: "));
                        //利用指针来打印出当前QQ进程的QQ号码，
                        //QQ号码之后的字符是'&'
                        for (;*ptsQQ != '&';*ptsQQ++)
                        {
                            //注意这里是循环打印出QQ号码的每个字符，而不是整个字符串
                            _tprintf(TEXT("%c"), *ptsQQ);
                        }
                        _tprintf(TEXT("\n"));
                        //找到1个就OK了，去除break可继续找
                        break;
                    }
                }
                free(MemBuf);
            }
        }
        //从下一块内存块继续找
        pAddress = (DWORD)mbi.BaseAddress + mbi.RegionSize;
    }
    //计算一下查找QQ号码的用时
    Count = GetTickCount() - Count;
    _tprintf(TEXT(">>>      Time: %d ms\n"), Count);
    return iPos;
}

DWORD FindByPID(PTSTR pszProcessName)
{
    DWORD dwProcessID = 0;
    HANDLE hProcessSnap;
    PROCESSENTRY32 pe32;

    hProcessSnap = CreateToolhelp32Snapshot(TH32CS_SNAPPROCESS, 0);
    if (hProcessSnap == INVALID_HANDLE_VALUE)
    {
        return 0;
    }
    pe32.dwSize = sizeof(PROCESSENTRY32);
    if(!Process32First(hProcessSnap, &pe32))
```

```
    {
        CloseHandle(hProcessSnap);
        return 0;
    }
    do
    {
        //找到QQ进程
        if (wcscmp(pszProcessName, pe32.szExeFile) == 0)
        {
            dwProcessID = pe32.th32ProcessID;
            wprintf(TEXT(">>> ------ PID = %d ------\n"), dwProcessID);
            //开始内存搜索
            ReadMem(dwProcessID);
            wprintf(TEXT(">>> ------------------------\n\n"), dwProcessID);
        }
    }
    //继续找下一个进程
    while(Process32Next(hProcessSnap, &pe32));
    CloseHandle(hProcessSnap);
    //如果存在QQ进程，此处return的是最后一个QQ进程的ID，
    //如果不在QQ进程，此处return的是dwProcessID的初始值0
    return dwProcessID;
}

int main()
{
    TCHAR pszP[] = TEXT("QQ.exe");
    DWORD dwPID = FindByPID(pszP);
    if (!dwPID)
    {
        wprintf(TEXT(">>> Do not found the QQ.exe\n"));
    }
    return 0;
}
```