Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение   
высшего профессионального образования   
«Алтайский государственный технический университет им. И.И. Ползунова»

Факультет информационных технологий

Кафедра прикладной математики

Отчет защищен с оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Л. И. Сучкова

(подпись) (и.о., фамилия)

“\_\_\_\_”\_\_\_\_\_\_\_\_\_\_\_\_\_2014 г.

дата

Отчет по лабораторной работе №5

по дисциплине Периферийные устройства

Студент группы ПИ-21 С. Ю. Искуснов

и.о., фамилия

Преподаватель к.т.н., профессор Л. И. Сучкова\_

должность, ученое звание и.о., фамилия

Барнаул 2014

Задание:

Определить для файла, заданного именем, каталог, который является для него родительским. Если родительский каталог создан раньше указанной даты, то скопировать его содержимое в новый ката-лог.

Код программы:

.386

.model flat,stdcall

option casemap:none

include d:\masm32\include\windows.inc

include d:\masm32\include\user32.inc

include d:\masm32\include\kernel32.inc

include d:\masm32\include\oleaut32.inc

include d:\masm32\include\DateTime.inc

includelib d:\masm32\lib\user32.lib

includelib d:\masm32\lib\kernel32.lib

includelib d:\masm32\lib\oleaut32.lib

includelib d:\masm32\lib\datetime.lib

; //прототипы функций

WinMain proto :DWORD,:DWORD,:DWORD,:DWORD

WndProc proto :DWORD,:DWORD,:DWORD,:DWORD

CopyData proto :LPSTR, :LPSTR

MainFun proto :HWND, :LPSTR, :LPSTR, :DATETIME

GetParentDirectory proto :LPSTR, :LPSTR

m2m macro x,y

push y

pop x

endm

.data

ClassName db "Form1", 0 ;Название класса

AppName db "PL5", 0 ;Название приложения

hInstance HINSTANCE ? ;Дескриптор приложения

hEditSource HWND ?

hEditRecipient HWND ?

hEditDate HWND ?

CommandLine LPSTR ? ;Дескриптор консоли

buttonID equ 1000

exit\_ID equ 2000

help\_ID equ 2001

szListBox db "LISTBOX",0

szButton db "BUTTON",0

szEdit db "EDIT",0

szStatic db "STATIC",0

szMenu db "APP\_MENU",0

;Строки

buttonText1 db "Выполнить", 0

staticText1 db "Путь к файлу:", 0

staticText2 db "Скопировать в:", 0

staticText3 db "Дата:", 0

messageText1 db "Дата введена в неверном формате", 0

messageText2 db "Указанный путь к файлу несуществует", 0

messageText3 db "Не удалось создать окно",0

messageText4 db "Путь для копирования не существует",0

messageText5 db "Путь для копирования указывает на файл",0

messageText6 db "Обьект по указанному пути не является файлом",0

helpText db "Помощь",0

messageCaption1 db "Ошибка!",0

messageCaption2 db "Помощь",0

smask db "\\\*",0

slesh db '\\',0

currentdirectory db ".",0

parentdirectory db "..",0

newdirectory db "NewDirectory",0

buffer1 db 128 dup(0)

buffer2 db 128 dup(0)

buffer3 db 128 dup(0)

.code

start:

invoke GetModuleHandle, NULL ; //получение идентификатора процесса

mov hInstance, eax

invoke GetCommandLine ; //получение указателя командной строки

mov CommandLine, eax

invoke WinMain,hInstance,NULL,CommandLine,SW\_SHOWDEFAULT; //вызов основной функции

invoke ExitProcess,eax ; //выход из программы (с кодом возврата в eax)

