МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего профессионального образования

УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Кафедра «Вычислительная техника»

Дисциплина «МОП»

**Лабораторная работа №3**

Обработка чисел и структурированных данных.

**Вариант 5**

Выполнил

студент группы ИВТВМбд-31

Захарычев Н. А.

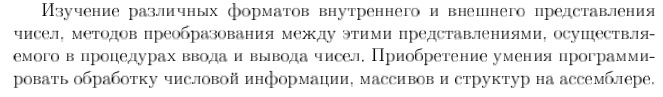
Проверил

ст. преподаватель кафедры ВТ

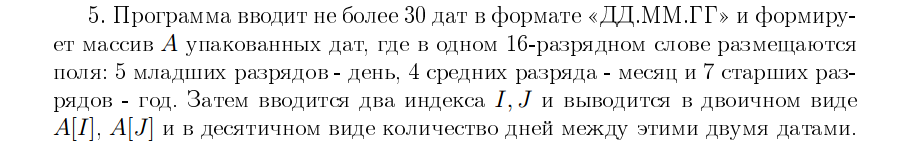
Лылова А. В.

**г.Ульяновск 2017**

**Цель работы:**



**Задание по варианту:**



**Ход работы:**

**Анализ исходных модулей лабораторной работы**

Назначения файлов:

*mopl3.asm –* основной код программы

*mopl3l.asm –* перечень процедур для обработки данных

*mopl3.inc –* перечень констант для работы программы

*mopl3.mac –* переченьмакросов

Назначение процедур файла *mopl3l.asm*

PUTSS – вывод на экран строки с задержкой по времени между символами

PUTC – вывод символа на терминал

GETCH – ввод символа с терминала

GETS – ввод строки в буфер

SLEN – определение длины строки

UBINARY – преобразование числа в беззнаковое, десятичное

DILAY – выполнение задержки при выполнении программы.

Назначение макросов файла *mopl3.mac*

PUTL – вывод строки на экран

PUTLS – вывод строки с задержкой

PUTLSC – вывод строки с задержкой и выравниванием по центру

FIXTIME – макрос фиксации начала измерения времени

DURAT – макрос окончания измерения времени и получения числа тактов

EXIT – завершение работы программы

PRINT – промежуточный вывод

PRINT\_DATE – вывод даты

PRINT\_DAYS – вывод итогового количества дней

**Представление объектов задачи в памяти**

**Структура даты:**

DATE STRUC

DAY DB 0 ;день

MONTH DB 0 ; месяц

YEAR DW 0 ; год

DATE ENDS

MONTHS DB 0, 31,28,31,30,31,30,31,31,30,31,30,31

DAYS\_COUNT DW 0 ; итоговое количество дней

DIFFER DW 0 ;разница между выбраннаи годами

I DB 0 ; индекс i

J DB 0 ; индекс j

NUM DB 0; количество

tmp DATE <> ;временная дата

ARR\_DATE DATE 10 DUP(<>) ; массив структуры для дат

FIRST DATE <> ;первая дата

SECOND DATE <> ;вторая дата

**Ввод дат и добавление в структуру:**

INPUT\_DATE:

;INPUT\_DATE

;-----------------------------------------

PUTL EMPTYS

PUTL STR\_TASK

PUTL EMPTYS

XOR BX, BX

XOR SI, SI

@INPUT\_CYCLE:

MOV DI, 10

XOR AX, AX

XOR CX, CX

MOV AH, 1

INT 21H

INC BX

CMP AX, 010DH

JE @NE

CMP AL, '.'

JE @ADD\_TOSTRCT

AND AX, 0FFh

SUB AL, 30H

MOV CL, AL

MOV AX, DX

MUL DI

MOV DX, AX

ADD DX, CX

CMP BX, 4

JE @ADD\_TOSTRCT

JMP @INPUT\_CYCLE

@ADD\_TOSTRCT:

MOV DI, WORD PTR NUM

MOV CX, DX

MOV AX, DI

MOV SP, 4

MUL SP

MOV DI, AX

CMP SI, 0

MOV BX, 0

JNE @MONTHS

MOV ARR\_DATE[DI].DAY, CL

MOV dl, ARR\_DATE[DI].DAY

INC SI

XOR DX, DX

XOR CX, CX

JMP @INPUT\_CYCLE

@MONTHS:

CMP SI, 1

JNE @YEAR

MOV ARR\_DATE[DI].MONTH, CL

INC SI

XOR DX, DX

XOR CX, CX

JMP @INPUT\_CYCLE

@YEAR:

MOV ARR\_DATE[DI].YEAR, CX

PUTL EMPTYS

ADD NUM, 1

XOR DX, DX

XOR CX, CX

MOV SI, 0

JMP @INPUT\_CYCLE

**Конверт в цифры и запись в индексы:**

@NE:

