**Федеральное государственное образовательное учреждение   
высшего образования**

**«ОРЛОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ   
ИМЕНИ И.С. ТУРГЕНЕВА»**

**Институт приборостроения, автоматизации и информационных технологий**

**Кафедра информационных систем и цифровых технологий**

Отчет к лабораторной работе № 6

«Обработка структур. Ведение базы данных о пациентах»

Выполнила:

Студентка группы 21ПГ

Банных Мария Алексеевна

Приняла:

доцент кафедры   
информационных систем   
и цифровых технологий  
Конюхова Оксана Владимировна  
  
доцент кафедры   
информационных систем   
и цифровых технологий  
Амелина Ольга Викторовна

Орёл 2022 год

Ответы на контрольные вопросы

1. *Структуры и определение шаблона структуры в программе.*

Cтруктура — это тип данных, состоящий из фиксированного числа элементов разного типа.

Шаблон структуры:

имя\_структуры STRUC

<описание полей>

имя\_структуры ENDS

Местоположение шаблона в программе может быть произвольным, но, следуя логике работы однопроходного транслятора, он должен быть расположен до того места, где определяется переменная с типом данной структуры. То есть при описании в сегменте данных переменной с типом некоторой структуры ее шаблон необходимо поместить в начале сегмента данных либо перед ним.

1. *Инициализация полей структуры в программе.*

<описание полей> представляет собой последовательность директив описания данных db, dw, dd, dq и dt.

Их операнды определяют размер полей и, при необходимости, начальные значения. Этими значениями будут, возможно, инициализироваться соответствующие поля при определении структуры.

Для использования описанной с помощью шаблона структуры в программе необходимо определить переменную с типом данной структуры. Для этого используется следующая синтаксическая конструкция:

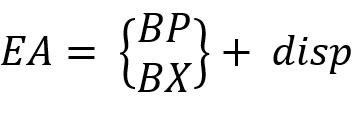
[имя переменной] имя\_структуры <[список значений]>

имя переменной – идентификатор переменной данного структурного типа.

Задание имени переменной необязательно. Если его не указать, будет просто выделена область памяти размером в сумму длин всех элементов структуры.

Список значений – заключенный в угловые скобки список начальных значений элементов структуры, разделенных запятыми. Его задание также необязательно. Если список указан не полностью, то все поля структуры для данной переменной инициализируются значениями из шаблона, если таковые заданы.

1. *Режимы адресации для доступа к элементам структуры, для доступа к элементам массива структур.*
2. Базовый. Применяется для доступа к структурам данных переменной длины. Тогда базовый адрес, определяющий начало набора элементов, хранится в базовом регистре, а смещение в команде определяет расстояние до определённого элемента.



Этот режим адресации удобно использовать для записей – структур данных, содержащих поля различной длины и, возможно, различных типов.

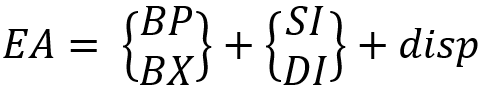
2.Базово-индексный. Используется для доступа к элементам массива, адресуемого указателем. Базовый адрес массива задаётся указателем базы (базовым регистром), а номер элемента массива – содержимым индексного регистра.



Если в bx содержится 100, а в si находится 52, то по адресу (смещению) 152 в сегменте данных находится искомое данное.

Такой режим адресации удобно использовать при работе со сложными структурами данных, поскольку он позволяет изменять две адресные компоненты.

3.Относительный базовый индексный. Используется для адресации элементов в указываемом массиве записей. Базовый адрес массива задаётся указателем базы, номер записи (т.е., элемента массива) определяется содержимым индексного регистра, а смещение в команде указывает расстояние до записи.



1. *Каким образом вычисляется расстояние до некоторого поля отдельного элемента массива структур?*

Идея введения структурного типа в любой язык программирования состоит в объединении разнотипных переменных в один объект.

В языке должны быть средства доступа к этим переменным внутри конкретного экземпляра структуры. Для того, чтобы сослаться в команде на поле некоторой структуры, используется специальный оператор - символ "." (точка). Он используется в следующей синтаксической конструкции:

адресное\_выражение.имя\_поле\_структуры

Адресное\_выражение - идентификатор переменной некоторого структурного типа или выражение в скобках в соответствии с указанными ниже синтаксическими правилами;

Имя\_поле\_структуры - имя поля из шаблона структуры

Это,на самом деле, тоже адрес, а точнее, смещение поля от начала структуры.

