Федеральное государственное бюджетное образовательное учреждение высшего

образования

«Чувашский государственный университет им. И.Н. Ульянова»

Кафедра вычислительной техники

# **Лабораторная работа 4.**

**СОЗДАНИЕ ДИНАМИЧЕСКИХ БИБЛИОТЕК**

Вариант 9

Выполнил: Иванов В.С.

студент группы ИВТ-41-22

Проверила:

Доцент Андреева А.А.

Чебоксары, 2024

Цель работы:

Изучить основные сведения об использовании DDL библиотек при их практическом использовании на языке ассемблера

Задание:

Разработать динамическую библиотеку, реализующую функции в соответствии с заданным вариантом, и программу для демонстрации ее возможностей. Использовать как явное, так и неявное связывание.

9. function PadCh(S: pchar; C: char; Len: byte): pchar. Возвращает строку, в которой S смещена влево, а остаток строки заполнен символами С. Для этого знаки С включаются справа от конца S до тех пор, пока общая длина строки не станет равной Len. Если S длиннее чем Len, то строка не изменяется. Если S - пустая строка, то возвращается строка из Len символов С.



Текст подпрограммы:  
.386

.model flat

public PadCh

.code

start@12:

mov al,1

ret 12

;function PadCh(S: string; C: char; Len: byte): string,

;Возвращает строку, в которой S смещена влево, а остаток строки заполнен символами С.

;Для этого знаки С включаются справа от конца S до тех пор пока общая длина строки не

;станет равной Len. Если S длиннее чем Len, то строка не изменяется.

;Если S пустая строка, то возвращается строка из Len символов С.

PadCh proc near

; адреса параметров в стеке:

S equ dword ptr [ebp+16] ; адрес строки S:string

Ch1 equ byte ptr [ebp+12] ; адрес параметра С :Char

Len equ byte ptr [ebp+8] ; адрес параметра Len:Byte

Res equ dword ptr [ebp+20] ; адрес строки результата

push ebp ; сохранение bp

mov ebp, esp ; настройка bp на вершину стека

;push ds ; сохранение ds

mov edi, [Res] ; es:di:=адрес результата

mov esi, [S] ; ds:si:=адрес исходной

; строки

mov ecx,0

cld ; очистка флага направления (инкремент)

lodsb ; al:=(ds:[si]), si:=si+1 (al - длина S)

stosb ; копируем длину строки S в Res

mov ah, al ; сохраняем длину строки S

cmp al, [Len] ; сравниваем длину S с Len

jae StoreLen ; если S >= Len, то копируем S и выходим

Pad:

mov al, [Len]

mov [edi-1], al ; Дополняем строку

mov cl, ah

rep movsb ; записать очередной символ результата Res

StoreLen:

mov cl, [Len]

sub cl, ah ; S длиннее, чем Len

mov al, [Ch1] ; добавляем очередной символ

;Копируем строку

rep stosb

Exit: ; pop ds ; восстановить ds

pop ebp ; восстановить bp

ret 16 ; выход с удалением параметров Ch1,

; Ch2 и адреса S (Res удалять нельзя!)

PadCh endp

end start@12  
  
Текст основной программы с неявным преобразованием:  
include include.inc

includelib 4\_kab\_d.lib

extrn PadCh: near ; Объявление внешней подпрограммы

.386

.model flat, stdcall

.const

title\_S\_before db 'Before neyav add:', 0

title\_S\_after db 'After add:', 0

.data

S db 5, 'qwert', 0 ; Исходная строка с длиной

Res db 255 dup(0) ; Буфер для результата, инициализированный нулями

Ch1 db 'A' ; Символ для добавления

Len1 db 5 ; Длина строки (должна соответствовать фактической длине)

.code

start:

; Вывод сообщения перед добавлением

call MessageBox, 0, offset S, offset title\_S\_before, MB\_OK

; Вызов вашей подпрограммы PadCh

push offset Res ; адрес результата

push offset S ; адрес строки S

mov al, [Ch1] ; символ для добавления

push eax ; символ

mov al, [Len1] ; длина строки

push eax ; длина

call PadCh ; вызов вашей подпрограммы

; Вывод сообщения после добавления

call MessageBox, 0, offset Res, offset title\_S\_after, MB\_OK

; Завершение программы

push 0 ; код завершения

call ExitProcess ; завершение программы

ends

end start  
  
Текст основной программы с явным преобразованием:  
include include.inc

.386

.model flat, stdcall

.const

title\_S\_before db 'Before yav add:', 0

title\_S\_after db 'After add:', 0

libr db '4\_kab\_d.dll',0

nameproc db 'PadCh',0

.data

S db 5, 'qwert', 0 ; Исходная строка с длиной

Res db 255 dup(0) ; Буфер для результата, инициализированный нулями

Ch1 db 'A' ; Символ для добавления

Len1 db 5 ; Длина строки (должна соответствовать фактической длине)

hlib dd ?

PadCh dd ?

.code

start:

call LoadLibrary,offset libr ;загрузка библиотеки

mov hlib,eax

call GetProcAddress,hlib,offset nameproc ;получение адреса функции

mov PadCh,eax

; Вывод сообщения перед добавлением

call MessageBox, 0, offset S, offset title\_S\_before, MB\_OK

; Вызов вашей подпрограммы PadCh

push offset Res ; адрес результата

push offset S ; адрес строки S

mov al, [Ch1] ; символ для добавления

push eax ; символ

mov al, [Len1] ; длина строки

push eax ; длина

call PadCh ; вызов вашей подпрограммы

; Вывод сообщения после добавления

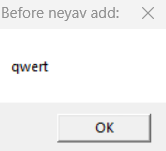
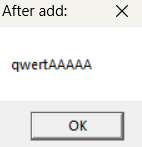
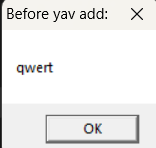
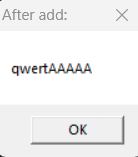
call MessageBox, 0, offset Res, offset title\_S\_after, MB\_OK

; Завершение программы

push 0 ; код завершения

call ExitProcess ; завершение программы

ends

end start  
  
Результаты кода:  
     
  
Вывод: изучил основные сведения об использовании DDL библиотек при их практическом использовании на языке ассемблера