PUTL LINE\_SPACE

PUTL INDEX\_IJ

MOV BX, 2

@CONVERT\_CYCLE1:

XOR AX, AX

MOV AH, 1

INT 21H

SUB AL, 30H

CMP AL, 9H

JG @CYCLE\_OUTPUT1

CMP AL, 0H

JL @CYCLE\_OUTPUT1

MOV CL, AL

MOV AL, DL

MOV DL, 10

MUL DL

MOV DL, AL

ADD DL, CL

DEC BX

JNZ @CONVERT\_CYCLE1

@CYCLE\_OUTPUT1:

MOV I, DL

XOR DX, DX

MOV BX, 2

@CONVERT\_CYCLE2:

XOR AX, AX

MOV AH, 1

INT 21H

SUB AL, 30H

CMP AL, 9H

JG @CYCLE\_OUTPUT2

CMP AL, 0H

JL @CYCLE\_OUTPUT2

MOV CL, AL

MOV AL, DL

MOV DL, 10

MUL DL

MOV DL, AL

ADD DL, CL

DEC BX

JNZ @CONVERT\_CYCLE2

@CYCLE\_OUTPUT2:

MOV J, DL

**Определение «первой» даты и смена местами дат:**

;ОПРЕДЕЛЕНИЕ первой даты

MOV AX, SECOND.YEAR

MOV BX, FIRST.YEAR

CMP AX, BX

JG @SWAP

JNE @NEXT

MOV AH, SECOND.MONTH

MOV BH, FIRST.MONTH

CMP AH, BH

JG @SWAP

JNE @NEXT

MOV AH, SECOND.DAY

MOV BH, FIRST.DAY

CMP AH, BH

JG @SWAP

JNE @NEXT

JMP @EQUALS

;СМЕНА ДАТ

@SWAP:

MOV EAX, SECOND

MOV EBX, FIRST

MOV SECOND, EBX

MOV FIRST, EAX

**Подсчет разницы между годами:**

@NEXT:

MOV AX, FIRST.YEAR

MOV BX, SECOND.YEAR

SUB AX, BX

MOV DIFFER, AX

MOV AH, SECOND.MONTH

MOV BH, FIRST.MONTH

CMP AH, BH

JL @N

JNE @DEC

MOV AH, SECOND.DAY

MOV BH, FIRST.DAY

CMP AH, BH

JL @N

@DEC:

SUB DIFFER, 1

@N:

MOV BX, DIFFER

MOV AX, 365

MUL BX

MOV DAYS\_COUNT, AX

MOV CX, SECOND.YEAR

MOV AX, CX

MOV BX, 4

DIV BX

CMP DX, 0

JNE @M

MOV AL, SECOND.MONTH

CMP AL, 2

JG @M

ADD DAYS\_COUNT, 1

@M:

INC CX

@MLOOP:

MOV AX, CX

DIV BX

CMP DX, 0

JNE @NY

ADD DAYS\_COUNT, 1

@NY:

INC CX

CMP CX, FIRST.YEAR

JLE @MLOOP

MOV AX, FIRST.YEAR

DIV BX

CMP DX, 0

JNE @NM

MOV AL, FIRST.MONTH

CMP AL, 2

JL @NM

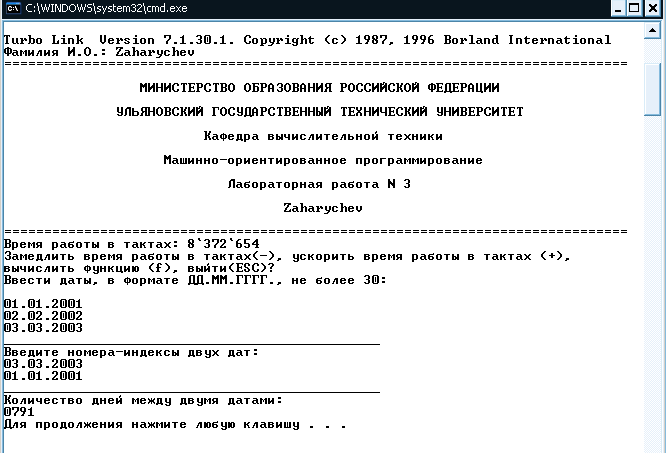
ADD DAYS\_COUNT, 1

@NM:

