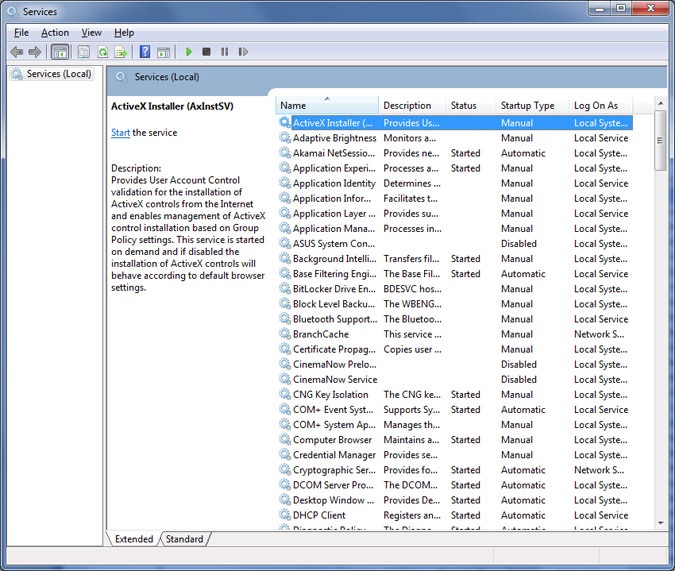
**用c写windows服务程序**

**一、前言**

最近公司需要在windows平台上做一个服务程序,不幸的是这个任务落在我这个不怎么会c/c++的人身上了,于是拿起一本«C语言入门到精通»就开始干了.在编码期间发现win32 API 有不少的坑.在此记录一下.

**二、上代码**

1. 安装服务,相当于在服务控制器中注册一个服务.



//判断是否已经安装过服务

BOOL IsInstalled() {

BOOL bResult = FALSE;

SC\_HANDLE hScm = OpenSCManager(NULL, NULL, SC\_MANAGER\_CREATE\_SERVICE);

if (hScm != NULL) {

SC\_HANDLE hService = OpenService(hScm, L"XXXX", SERVICE\_QUERY\_CONFIG);

if (hService != NULL) {

bResult = TRUE;

CloseServiceHandle(hService);

}

CloseServiceHandle(hScm);

}

return bResult;

}

//安装服务

BOOL InstallService() {

log\_i(\_T("安装服务中...\n"));

if (IsInstalled()) {

log\_i(\_T("服务已安装...\n"));

return FALSE;

}

//打开服务控制器

SC\_HANDLE hScm = OpenSCManager(NULL, NULL, SC\_MANAGER\_CREATE\_SERVICE);

if (hScm == NULL) {

log\_e(\_T("打开服务控制器失败...\n"));

return FALSE;

}

TCHAR \* path = GetFullPath();

SC\_HANDLE hService = CreateService(hScm,

L"TestService",

L"TestService",

SERVICE\_QUERY\_STATUS,

SERVICE\_WIN32\_OWN\_PROCESS,

SERVICE\_AUTO\_START,

SERVICE\_ERROR\_NORMAL,

path,

NULL,

NULL,

NULL,

NULL,

NULL);

if (hService == NULL){

log\_e(\_T("服务创建失败...\n"));

return FALSE;

}

//释放句柄

CloseServiceHandle(hScm);

CloseServiceHandle(hService);

return TRUE;

}

上面代码执行后,不出问题的话就可以在服务控制管理器看到自己创建的服务了.