WinMain proc hInst:HINSTANCE,hPrevInst:HINSTANCE,CmdLine:LPSTR,CmdShow:DWORD

LOCAL wc:WNDCLASSEX

LOCAL msg:MSG

LOCAL hWnd:HWND

;Заполнение стpуктуpы wc

mov wc.cbSize,SIZEOF WNDCLASSEX

mov wc.style, CS\_BYTEALIGNWINDOW

mov wc.lpfnWndProc, OFFSET WndProc

mov wc.cbClsExtra,NULL

mov wc.cbWndExtra,NULL

push hInstance

pop wc.hInstance

mov wc.hbrBackground,COLOR\_WINDOW

mov wc.lpszMenuName, offset szMenu

mov wc.lpszClassName,OFFSET ClassName

invoke LoadIcon,NULL,IDI\_APPLICATION

mov wc.hIcon,eax

mov wc.hIconSm,eax

invoke LoadCursor,NULL,IDC\_ARROW

mov wc.hCursor,eax

invoke RegisterClassEx, addr wc ; pегистpация нашего класса окна

invoke CreateWindowEx, NULL,

ADDR ClassName,

ADDR AppName,

WS\_OVERLAPPEDWINDOW, ;Стиль окна

CW\_USEDEFAULT,

CW\_USEDEFAULT,

CW\_USEDEFAULT,

CW\_USEDEFAULT,

NULL, ;Дескриптор родительского окна

NULL,

hInst, ;Дескриптор приложения

NULL

.if eax==NULL

invoke MessageBox, NULL, addr messageText3, addr messageCaption1, MB\_OK

ret

.endif

mov hWnd,eax

invoke ShowWindow, hWnd, SW\_SHOWNORMAL ; отобpазить окно

invoke UpdateWindow, hWnd ; обновить клиентскую область

.WHILE TRUE ;Цикл обработки сообщений

invoke GetMessage, ADDR msg,NULL,0,0

.BREAK .IF (!eax)

invoke TranslateMessage, ADDR msg

invoke DispatchMessage, ADDR msg

.ENDW

mov eax,msg.wParam ; сохpанение возвpащаемого значения в eax

ret

WinMain endp

WndProc proc hWnd:HWND, uMsg:UINT, wParam:WPARAM,lParam:LPARAM

local datetime:DATETIME

.if uMsg==WM\_DESTROY

invoke PostQuitMessage,NULL

.elseif uMsg == WM\_CREATE

;Поля редактирования

invoke CreateWindowEx,NULL,addr szEdit,NULL, WS\_BORDER or WS\_VISIBLE or WS\_CHILD,10,30,300,20,hWnd,NULL,hInstance,NULL

mov hEditSource,eax

invoke CreateWindowEx,NULL,addr szEdit,NULL, WS\_BORDER or WS\_VISIBLE or WS\_CHILD,10,70,300,20,hWnd,NULL,hInstance,NULL

mov hEditRecipient,eax

invoke CreateWindowEx,NULL,addr szEdit,NULL, WS\_BORDER or WS\_VISIBLE or WS\_CHILD,10,110,300,20,hWnd,NULL,hInstance,NULL

mov hEditDate,eax

;Кнопки

invoke CreateWindowEx,0, addr szButton,addr buttonText1,WS\_BORDER or WS\_CHILD or WS\_VISIBLE,10,140,80,20,hWnd,buttonID,hInstance,0

;Надписи

invoke CreateWindowEx,NULL,addr szStatic,addr staticText1, WS\_VISIBLE or WS\_CHILD,10,10,300,20,hWnd,NULL,hInstance,NULL

invoke CreateWindowEx,NULL,addr szStatic,addr staticText2, WS\_VISIBLE or WS\_CHILD,10,50,300,20,hWnd,NULL,hInstance,NULL

invoke CreateWindowEx,NULL,addr szStatic,addr staticText3, WS\_VISIBLE or WS\_CHILD,10,90,300,20,hWnd,NULL,hInstance,NULL

.elseif uMsg==WM\_COMMAND

.if wParam == buttonID

invoke GetWindowText, hEditSource, addr buffer1, 128

invoke GetWindowText, hEditRecipient, addr buffer2, 128

invoke GetWindowText, hEditDate, addr buffer3, 128

invoke StringToDateTime, addr buffer3, addr datetime

.if eax!=NULL

invoke MessageBox, hWnd, addr messageText1, addr messageCaption1, MB\_OK

xor eax,eax

ret

.endif

invoke MainFun, hWnd, addr buffer1, addr buffer2, datetime

.elseif wParam == exit\_ID

invoke PostQuitMessage,NULL

.elseif wParam == help\_ID

invoke MessageBox, hWnd, addr helpText, addr messageCaption2, MB\_OK

.endif

.else

invoke DefWindowProc,hWnd,uMsg,wParam,lParam

ret

.endif

xor eax,eax

ret

WndProc endp

MainFun proc hWnd:HWND, lpSource:LPSTR, lpRecipient:LPSTR, date:DATETIME

local systemtime:SYSTEMTIME

local datetime:DATETIME

local FindFileData:WIN32\_FIND\_DATA

invoke CreateFile, lpSource, GENERIC\_READ, NULL, NULL, OPEN\_EXISTING, FILE\_ATTRIBUTE\_NORMAL, NULL