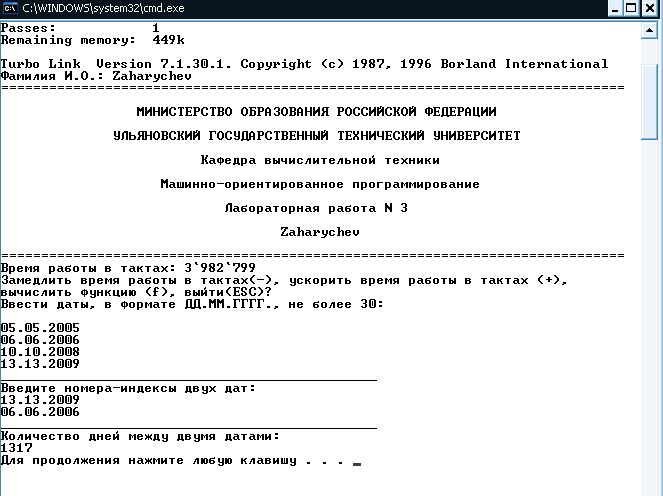
MOV AX, DAYS\_COUNT

**Контрольные примеры**

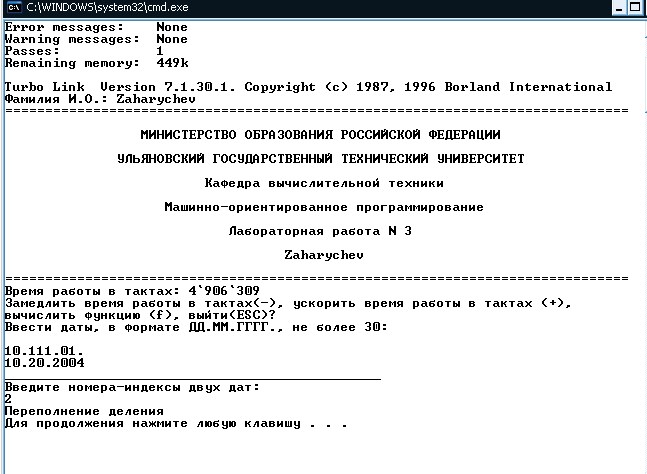
**Пример №1**



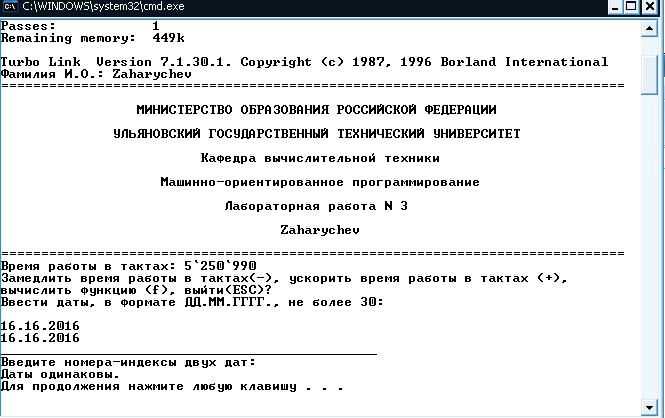
**Пример №2**



**Пример №3**



**Пример №4**



**Исходный код программы:**

mopl3.asm

*;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*; MOPL3.ASM - учебный пример для выполнения*

*; лабораторной работы N1 по машинно-ориентированному программированию*

*;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*.MODEL SMALL*

*.STACK 200h*

*.386*

*; Используются декларации констант и макросов*

*INCLUDE MOPL3.INC*

*INCLUDE MOPL3.MAC*

*; Декларации данных*

*.DATA*

*DATE STRUC*

*DAY DB 0*

*MONTH DB 0*

*YEAR DW 0*

*DATE ENDS*

*MONTHS DB 0, 31,28,31,30,31,30,31,31,30,31,30,31*

*DAYS\_COUNT DW 0*

*DIFFER DW 0 ;разница между годами*

*I DB 0*

*J DB 0*

*NUM DB 0; КОЛИЧЕТВО ДАТ*

*;даты*

*tmp DATE <> ;временная дата*

*ARR\_DATE DATE 10 DUP(<>)*

*FIRST DATE <> ;большая дата*

*SECOND DATE <> ;меньшая дата*

*SLINE DB 78 DUP (CHSEP), 0*

*REQ DB "Фамилия И.О.: ",0FFh*

*MINIS DB "МИНИСТЕРСТВО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ",0*

*ULSTU DB "УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ",0*

*DEPT DB "Кафедра вычислительной техники",0*

*MOP DB "Машинно-ориентированное программирование",0*

*LABR DB "Лабораторная работа N 3",0*

*REQ1 DB "Замедлить время работы в тактах(-), ускорить время работы в тактах (+),", 0*

*;------------- Новые переменные ------------------------------------------------------------------*

*REQ2 DB "вычислить функцию (f), выйти(ESC)?", 0FFh*

*;-------------------------------------------------------------------------------------------------*