* + 服务名称和二进制文件路径全部都要是宽字符,不能是char(窄字符)类型的,如果是char类型那么服务名称和二进制文件路径就会出现乱码的情况.使用L"XXXX"就表示宽字符,也可以用\_T("XXX") 来转换为宽字符

1. 运行服务,注册好服务以后,点击启动服务,windows 会调用特定函数进行调用.
2. SERVICE\_STATUS ServiceStatus; //服务状态
3. SERVICE\_STATUS\_HANDLE hStatus; //服务状态句柄
5. //服务入库函数
6. void WINAPI ServiceMain(DWORD argc, PWSTR\* argv) {
8. ServiceStatus.dwServiceType = SERVICE\_WIN32\_OWN\_PROCESS | SERVICE\_INTERACTIVE\_PROCESS;
9. ServiceStatus.dwCurrentState = SERVICE\_START\_PENDING;
10. ServiceStatus.dwControlsAccepted = SERVICE\_ACCEPT\_SHUTDOWN | SERVICE\_ACCEPT\_STOP;
11. ServiceStatus.dwWin32ExitCode = NO\_ERROR;
12. ServiceStatus.dwServiceSpecificExitCode = NO\_ERROR;
13. ServiceStatus.dwCheckPoint = 0;
14. ServiceStatus.dwWaitHint = 0;
16. hStatus = RegisterServiceCtrlHandler(ServiceName, ServiceCtrlHandler);
17. if (!hStatus)
18. {
19. DWORD dwError = GetLastError();
20. log\_e(\_T("启动服务失败!%d\n"), dwError);
21. return;
22. }
24. //设置服务状态
25. ServiceStatus.dwCurrentState = SERVICE\_RUNNING;
26. SetServiceStatus(hStatus, &ServiceStatus);
28. Run();
30. //停止服务
31. ServiceStatus.dwCurrentState = SERVICE\_STOP\_PENDING;
32. SetServiceStatus(hStatus, &ServiceStatus);
33. }
35. //服务回调
36. void WINAPI ServiceCtrlHandler(DWORD fdwControl)
37. {
38. switch (fdwControl) {
39. case SERVICE\_CONTROL\_STOP:
40. log\_i(\_T("服务停止...\n"));
41. ServiceStatus.dwCurrentState = SERVICE\_STOPPED;
42. ServiceStatus.dwWin32ExitCode = 0;
43. SetServiceStatus(hStatus, &ServiceStatus);
44. break;
45. case SERVICE\_CONTROL\_SHUTDOWN:
46. log\_i(\_T("服务终止...\n"));
47. ServiceStatus.dwCurrentState = SERVICE\_STOPPED;
48. ServiceStatus.dwWin32ExitCode = 0;
49. SetServiceStatus(hStatus, &ServiceStatus);
50. break;
51. default:
52. break;
53. }
54. }
56. //运行真正的程序逻辑diamante
57. void Run() {
58. while(TRUE){
59. printf("test\n");
60. }
61. }
63. //运行服务
64. void RunService() {
65. SERVICE\_TABLE\_ENTRY ServiceTable[2];
66. ServiceTable[0].lpServiceName = ServiceName;
67. ServiceTable[0].lpServiceProc = (LPSERVICE\_MAIN\_FUNCTION)ServiceMain;//函数指针
68. ServiceTable[1].lpServiceName = NULL;
69. ServiceTable[1].lpServiceProc = NULL;
70. StartServiceCtrlDispatcher(ServiceTable);
71. }
73. //程序主函数
74. int wmain(int argc, wchar\_t \*argv[]) {
75. //运行服务
76. RunService();
77. }
78. * 在windows 调用ServiceMain 这个函数的过程中,一定不要费时,也就是说中间不要写耗费时间的代码,或者出现导致程序退出的错误,否则服务启动不成功.
    * 如果程序访问了C盘的文件,要注意权限的问题.否则服务启动不成功.
    * 一定要先注册服务在启动服务,如果没有注册服务就执行RunService这个方法,程序会报错并退出,RunService这个方法一定要由windows进行调用,否则失败.
    * 不能调试.
79. 看完上面的两点,是不是觉得安装服务,和启动服务的代码不能写在一个程序里面呢?其实是可以的,通过程序的启动参数设置程序不同的行为,比如不带参数启动程序视为安装服务,带参数启动程序视为启动服务.

**三、总结**

不知道国内是资料少还是我关键词不对,在写的过程中遇到了很多的问题,还好都能通过google或者百度解决.以上就是我遇到的坑的总结,可能总结的不够全面,有些地方描述的不够准确,望指正!