|  |  |  |
| --- | --- | --- |
| Санкт-Петербургский политехнический университет Петра Великого  Институт компьютерных наук и технологий  Высшая школа программной инженерии | | |
| Курсовая работа | | |
|  | | |
| по дисциплине «Архитектура ЭВМ» | | |
| Выполнил | | |
| студент гр. в3530904/00022 |  | В.Я. Копылов |
|  | | |
| Руководитель |  | С.А. Молодяков |
| «10» декабря 2021 г. | | |
| Санкт-Петербург | | |
| 2021 | | |

Оглавление

Введение 3

Блок-схема 4

Список прерываний 5

Код программы 6

# **Введение**

Ассемблер - низкоуровневый машинно-ориентированный язык программирования. Реализация языка зависит от типа процессора и определяется архитектурой вычислительной системы. Ассемблер позволяет напрямую работать с аппаратурой компьютера. Программа на языке ассемблера включает в себя набор команд, которые после трансляции преобразуются в машинные команды.

Программа 1

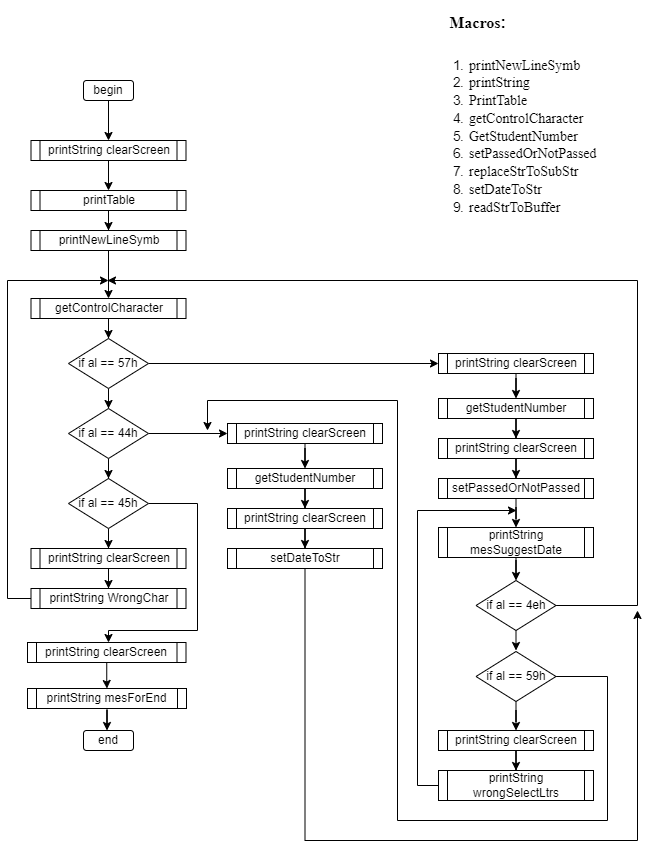
Вариант 11. Составить электронную зачетную ведомость для вашей группы по ТТО:

----------------------------------------------------------------------------------------------------

N | Фамилия И. О. | Номер зачетной книжки | Отметка о зачете | дата п/п

----------------------------------------------------------------------------------------------------

Предусмотреть возможность включения слова «зачет» и даты получения зачета. **Блок-схема**

**Список прерываний**

|  |  |  |
| --- | --- | --- |
| Interrupt | AH | Description |
| **16H** | 00h | get keystroke from keyboard (no echo) |
| **21H** | 01h | read character from standard input, with echo, result is stored in AL. if there is no character in the keyboard buffer, the function waits until any key is pressed |
| **21H** | 02h | write character to standard output. entry: DL = character to write, after execution AL = DL |
| **21H** | 09h | output of a string at DS:DX. String must be terminated by '$'. |
| **21H** | 0ah | input of a string to DS:DX, fist byte is buffer size, second byte is number of chars actually read. this function does not add '$' in the end of string. |
| **21H** | 4ch | return control to the operating system (stop program). |

**Текст программы**

DOSSEG

.MODEL small

.STACK 100H

.DATA

*; повсеместно в программе*

    clearScreen     DB 50 dup(0ah), '$'

*; главный цикл*

    mesForUser      DB "Please press:", 0ah,

            "1) W - to select a student for passed/not passed", 0ah,

            "2) D - for write date", 0ah,

            "3) E - for exit", 0ah, '$'

    wrongChar       DB "You entered an invalid check symbol, please try again!", 0ah, 0ah,

        "For continue press any button!", 0ah, '$'

    mesForEnd       DB "Good bye!", 0ah, '$'

*; выбор студента*

    mesSelectStudId DB "enter the student's serial number, like 1, 2, ..., 16.", 0ah,