*STR\_TASK DB "Ввести даты, в формате ДД.ММ.ГГГГ., не более 30:", 0*

*INDEX\_IJ DB "Введите номера-индексы двух дат: ", 0*

*LINE\_SPACE DB "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_", 0*

*DAY\_CNT DB "Количество дней между двумя датами: ", 0*

*SEP DB ".", 0FFh*

*EMPTYS DB 0*

*EQUAL DB "Даты одинаковы.", 0*

*TACTS DB "Время работы в тактах: ",0FFh*

*BUFLEN = 70*

*BUF DB BUFLEN*

*LENS DB ?*

*SNAME DB BUFLEN DUP (0)*

*BUF\_LEN\_NUMBER = 21*

*BUF\_NUMBER DB BUF\_LEN\_NUMBER*

*LEN\_NUMBER DB ?*

*W\_NUMBER DB BUF\_LEN\_NUMBER DUP (0)*

*PAUSE DW 0, 0 ; младшее и старшее слова задержки при выводе строки*

*TI DB LENNUM+LENNUM/2 DUP(?), 0 ; строка вывода числа тактов*

*; запас для разделительных "`"*

*;========================= Программа =========================*

*.CODE*

*; Макрос заполнения строки LINE от позиции POS содержимым CNT объектов,*

*; адресуемых адресом ADR при ширине поля вывода WFLD*

*BEGIN LABEL NEAR*

*; инициализация сегментного регистра*

*MOV AX, @DATA*

*MOV DS, AX*

*; инициализация задержки*

*MOV PAUSE, PAUSE\_L*

*MOV PAUSE+2,PAUSE\_H*

*PUTLS REQ ; запрос имени*

*; ввод имени*

*LEA DX, BUF*

*CALL GETS*

*@@L: ; циклический процесс повторения вывода заставки*

*; вывод заставки*

*; ИЗМЕРЕНИЕ ВРЕМЕНИ НАЧАТЬ ЗДЕСЬ*

*FIXTIME*

*PUTL EMPTYS*

*PUTL SLINE ; разделительная черта*

*PUTL EMPTYS*

*PUTLSC MINIS ; первая*

*PUTL EMPTYS*

*PUTLSC ULSTU ; и*

*PUTL EMPTYS*

*PUTLSC DEPT ; последующие*

*PUTL EMPTYS*

*PUTLSC MOP ; строки*

*PUTL EMPTYS*

*PUTLSC LABR ; заставки*

*PUTL EMPTYS*

*; приветствие*

*PUTLSC SNAME ; ФИО студента*

*PUTL EMPTYS*

*; разделительная черта*

*PUTL SLINE*

*; ИЗМЕРЕНИЕ ВРЕМЕНИ ЗАКОНЧИТЬ ЗДЕСЬ*

*DURAT ; подсчет затраченного времени*

*; Преобразование числа тиков в строку и вывод*

*LEA DI, TI*

*CALL UTOA10*

*PUTL TACTS*

*PUTL TI ; вывод числа тактов*

*; обработка команды*

*PUTL REQ1*

*;------Вывод своих строк с действиями -------------------*

*PUTL REQ2*

*;--------------------------------------------------------*

*CALL GETCH*

*CMP AL, 'f'*

*JNE PROC\_TIME*

*JMP INPUT\_DATE*

*INPUT\_DATE:*

*;INPUT\_DATE*

*;-----------------------------------------*

*PUTL EMPTYS*

*PUTL STR\_TASK*

*PUTL EMPTYS*

*XOR BX, BX*

*XOR SI, SI*

*@INPUT\_CYCLE:*

*MOV DI, 10*

*XOR AX, AX*

*XOR CX, CX*

*MOV AH, 1*

*INT 21H*

*INC BX*

*CMP AX, 010DH*

