Windows

#include <Windows.h>

CreateFileA()

Флаги доступа (dwDesiredAccess)

|  |
| --- |
| GENERIC\_READ |
| GENERIC\_WRITE |
| GENERIC\_READ | GENERIC\_WRITE |
| FILE\_SHARE\_DELETE |
| FILE\_SHARE\_READ |
| FILE\_SHARE\_WRITE |

Действие, выполняемое на файле или устройстве, которое существует или не существует. (dwCreationDisposition)

|  |
| --- |
| OPEN\_ALWAYS |
| CREATE\_ALWAYS |
| CREATE\_NEW |
| OPEN\_EXISTING |
| TRUNCATE\_EXISTING |

Атрибуты файла

|  |
| --- |
| FILE\_ATTRIBUTE\_NORMAL |
| FILE\_FLAG\_OVERLAPPED |

ReadFile();

WriteFile();

CloseHandle();

DeleteFile();

CopyFile();

FindFirstFile();

FindNextFile();

GetFileSize();

STARTUPINFO si;

PROCESS\_INFORMATION pi;

ZeroMemory();

CreateProcessA();

WaitForSingleObject();

GetExitCodeProcess();

CreateThread(); -создать

WaitForMultipleObjects(); -подождать

ExitThread(); - выйти из потока

ResumeThread(); -возобновить поток

TerminateThread(); - остановить другой поток

SuspendThread(); - приостановить поток

InitializeCriticalSection();

DeleteCriticalSection();

EnterCriticalSection();

LeaveCriticalSection();

\_\_declspec(dllimport)

\_\_declspec(dllexport)

typedef int(\_\_cdecl\* MyFunction)(LPWSTR);

LoadLibrary();

GetProcAddress();

FreeLibrary();

CreatePipe();

SECURITY\_ATTRIBUTES sa = { sizeof(sa),NULL,TRUE };

STARTUPINFO\* sti = calloc(1, sizeof(STARTUPINFO));

sti->cb = sizeof(STARTUPINFO);

PROCESS\_INFORMATION\* pi = calloc(1,sizeof(PROCESS\_INFORMATION));

#define PIPE\_NAME "\\\\.\\pipe\\MyPipe"

CreateNamedPipeA();

Флаги открытия(dwOpenMode)

|  |
| --- |
| PIPE\_ACCESS\_DUPLEX |
| PIPE\_ACCESS\_INBOUND |
| PIPE\_ACCESS\_OUTBOUND |

Флаги режима работы канала(dwPipeMode)

|  |
| --- |
| PIPE\_TYPE\_MESSAGE |
| PIPE\_READMODE\_MESSAGE |
| PIPE\_WAIT |

Режим ожидания входящих соединений к каналу(nMaxInstances)

|  |
| --- |
| PIPE\_UNLIMITED\_INSTANCES |

ConnectNamedPipe();

SetNamedPipeHandleState();

OpenClipboard();

ClipboardInputText();

GlobalAlloc();

GlobalUnlock();

EmptyClipboard();

SetClipboardData();

Форматы буфера обмена

|  |
| --- |
| CF\_UNICODETEXT |

CloseClipboard();

RegOpenKeyA();

RegSetValueExА();

RegSetKeyValueA();

RegGetValueA();

RegCloseKey();

|  |
| --- |
| HKEY\_CLASSES\_ROOT |
| HKEY\_CURRENT\_USER |
| HKEY\_LOCAL\_MACHINE |
| HKEY\_USERS |
| HKEY\_CURRENT\_CONFIG |

Типы параметров

|  |
| --- |
| REG\_SZ |
| REG\_BINARY |
| REG\_DWORD |
| REG\_QWORD |
| REG\_MULTI\_SZ |
| REG\_EXPAND\_SZ |

Флаги, ограничивающие тип данных значения для запроса.

