Федеральное государственное бюджетное образовательное учреждение высшего

образования

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

Кафедра вычислительной техники

Лабораторная работа № 2

ОРГАНИЗАЦИЯ ВНЕШНИХ ПОДПРОГРАММ НА ЯЗЫКЕ АССЕМБЛЕРА

Вариант 9

Выполнил: Иванов В.С.

студент группы ИВТ-41-22

Проверила:

Доцент Андреева А.А.

Чебоксары, 2024

Цель работы: ознакомиться с организацией внешних подпрограмм на языке ассемблера.

Текст задания: 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.

Алгоритм:



Код подпрограммы:

code segment byte public

assume cs:code,ds:nothing

public PadCh

;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 far

; адреса параметров в стеке:

S equ dword ptr [bp+10] ; адрес строки S:string

Ch1 equ byte ptr [bp+8] ; адрес параметра С :Char

Len equ byte ptr [bp+6] ; адрес параметра Len:Byte

Res equ dword ptr [bp+14] ; адрес строки результата

push bp ; сохранение bp

mov bp, sp ; настройка bp на вершину стека

push ds ; сохранение ds

les di, [Res] ; es:di:=адрес результата

lds si, [S] ; ds:si:=адрес исходной

; строки

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 es: [di-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 bp ; восстановить bp

ret 12 ; выход с удалением параметров Ch1,

; Ch2 и адреса S (Res удалять нельзя!)

PadCh endp

code ends

end

**в указатель команд, в регистр кода и как изменится sp после ret 12 (SP, SS, IP, CS)**

SP - SP + 4 + 12 = SP + 16

SS – не изменяется

IP – M{SS:SP}

CS - M{SS:SP+2}

|  |  |  |
| --- | --- | --- |
| Команда | Стек | Адрес |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Код основной программы:

\_STACK segment para stack

db 1024 dup(?)

\_STACK ends

data segment

S db 5, 'qwert'

Res db 0, 255 dup('$')

Ch1 db 'A'

Len1 db 10

data ends

code segment byte public

assume cs:code, ds:data, ss:\_STACK

extrn PadCh:far

main:

;инициализация сегментного регистра данных

mov ax, data

mov ds, ax

push ds

mov ax, offset Res

push ax

push ds

mov ax, offset S

push ax

mov al, Ch1

push ax

mov al, [Len1]

push ax

call PadCh

mov dx, offset Res+1

mov ah, 9

int 21h

;завершение программы

mov ax, 4C00h

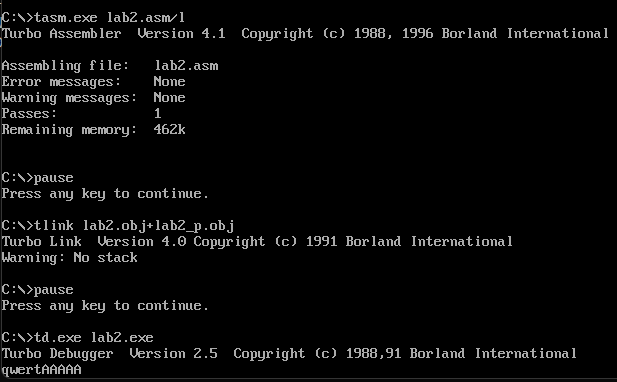
int 21h

code ends

end main

Содержимое стека после выполнения программы:



Результат выполнения работы:  


Вывод: ознакомился с организацией внешних подпрограмм на языке ассемблера.