*JE @NE*

*CMP AL, '.'*

*JE @ADD\_TOSTRCT*

*AND AX, 0FFh*

*SUB AL, 30H*

*MOV CL, AL*

*MOV AX, DX*

*MUL DI*

*MOV DX, AX*

*ADD DX, CX*

*CMP BX, 4*

*JE @ADD\_TOSTRCT*

*JMP @INPUT\_CYCLE*

*@ADD\_TOSTRCT:*

*MOV DI, WORD PTR NUM*

*MOV CX, DX*

*MOV AX, DI*

*MOV SP, 4*

*MUL SP*

*MOV DI, AX*

*CMP SI, 0*

*MOV BX, 0*

*JNE @MONTHS*

*MOV ARR\_DATE[DI].DAY, CL*

*MOV dl, ARR\_DATE[DI].DAY*

*INC SI*

*XOR DX, DX*

*XOR CX, CX*

*JMP @INPUT\_CYCLE*

*@MONTHS:*

*CMP SI, 1*

*JNE @YEAR*

*MOV ARR\_DATE[DI].MONTH, CL*

*INC SI*

*XOR DX, DX*

*XOR CX, CX*

*JMP @INPUT\_CYCLE*

*@YEAR:*

*MOV ARR\_DATE[DI].YEAR, CX*

*PUTL EMPTYS*

*ADD NUM, 1*

*XOR DX, DX*

*XOR CX, CX*

*MOV SI, 0*

*JMP @INPUT\_CYCLE*

*;-----------------------------------------*

*@NE:*

*PUTL LINE\_SPACE*

*PUTL INDEX\_IJ*

*MOV BX, 2*

*@CONVERT\_CYCLE1:*

*XOR AX, AX*

*MOV AH, 1*

*INT 21H*

*SUB AL, 30H*

*CMP AL, 9H*

*JG @CYCLE\_OUTPUT1*

*CMP AL, 0H*

*JL @CYCLE\_OUTPUT1*

*MOV CL, AL*

*MOV AL, DL*

*MOV DL, 10*

*MUL DL*

*MOV DL, AL*

*ADD DL, CL*

*DEC BX*

*JNZ @CONVERT\_CYCLE1*

*@CYCLE\_OUTPUT1:*

*MOV I, DL*

*XOR DX, DX*

*MOV BX, 2*

*@CONVERT\_CYCLE2:*

*XOR AX, AX*

*MOV AH, 1*

*INT 21H*

*SUB AL, 30H*

*CMP AL, 9H*

*JG @CYCLE\_OUTPUT2*

*CMP AL, 0H*

*JL @CYCLE\_OUTPUT2*

*MOV CL, AL*

*MOV AL, DL*

*MOV DL, 10*

*MUL DL*

*MOV DL, AL*

*ADD DL, CL*

*DEC BX*

*JNZ @CONVERT\_CYCLE2*

*@CYCLE\_OUTPUT2:*

*MOV J, DL*

*;-------------------------------------------*

*MOV BX, 4H*

*MOV AL, I*

*DEC AL*

*AND AX, 00FFH*

*MUL BL*

*MOV SI, AX*

*MOV EAX, ARR\_DATE[SI]*

*MOV FIRST, EAX*

*MOV AL, J*

*DEC AL*

*AND AX, 00FFH*

*MUL BL*

*MOV SI, AX*

*MOV EAX, ARR\_DATE[SI]*

*MOV SECOND, EAX*

*;-------------------------------------------*

*;ОПРЕДЕЛЕНИЕ первой даты*

*MOV AX, SECOND.YEAR*

*MOV BX, FIRST.YEAR*

*CMP AX, BX*

*JG @SWAP*

*JNE @NEXT*

*MOV AH, SECOND.MONTH*

*MOV BH, FIRST.MONTH*

*CMP AH, BH*

*JG @SWAP*

*JNE @NEXT*

*MOV AH, SECOND.DAY*

*MOV BH, FIRST.DAY*

*CMP AH, BH*

*JG @SWAP*

*JNE @NEXT*

*JMP @EQUALS*

*;СМЕНА ДАТ*

*@SWAP:*

*MOV EAX, SECOND*

*MOV EBX, FIRST*

*MOV SECOND, EBX*

*MOV FIRST, EAX*

*@NEXT:*

*MOV AX, FIRST.YEAR*

*MOV BX, SECOND.YEAR*

*SUB AX, BX*

*MOV DIFFER, AX*

*MOV AH, SECOND.MONTH*

*MOV BH, FIRST.MONTH*

*CMP AH, BH*

*JL @N*

*JNE @DEC*

*MOV AH, SECOND.DAY*