        "Students count - 16", 0ah, '$'

    mesShowStudent  DB "For this student:", 0ah, '$'

    mesSelPassOrNot DB "Input one of these digits:", 0ah,

            "1) Student passed", 0ah,

            "2) Student not passed", 0ah, '$'

    wrongStudCount  DB "You entered the wrong student number!",0ah,

        "Maximum number 16, try again!", 0ah, 0ah, 0ah,'$'

    wrongSelect     DB "You write wrong digit! Try again!", 0ah, '$'

    wrongSelectLtrs DB "You write wrong letter! Try again!", 0ah, '$'

    passed          DB "    passed"

    notPassed       DB "not passed"

    mesSuggestDate  DB "Are you wanted to add date? input letter (Y or N)", 0ah,

        "Y) Yes", 0ah,

        "N) No", 0ah, '$'

*; ввод даты*

    mesInputDate    DB "Input date in format 'dd.mm.yy'",  0ah,'$'

*; таблица*

    border          DB 52 dup('-'), 0ah, '$'

    columns         DB "N |     Name      | grade Book |   mark   |  date  |", 0ah, '$'

    stud01          DB "01 Aksenov F.V.         3124563",' ', 10 dup('x'), ' ', 8 dup('x'), 0ah, '$'

    stud02          DB "02 Varakasa G.A.        1312456",' ', 10 dup('x'), ' ', 8 dup('x'), 0ah, '$'

    stud03          DB "03 Dvorianinov S.A.     4321231",' ', 10 dup('x'), ' ', 8 dup('x'), 0ah, '$'

    stud04          DB "04 Erin I.A.            7412456",' ', 10 dup('x'), ' ', 8 dup('x'), 0ah, '$'

    stud05          DB "05 Ivanov A.E.          1813575",' ', 10 dup('x'), ' ', 8 dup('x'), 0ah, '$'

    stud06          DB "06 Ivanushkin O.V.      4210684",' ', 10 dup('x'), ' ', 8 dup('x'), 0ah, '$'

    stud07          DB "07 Karaulov V.G.        3115393",' ', 10 dup('x'), ' ', 8 dup('x'), 0ah, '$'

    stud08          DB "08 Kolosov D.A.         0519102",' ', 10 dup('x'), ' ', 8 dup('x'), 0ah, '$'

    stud09          DB "09 Kopylov V.Y.         5217251",' ', 10 dup('x'), ' ', 8 dup('x'), 0ah, '$'

    stud10          DB "10 Kurnikov D.K.        1311251",' ', 10 dup('x'), ' ', 8 dup('x'), 0ah, '$'

    stud11          DB "11 Melnik E.V.          3217251",' ', 10 dup('x'), ' ', 8 dup('x'), 0ah, '$'

    stud12          DB "12 Milovskiy D.A.       2511251",' ', 10 dup('x'), ' ', 8 dup('x'), 0ah, '$'

    stud13          DB "13 Mosina A.E.          4613651",' ', 10 dup('x'), ' ', 8 dup('x'), 0ah, '$'

    stud14          DB "14 Nazarov V.A.         5712700",' ', 10 dup('x'), ' ', 8 dup('x'), 0ah, '$'

    stud15          DB "15 Novickiy I.V.        6213450",' ', 10 dup('x'), ' ', 8 dup('x'), 0ah, '$'

    stud16          DB "16 Noskova E.A.         7224211",' ', 10 dup('x'), ' ', 8 dup('x'), 0ah, '$'

    allStudentsStr  DW  offset stud01, offset stud02, offset stud03,

                        offset stud04, offset stud05, offset stud06,

                        offset stud07, offset stud08, offset stud09,

                        offset stud10, offset stud11, offset stud12,

                        offset stud13, offset stud14, offset stud15,

                        offset stud16

*; буферные переменные*

    buffer          DB 9, ?, 9 dup(0)

    studId          DW 16

*; ==== MACROS ====*

    printNewLineSymb macro

*; печать \n*

        pusha

        mov dl, 10

        mov ah, 02h

        int 21h

        popa

    endm

    printString macro str

*; печать переданной строки*

        pusha

        mov ah, 09h

        mov dx, offset str

        int 21hы

        popa

    endm

    printTable macro

*; печать всей таблицы*

*; cx - кол-во студентов*

*; bx - массив строк их студентов*

        pusha

        printString border

        printString columns

        printString border

        mov bx, offset allStudentsStr

        mov cx, 16

        @tableLoop:

            mov dx, [bx]

            mov ah, 09h

            int 21h

            add bx, 2

        loop @tableLoop

        printString border

        popa

    endm

    getControlCharacter macro

*; считывает символ пользователя, в зависимости от символа исполняются 3 разные метки*