.if eax==NULL

invoke MessageBox, hWnd, addr messageText2, addr messageCaption1, MB\_OK

xor eax, eax

ret

.endif

invoke CloseHandle,eax

invoke GetFileAttributes, lpSource

and eax,FILE\_ATTRIBUTE\_DIRECTORY

.if eax!=NULL

invoke MessageBox, hWnd, addr messageText6, addr messageCaption1, MB\_OK

xor eax, eax

ret

.endif

invoke CreateFile, lpRecipient, GENERIC\_READ, NULL, NULL, OPEN\_EXISTING, FILE\_ATTRIBUTE\_NORMAL, NULL

.if eax==NULL

invoke MessageBox, hWnd, addr messageText4, addr messageCaption1, MB\_OK

xor eax, eax

ret

.endif

invoke CloseHandle,eax

invoke GetFileAttributes, lpRecipient

and eax,FILE\_ATTRIBUTE\_DIRECTORY

.if eax==NULL

invoke MessageBox, hWnd, addr messageText5, addr messageCaption1, MB\_OK

xor eax, eax

ret

.endif

invoke GetParentDirectory, lpSource, addr buffer3

invoke FindFirstFile,addr buffer3,addr FindFileData

invoke FileTimeToSystemTime, addr FindFileData.ftCreationTime, addr systemtime

invoke TziDateToDateTime, addr systemtime, addr datetime

invoke CompareDateTime, addr date, addr datetime

.if eax==1

invoke lstrcat, lpRecipient, addr newdirectory

invoke CopyData, addr buffer3, lpRecipient

.endif

mov eax,TRUE

ret

MainFun endp

CopyData proc lpSource:LPSTR, lpRecipient:LPSTR

local FindFileData:WIN32\_FIND\_DATA

local h:HANDLE

local buffers[128]:BYTE

local bufferr[128]:BYTE

local maska[128]:BYTE

invoke CreateDirectory, lpRecipient, NULL

invoke lstrcpy,addr maska, lpSource

invoke lstrcat,addr maska, addr smask

invoke FindFirstFile,addr maska,addr FindFileData

mov h, eax

.if h==INVALID\_HANDLE\_VALUE

mov eax, FALSE

ret

.endif

.repeat

mov eax, FindFileData.dwFileAttributes

and eax, FILE\_ATTRIBUTE\_DIRECTORY

.if eax

push ebx

invoke lstrcmp, addr FindFileData.cFileName, addr currentdirectory

mov ebx, eax

invoke lstrcmp, addr FindFileData.cFileName, addr parentdirectory

and eax,ebx

pop ebx

.if eax!=0

invoke lstrcpy,addr buffers, lpSource

invoke lstrcat,addr buffers, addr slesh

invoke lstrcat,addr buffers, addr FindFileData.cFileName

invoke lstrcpy,addr bufferr, lpRecipient

invoke lstrcat,addr bufferr, addr slesh

invoke lstrcat,addr bufferr, addr FindFileData.cFileName

invoke CopyData,addr buffers, addr bufferr

.endif

.else

invoke lstrcpy,addr buffers, lpSource

invoke lstrcat,addr buffers, addr slesh

invoke lstrcat,addr buffers, addr FindFileData.cFileName

invoke lstrcpy,addr bufferr, lpRecipient

invoke lstrcat,addr bufferr, addr slesh

invoke lstrcat,addr bufferr, addr FindFileData.cFileName

invoke CopyFile,addr buffers, addr bufferr, FALSE

.endif

invoke FindNextFile,h,addr FindFileData

.until eax==0

invoke FindClose, h

mov eax,TRUE

ret

CopyData endp

GetParentDirectory proc lpDirectory:LPSTR, lpBuffer:LPSTR

pusha

mov ebx, lpDirectory

invoke lstrlen, lpDirectory

mov edi,eax

dec edi

mov dl,92

.while [ebx+edi]!=dl

dec edi

.endw

inc edi

invoke lstrcpyn, lpBuffer, lpDirectory, edi

popa

xor eax,eax

ret

GetParentDirectory endp

end start