*MOV BH, FIRST.DAY*

*CMP AH, BH*

*JL @N*

*@DEC:*

*SUB DIFFER, 1*

*@N:*

*MOV BX, DIFFER*

*MOV AX, 365*

*MUL BX*

*MOV DAYS\_COUNT, AX*

*MOV CX, SECOND.YEAR*

*MOV AX, CX*

*MOV BX, 4*

*DIV BX*

*CMP DX, 0*

*JNE @M*

*MOV AL, SECOND.MONTH*

*CMP AL, 2*

*JG @M*

*ADD DAYS\_COUNT, 1*

*@M:*

*INC CX*

*@MLOOP:*

*MOV AX, CX*

*DIV BX*

*CMP DX, 0*

*JNE @NY*

*ADD DAYS\_COUNT, 1*

*@NY:*

*INC CX*

*CMP CX, FIRST.YEAR*

*JLE @MLOOP*

*MOV AX, FIRST.YEAR*

*DIV BX*

*CMP DX, 0*

*JNE @NM*

*MOV AL, FIRST.MONTH*

*CMP AL, 2*

*JL @NM*

*ADD DAYS\_COUNT, 1*

*@NM:*

*MOV AX, DAYS\_COUNT*

*;------------------------------------------------------*

*;ПОДСЧЕТ МЕСЯЦЕВ*

*XOR BX, BX*

*MOV BH, FIRST.MONTH*

*MOV BL, SECOND.MONTH*

*CMP BH, BL*

*JNE @N1*

*MOV CL, FIRST.DAY*

*XOR BX, BX*

*MOV BL, SECOND.DAY*

*CMP CL, BL*

*JG @ND*

*ADD DAYS\_COUNT, 365*

*@ND:*

*AND BX, 00FFH*

*AND CX, 00FFH*

*MOV AX, DAYS\_COUNT*

*ADD AX, CX*

*SUB AX, BX*

*MOV DAYS\_COUNT, AX*

*JMP @END*

*@N1:*

*MOV SI, BX*

*SHR SI, 8H*

*AND BX, 00FFH*

*DEC SI*

*@LOOP:*

*CMP SI, BX*

*JE @N2*

*MOV DL, MONTHS[SI]*

*ADD AX, DX*

*DEC SI*

*CMP SI, 0*

*JG @NOSET12*

*MOV SI, 12*

*@NOSET12:*

*JMP @LOOP*

*@N2:*

*MOV DAYS\_COUNT, AX*

*XOR AX, AX*

*XOR BX, BX*

*XOR CX, CX*

*MOV CL, FIRST.MONTH*

*MOV BL, SECOND.MONTH*

*MOV SI, BX*

*MOV AL, MONTHS[SI]*

*SUB AL, SECOND.DAY*

*ADD DAYS\_COUNT, AX*

*MOV AL, FIRST.DAY*

*ADD DAYS\_COUNT, AX*

*@END:*

*XOR EAX, EAX*

*MOV AX, DAYS\_COUNT*

*PRINT\_DATE FIRST*

*PUTL EMPTYS*

*PRINT\_DATE SECOND*

*PUTL EMPTYS*

*PUTL LINE\_SPACE*

*PUTL DAY\_CNT*

*PRINT\_DAYS DAYS\_COUNT*

*PUTL EMPTYS*

*jmp @@E*

*@EQUALS:*

*PUTL EQUAL*

*jmp @@E*

*PROC\_TIME: CMP AL, '-' ; удлиннять задержку?*

*JNE CMINUS*

*INC PAUSE+2 ; добавить 65536 мкс*

*JMP @@L*

*CMINUS: CMP AL, '+' ; укорачивать задержку?*

*JNE CEXIT*

*CMP WORD PTR PAUSE+2, 0*

*JE BACK*

*DEC PAUSE+2 ; убавить 65536 мкс*

*BACK: JMP @@L*

*CEXIT: CMP AL, CHESC*

*JE @@E*

*TEST AL, AL*

*JNE BACK*

*CALL GETCH*

*JMP @@L*

*; Выход из программы*

*@@E: EXIT*

*EXTRN PUTSS: NEAR*

*EXTRN PUTC: NEAR*

*EXTRN GETCH: NEAR*

*EXTRN GETECH: NEAR*

*EXTRN GETS: NEAR*

*EXTRN SLEN: NEAR*

*EXTRN UTOA10: NEAR*

*END BEGIN*

Mopl3l.asm

*;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*; Процедуры для учебного примера лабораторной работы N 1 \**