|  |
| --- |
| RRF\_RT\_ANY |
| RRF\_RT\_DWORD |
| RRF\_RT\_QWORD |
| RRF\_RT\_REG\_BINARY |
| RRF\_RT\_REG\_DWORD |
| RRF\_RT\_REG\_EXPAND\_SZ |
| RRF\_RT\_REG\_MULTI\_SZ |
| RRF\_RT\_REG\_NONE |
| RRF\_RT\_REG\_QWORD |
| RRF\_RT\_REG\_SZ |

SetWindowsHookEx();

Тип установленной процедуры перехватчика(idHook).

|  |
| --- |
| WH\_KEYBOARD\_LL |
| WH\_MOUSE\_LL |

GetModuleHandle();

GetMessageW();

TranslateMessage();

GetKeyState();

GetKeyboardState();

DispatchMessage();

UnhookWindowsHookEx();

PKBDLLHOOKSTRUCT pHook = (PKBDLLHOOKSTRUCT)lParam;

|  |
| --- |
| WM\_KEYDOWN |
| WM\_SYSKEYDOWN |
| WM\_RBUTTONDOWN |
| WM\_MBUTTONDOWN |
| WM\_LBUTTONDOWN |

GetKeyboardLayout();

GetWindowThreadProcessId();

GetForegroundWindow();

MapVirtualKeyW();

GetKeyNameTextA();

CallNextHookEx();

Linux:

#include <sys/stat.h>

#include <fcntl.h>

#include <unistd.h>

open()

Значения флагов

|  |
| --- |
| O\_RDONLY |
| O\_WRONLY |
| O\_RDWR |
| O\_CLOEXEC |
| O\_CREAT |
| O\_DIRECTORY |
| O\_EXCL |
| O\_LARGEFILE |
| O\_NOFOLLOW |
| O\_TRUNC |
| O\_APPEND |
| O\_ASYNC |
| O\_DIRECT |
| O\_NOATIME |
| O\_NONBLOCK |
| O\_SYNC |

Значения констант mode:

|  |
| --- |
| S\_IRWXU (00700) |
| S\_IRUSR (00400) |
| S\_IWUSR (00200) |
| S\_IXUSR (00100) |
| S\_IRWXG (00070) |
| S\_IRGRP (00040) |
| S\_IWGRP (00020) |
| S\_IXGRP (00010) |
| S\_IRWXO (00007) |
| S\_IROTH (00004) |
| S\_IWOTH (00002) |
| S\_IXOTH (00001) |

read()

write();

close();

#include <signal.h>

signal();

Сигналы:

|  |
| --- |
| SIGINT |
| SIGTSTP |
| SIGQUIT |

Значения вместо адреса функции:

|  |
| --- |
| SIG\_DFL |
| SIG\_IGN |

kill();

raise();

killpg();

strsignal();

psignal();

#include <unistd.h> fff

fork();

execve();

getpid();

#include <sys/wait.h>

wait();

WEXITSTATUS();

waitpid();

#include <pthread.h>

pthread\_create();

pthread\_join();

pthread\_detach();

pthread\_mutex\_t mtx = PTHREAD\_MUTEX\_INITIALIZER

pthread\_mutex\_lock();

pthread\_mutex\_unlock();

pthread\_exit();

#include <errno.h>

perror();

Потоки:

-pthread

Программные модули:

-c

Статические библиотеки:

ar rc

ranlib

Разделяемые библиотеки:

-fPIC

-shared

#include <dlfcn.h>

dlopen();

Флаг для открытия библиотеки:

|  |
| --- |
| RTLD\_LAZY |
| RTLD\_NOW |
| RTLD\_GLOBAL |
| RTLD\_LOCAL |

dlsym();

dlclose();

dlerror();

-ldl

#include <unistd.h>

pipe();

#include <sys/stat.h>

#define PATH "/tmp/FIFOservertext"

remove();

mkfifo();

STDOUT\_FILENO.

#include <unistd.h>

daemon();

Доп. функции

#include <string.h>

strncpy();

strtok();

memcpy();

#include <stdio.h>

wprintf();

perror();

#include <stdlib.h>

atof()

atoi()