*; 4 метка - неправильный символ, возврат в цикл*

        pusha

        mov ax, 0

        mov ah, 00h

        int 16h

        cmp al, 57h *; 45h - char W*

        je setPassetOrNotPassed

        cmp al, 44h *; 45h - char D*

        je setDate

        cmp al, 45h *; 45h - char E*

        je exit

        printString clearScreen

        printString wrongChar

        mov ax, 0

        mov ah, 00h

        int 16h

        popa

        jmp programmLoop

    endm

    getStudentNumber macro

*; получает число, означающее смещение в массиве от 0*

*; тем самым получает того студента, отметку или время которого нужно изменить*

        LOCAL start ,loopLocal, endLocal, calculateOffset, NumError

*; получает номер студента, которому будет поставлено зачет/незачет*

        pusha

        start:

            printString mesSelectStudId

*; --- считывание числа ---*

            mov ah,01h

            int 21h

            sub al, 30h

            mov ah, 0

            mov bx, 10

            mov cx, ax

            loopLocal:

                mov ah, 01h

                int 21h

                cmp al, 0dh *; сравнение с enter*

                je endLocal

                sub al, 30h

                cbw         *; расширение до слова*

                xchg ax, cx *; теперь в ax - предыдущее число, в cx - следующая*

                mul bx

                add cx, ax

                jmp loopLocal

            endLocal:

                xor ax, ax

*; --- проверка, что введенное число в рамках*

*; от 1 до 16 (т.е. в пределах студентов) ---*

                cmp cx, 16

                ja NumError

                cmp cx, 0

                je NumError

                jmp calculateOffset

                NumError:

                    printString clearScreen

                    printString wrongStudCount

                    jmp start

                calculateOffset:

                    add ax, 2

                    loop calculateOffset

                sub ax, 2

                mov [studId], ax

                popa

    endm

    setPassedOrNotPassed macro

        LOCAL start, errorSelect, studPassed, studNotPassed, errorDateSelect, addDateCase, notAddDateCase

*; выбирает сдал ли студент или нет и передает управление дальнейшей макро-подстановке*

        pusha

        start:

            mov bx, offset allStudentsStr

            add bx, [studId]

            printString mesShowStudent

            mov dx, [bx]

            mov ah, 09h

            int 21h

            printNewLineSymb

            printString mesSelPassOrNot

            mov ax, 0

            mov ah, 00h

            int 16h

            cmp al, 31h *; 45h - char 1*

            je studPassed

            cmp al, 32h *; 45h - char 2*

            je studNotPassed

            jmp errorSelect

            errorSelect:

                printString clearScreen

                printString wrongSelect

                printNewLineSymb

                jmp start

            studPassed:

                replaceStrToSubStr [bx], 32, Passed, 10

                jmp endStep

            studNotPassed:

                replaceStrToSubStr [bx], 32, notPassed, 10

                jmp endStep

            endStep:

*; стоит ли добавить дату экзамена?*

                printString clearScreen

                addOrNotDate:

                    printString mesShowStudent

                    mov dx, [bx]

                    mov ah, 09h

                    int 21h

                    printNewLineSymb

                    printString mesSuggestDate

                    mov ax, 0

                    mov ah, 00h

                    int 16h

                    cmp al, 59h *; 45h - char Y*

                    je addDateCase

                    cmp al, 4eh *; 45h - char N*

                    je notAddDateCase

                    jmp errorDateSelect

                errorDateSelect:

                    printString clearScreen

                    printString wrongSelectLtrs

                    printNewLineSymb

                    jmp addOrNotDate

                addDateCase:

                    setDateToStr

                    jmp notAddDateCase

                notAddDateCase:

                    popa

    endm

    replaceStrToSubStr macro str, position, subStr, subStrLen

*; вставляет в строку на определенную позицию подстроку, что была передана*

        pusha

        mov di, str

        lea si, subStr

        add di, position

        mov cx, subStrLen

        rep movsb

        popa

    endm

    setDateToStr macro

*; устанавливает дату для определенного пользователя*

        pusha

        mov bx, offset allStudentsStr

        add bx, [studId]

        printString mesInputDate

        readStrToBuffer buffer, 8

        replaceStrToSubStr [bx], 43, buffer+2, 8

        popa

    endm

    readStrToBuffer macro buffer, bufLen

*; считывает с ввода строку и записывает ее в переменную buffer*

        pusha

            mov dx, offset buffer

            mov ah, 0ah

            int 21h

        popa

    endm

.CODE

    MAIN:

        printString clearScreen

        mov ax,@Data

        mov ds,ax

        mov es,ax

        programmLoop:

            printTable

            printNewLineSymb

            printString mesForUser

            getControlCharacter

        setPassetOrNotPassed:

            printString clearScreen

            getStudentNumber

            printString clearScreen

            setPassedOrNotPassed

            jmp programmLoop

        setDate:

            printString clearScreen

            getStudentNumber

            printString clearScreen

            setDateToStr

            jmp programmLoop

        exit:

*; конец программы*

            printString clearScreen

            printString mesForEnd

            mov ah, 4ch

            int 21h

    END MAIN

**Скриншоты**

**Изображение выглядит как текст

Автоматически созданное описание**

Рисунок 1 – основное меню программы

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 2 – обработчик ошибок на неправильно введенный символ

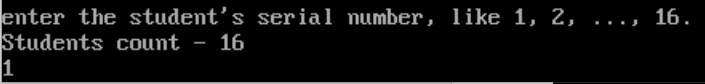


Рисунок 3 – окно ввода номера студента

Изображение выглядит как текст

Автоматически созданное описаниеРисунок 4 – окно ввода зачета или незачета студента

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 5 – Предложение также сразу добавить дату для данного студента

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 6 – окно ввода даты для студента

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 7 – измененные данные в таблице после манипуляций