*; по МОП. \**

*;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*.MODEL SMALL*

*.CODE*

*.386*

*INCLUDE MOPL2.INC*

*LOCALS*

*;=====================================================*

*; Подпрограмма вывода на экран строки, адресуемой SI,*

*; с задержкой времени между символами в <CX,DX> mcs.*

*; Завершителями строки являеются байты 0 или 0FFh.*

*; ЕСЛИ строка заканчивается байтом 0,*

*; ТО добавляется переход в начало новой строки*

*;*

*;=====================================================*

*PUTSS PROC NEAR*

*@@L: MOV AL, [SI]*

*CMP AL, 0FFH*

*JE @@R*

*CMP AL, 0*

*JZ @@E*

*CALL PUTC*

*INC SI*

*; CALL DILAY*

*JMP SHORT @@L*

*; Переход на следующую строку*

*@@E: MOV AL, CHCR*

*CALL PUTC*

*MOV AL, CHLF*

*CALL PUTC*

*@@R: RET*

*PUTSS ENDP*

*;==============================================*

*; Подпрограмма вывода AL на терминал*

*;==============================================*

*PUTC PROC NEAR*

*PUSH DX*

*MOV DL, AL*

*MOV AH, FUPUTC*

*INT DOSFU*

*POP DX*

*RET*

*PUTC ENDP*

*;==============================================*

*; Подпрограмма ввода символа в AL с терминала*

*;==============================================*

*GETCH PROC NEAR*

*MOV AH, FUGETCH*

*INT DOSFU*

*RET*

*GETCH ENDP*

*;=================================================*

*; Подпрограмма ввода строки в буфер, адресуемый DX*

*; и имеющий структуру:*

*; { char size; // размер буфера*

*; char len; // реально введено*

*; char str[size]; // символы строки }*

*;=================================================*

*GETS PROC NEAR*

*PUSH SI*

*MOV SI, DX*

*MOV AH, FUGETS*

*INT DOSFU*

*; прописать байт 0 в конец строки*

*XOR AH, AH*

*MOV AL, [SI+1]*

*ADD SI, AX*

*MOV BYTE PTR [SI+2], 0*

*POP SI*

*RET*

*GETS ENDP*

*;==============================================*

*; Подпрограмма подсчета числа символов в строке,*

*; адресуемой SI. Завершители строки: 0 и 0FFh*

*; Результат возвращается в AX*

*;==============================================*

*SLEN PROC NEAR*

*XOR AX, AX*

*LSLEN: CMP BYTE PTR [SI], 0*

*JE RSLEN*

*CMP BYTE PTR [SI], 0FFh*

*JE RSLEN*

*INC AX*

*INC SI*

*JMP SHORT LSLEN*

*RSLEN: RET*

*SLEN ENDP*

*;====================================================*

*; Подпрограмма преобразования <EDX,EAX> в беззнаковое*

*; десятичное, размещаемое по адресу DI*

*;==============================================*

*.DATA*

*UBINARY DQ 0 ; Исходное двоичное 64-разрядное*

*UPACK DT 0 ; Упакованные 18 десятичных цифр*

*.CODE*

*UTOA10 PROC NEAR*

*PUSH CX*

*PUSH DI*

*MOV DWORD PTR [UBINARY], EAX*

*MOV DWORD PTR [UBINARY+4], EDX*

*FINIT ; инициализация сопроцессора*

*FILD UBINARY ; забрасывание в него бинарного*

*FBSTP UPACK ; извлечение упаковонного десятичного*

*MOV CX, LENPACK ; получено 9 пар цифр*

*PUSH DS ; писать*

*POP ES ; будем*

*CLD ; через stosw*

*LEA SI, UPACK ; с конца*

*ADD SI, LENPACK ; буфера upack*

*; Цикл преобразования пар полубайтов в ASCII-коды цифр*

*@@L: XOR AX, AX*

*DEC SI*

*MOV AL, [SI]*

*SHL AX, 4*

*SHR AL, 4*

*ADD AX, 3030h*

*XCHG AL, AH*

*STOSW*

*LOOP @@L*

*; Фиксация конца строки*

*XOR AL, AL*

*STOSB*

*; Улучшим читабельность слишком длинного числа*

*CLD*

*MOV AX, LENNUM-4*

*@@L1: MOV CX, AX*

*POP DI ; встаем на начало строки*

*PUSH DI*

*MOV SI, DI*

*INC SI*

*REP MOVSB*

*MOV BYTE PTR [DI], CHCOMMA ; вставить разделитель троек цифр*

*SUB AX, 4 ; 3 цифры + разделитель обработаны*

*JS @@E ; прекратить, если осталось не больше 3-х цифр*

*JMP SHORT @@L1*

*@@E: POP SI*

*PUSH SI*

*XOR CX, CX*

*; Съедаем первые нули*

*; сначала подсчитываем*

*@@L2: CMP BYTE PTR [SI], '0'*

*JE @@N*

*CMP BYTE PTR [SI], CHCOMMA*

*JNE @@N1*

*@@N: INC CX*

*INC SI*

*JMP SHORT @@L2*

*@@N1: ; а теперь съедаем*

*POP DI*

*SUB CX, LENNUM+1*

*NEG CX*

*REP MOVSB*

*POP CX*

*RET*

*UTOA10 ENDP*

*;==============================================*

*; Подпрограмма задержки выполнения программы*

*; на <CX,DX> микросекунд*

*;==============================================*

*DILAY PROC NEAR*

*MOV AH, 86h*

*INT 15h*

*RET*

*DILAY ENDP*

*HEX2DEC PROC NEAR*

*MOV CX,0*

*MOV BX,10*

*LOOP1: MOV DX,0*

*DIV BX*

*ADD DL,30H*

*PUSH DX*

*INC CX*

*CMP AX,9*

*JG LOOP1*

*ADD AL,30H*

*MOV [SI],AL*

*LOOP2: POP AX*

*INC SI*

*MOV [SI],AL*

*LOOP LOOP2*

*RET*

*HEX2DEC ENDP*

*PUBLIC PUTSS, PUTC, GETCH, GETS, DILAY, SLEN, UTOA10, HEX2DEC*

*END*

Mopl3.inc

*;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*; Константы для учебного примера лабораторной работы N 1 \**