Таким образом оператор "." (точка) вычисляет выражение

(адресное\_выражение) + (имя\_поля\_структуры)

1. *Назначение оператора TYPE.*

Поле TYPE доопределяет тип дескриптора. Проверка типа позволяет обнаруживать ошибки при использовании сегмента не по назначению. Например, в сегмент кода нельзя записывать, а с сегментом таблицы LDT нельзя явно производить операции считывания, записи и выполнения.

Права, которыми может быть помечен несистемный дескриптор, ограничивают множество операций, которые можно производить с его сегментом. Права указываются также в поле TYPE несистемных дескрипторов. Например, исполняемый сегмент кода дополнительно может быть помечен как читаемый.

Практическая часть:

.model small ;создание структуры

Patients STRUC

number dw 0

gender db 0

birth dw 0

admission db 8 dup('/')

discharge db 8 dup('/')

Patients ENDS

data segment

massiv Patients <83,'m',1995,'11.11.22','23.11.22'>,<48,'w',2000,'10.09.22','22.09.22'>,<15,'m',2003,'22.09.22','29.09.22'>

date db '22.09.22'

nomer dw 15

year dw 1995

data ends

kod segment

assume DS: data, CS: kod

begin:

mov ax, data

mov ds, ax

xor ax, ax

mov bx, type Patients ;перемещение размера структуры в bx

;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

;КОЛИЧЕСТВО ПАЦИЕНТОВ, ПОСТУПИВШИХ НА КОНКРЕТНУЮ ДАТУ

mov cx, 3

lea di, massiv[0].admission

mov dx, 0 ;количество пациентов, поступивших 22.09.22

CALL sravnenie

;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

;КОЛИЧЕСТВО ПАЦИЕНТОВ-ЖЕНЩИН, ВЫПИСАННЫХ В КОНКРЕТНУЮ ДАТУ

mov cx, 3

lea di, massiv[0].gender

mov dx, 0 ;количество пациентов, выписавшхся 22.09.22

gender\_pr:

mov ah, 'w'

mov al, [di]

cmp ah, al

JNE vix

CALL sravn

vix:

add di, bx ;переход к началу даты в следующем сегменте структуры

LOOP gender\_pr

;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

;НАЙТИ ГОД РОЖДЕНИЯ ПАЦИЕНТА ПО НОМЕРУ МЕДКАРТЫ

mov cx, 3

lea di, massiv[0].number

mov dx, 0

medk:

mov ax, nomer

mov dx, [di]

cmp ax, dx

JNE konez

add di, 3

mov dx, [di] ;7D3(h) = 2003(d)

mov nomer, dx

sub di, 3

konez:

add di, bx ;переход к началу даты в следующем сегменте структуры

LOOP medk

;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

;НАЙТИ КОЛИЧЕСТВО ПАЦИЕНТОВ-МУЖЧИН ПО УКАЗАННОМУ ГОДУ РОЖДЕНИЯ

mov cx, 3

lea di, massiv[0].birth

lea si, massiv[0].gender

mov dx, 0

god\_r:

mov ax, year

mov dx, [di]

cmp ax, dx

JNE kz

mov al, 'm'

mov ah, [si]

JNE kz

mov dx, 1

push dx

kz:

add di, bx ;переход к началу даты в следующем сегменте структуры

add si, bx

LOOP god\_r

pop dx

;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

mov ah, 4ch

int 21h

sravnenie proc near

m1:

push cx

mov cx, 8 ;посимвольное сравнение строк из 8 символов, начала которых расположены в DI и SI

lea si, date ;для работы процедуры помещаем адреса начал сравниваемых

pov:

mov ah, [si]

mov al, [di]

cmp ah, al

JNE exit

inc si

inc di

LOOP pov

inc dx

sub di, 8

exit:

pop cx

add di, bx ;переход к началу даты в следующем сегменте структуры

LOOP m1

ret

sravnenie endp

sravn proc near

add di, 11

m2:

push cx

mov cx, 8 ;посимвольное сравнение строк из 8 символов, начала которых расположены в DI и SI

lea si, date ;для работы процедуры помещаем адреса начал сравниваемых

pvt:

mov ah, [si]

mov al, [di]

cmp ah, al

JNE vixod

inc si

inc di

LOOP pvt

inc dx

sub di, 8

vixod:

pop cx

sub di, 11

ret

sravn endp

kod ends

end begin