*; по МОП. \**

*;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*CHESC = 1Bh*

*CHCR = 13*

*CHSEP = '='*

*CHCOMMA = '`'*

*CHLF = 10*

*DOSFU = 21h*

*FUPUTC = 2*

*FUGETCH = 8*

*FUGETS = 0Ah*

*FUEXIT = 4Ch*

*PAUSE\_L = 0000 ; Младшее слово стартового значения паузы*

*PAUSE\_H = 0 ; Старшее слово*

*LENNUM = 18 ; Число символов в строке десятичного числа*

*LENPACK = 9 ; Упаковонное в два раза короче*

Mopl3.mac

*; Макрос вывода строки LINE*

*PUTL MACRO LINE*

*LEA SI, LINE*

*XOR CX, CX*

*XOR DX, DX*

*XOR AX, AX*

*CALL PUTSS*

*ENDM*

*; Макрос вывода строки LINE с задержкой PAUSE*

*PUTLS MACRO LINE*

*XOR AX, AX*

*LEA SI, LINE*

*; MOV DX, PAUSE*

*; MOV CX, PAUSE+2*

*CALL PUTSS*

*ENDM*

*; Как и PUTLS, но с центрированием посередине*

*PUTLSC MACRO LINE*

*LOCAL LO, NOMARG*

*LEA SI, LINE*

*CALL SLEN*

*MOV CX, 80*

*SUB CX, AX*

*SAR CX, 1*

*JZ NOMARG*

*LO: MOV AL, ' '*

*CALL PUTC*

*LOOP LO*

*NOMARG: PUTLS LINE*

*ENDM*

*PRINT\_DAYS MACRO NUM*

*PUSH EAX*

*PUSH EBX*

*XOR eax, eax*

*XOR ebx, ebx*

*MOV ax, NUM; В al кладём число x, которое нужно вывести*

*MOV bx,0A30h*

*DIV bh*

*MOV cl, ah*

*ADD cl, 30h*

*XOR ah,ah*

*DIV bh*

*ADD bl,ah*

*XOR ah,ah*

*DIV bh*

*MOV dx,3030h*

*ADD dx,ax*

*MOV ah,2*

*INT 21h*

*MOV dl,dh*

*INT 21h*

*MOV dl,bl*

*INT 21h*

*MOV dl,cl*

*INT 21h*

*POP EBX*

*POP EAX*

*ENDM*

*PRINT2 MACRO NUM*

*PUSH EAX*

*PUSH EBX*

*XOR eax, eax*

*XOR ebx, ebx*

*MOV al, NUM; В al кладём число x, которое нужно вывести*

*MOV bx,0A30h*

*XOR ah,ah*

*DIV bh*

*ADD bl,ah*

*XOR ah,ah*

*DIV bh*

*MOV dx,3030h*

*ADD dx,ax*

*MOV ah,2*

*MOV dl,dh*

*INT 21h*

*MOV dl,bl*

*INT 21h*

*POP EBX*

*POP EAX*

*ENDM*

*PRINT\_DATE MACRO DATE*

*PRINT2 DATE.DAY*

*PUTL SEP*

*PRINT2 DATE.MONTH*

*PUTL SEP*

*PRINT\_DAYS DATE.YEAR*

*ENDM*

*; Макрос фиксации начала измерения времени*

*; Заносит в стек младшее слово числа тактов системных часов*

*FIXTIME MACRO*

*DB 0Fh, 31h; Команда RDTSC чтения счетчика тактов*

*PUSH EDX*

*PUSH EAX*

*ENDM*

*; Макрос получения в <EDX,EAX> числа тактов, прошедших с последнего*

*; момента фиксации времени. Извлекает продукт FIXTIME из стека*

*DURAT MACRO*

*DB 0Fh, 31h; Команда RDTSC чтения счетчика тактов*

*POP EBX*

*SUB EAX, EBX*

*POP EBX*

*SBB EDX, EBX*

*ENDM*

*; Макрос завершения программы*

*EXIT MACRO*

*MOV AH, FUEXIT*

*MOV AL, 0*

*INT DOSFU*

*ENDM*