**Цель:** освоение средств обеспечивающих модульное программирование.

Код основного модуля main, который после запуска выводит на экран сообщение "Press any key". Затем управление передается модулю input, который осуществляет ожидание нажатия клавиши. после нажатия клавиши:

* если была нажата клавиша А, управление передается модулю input1, в котором процедура output1 выводит на экран сообщение ""
* если была нажата любая другая клавиша, управление передается модулю input2, в котором процедура output2 выводит на экран сообщение ""

**Код модулей:**

|  |  |
| --- | --- |
| TITLE MAINEXTRN INPUT:PROCEXTRN OUTPUT1:PROCEXTRN OUTPUT2:PROC.Model Small.stack 100h.Data message db 13,10, 'Press any key$'.Codebegin:mov ax, @datamov ds, axmov ah,09lea dx, messageint 21hcond:call INPUT ; вызов процедуры INPUTcmp al,61h ; сравнение нажатой клавиши с Аje lendcall OUTPUT2 ; вызов процедуры OUTPUT2jmp condlend:call OUTPUT1 ; вызов процедуры OUTPUT1mov ah,1h int 21h mov ax,4c00h int 21hend begin | TITLE INPUTPUBLIC INPUT.Model Small.Stack 100h.Data.CodeINPUT PROCmov ax,@datamov ds,ax;ожидание нажатия клавишиmov ah,1hint 21hretINPUT endpEnd |

|  |  |
| --- | --- |
| TITLE OUTPUT1PUBLIC OUTPUT1.model small.stack 100h.datames1 db 10,13, 'Key A was pressed. Exiting now$'.codeOUTPUT1 PROCmov ax,@datamov ds,ax;вывод сообщения 'Key A was pressed. Exiting now'mov ah,09hlea dx,mes1int 21hretOUTPUT1 endpend | TITLE OUTPUT2PUBLIC OUTPUT2.model small.stack 100h.datames2 db 10,13, 'Press key A$'.codeOUTPUT2 PROCmov ax,@datamov ds,axmov ah,09h;вывод сообщения "Press key A"lea dx,mes2int 21hretOUTPUT2 endpend |





**Модернизация:**

После разработки и отладки рассмотренной выше многомодульной программы требуется модернизировать текст модулей созданной программы таким образом, чтобы обеспечить демонстрацию передачи параметров процедурам через стек. Модернизированную программу необходимо отладить и составить отчет о проделанной работе.

|  |  |
| --- | --- |
| **TITLE MAIN**EXTRN INPUT:PROCEXTRN OUTPUT1:PROCEXTRN OUTPUT2:PROC.ModelSmall.stack 100h.Data message db 'Press any key$'.Codebegin:mov ax,@datamov ds,axmov ah,9lea dx,messageint 21hlabel1:push 9h ;заносим в стек значение, для того чтобы push 8h потом извлечь его в регистр ax**call INPUT**add sp,2 ;отметка 2 cmp al,61h ;сравнение с ASCII-кодом кнопки «a»je label\_endpush 9h ;заносим в стек значение, для того чтобы  потом извлечь его в регистр ax**call OUTPUT2**add sp,2jmp label1label\_end:push 9h ;заносим в стек значение, для того чтобы  потом извлечь его в регистр ax**call OUTPUT1** add sp,2mov ah,4Chint 21hend begin**Комментарий:** В процессе, когда программа дойдет до отметки 2, значение sp будет равно 00FC и для того, чтобы вернуть стек в исходное состояние нам необходимо использовать команду “add sp, 2”. После ее использования, видно, что стек вернулся в исходное состояние:**OUTPUT 1**TITLE OUTPUT1PUBLIC OUTPUT1.Model small.stack 100h.datames1 db 10,13,'Key A was pressed. Exiting now$'.codeOUTPUT1 PROC ;отметка 3 mov ax,@datamov ds,axpush bpmov bp, spmov ah,[bp+4]lea dx, mes1int 21hpop bpretOUTPUT1 endpend**Комментарий:** В процессе, когда программа дойдет до отметки 3, значение sp будет равно 00FC , так как происходят аналогичные действия, что и в предыдущем случае. | **TITLE INPUT**PUBLIC INPUT.model small.stack 100h.data.codeINPUT PROC ; отметка 1 mov ax,@datamov ds,axpush bpmov bp, spmov ah,[bp+4]int 21hpop bpretINPUT endpend**Комментарий:** В процессе, когда программа дойдет до отметки 1, значение sp будет равно **OUTPUT 2**TITLE OUTPUT2PUBLIC OUTPUT2.model small.stack 100h.datames2 db 10,13,'Press key A$'.codeOUTPUT2 PROC ;отметка 4 mov ax,@datamov ds,axpush bpmov bp, spmov ah,[bp+4]lea dx, mes2int 21hpop bpretOUTPUT2 endpEnd**Комментарий:** В процессе, когда программа дойдет до отметки 4, значение sp будет равно 00FC , так как происходят аналогичные действия, что и в предыдущих случаях (отметки 2 и 3) |



**Вывод:** в проделанной лабораторной работе было частично изучено модульное программирование и вызов отдельных модулей программы (директивы PUBLIC и EXTRN).

Для вызова процедуры необходима команда call (в процедуре обязательно должна быть команда ret – она возращает программу в то место , где была выполнена команда call). Для реализации этой задачи мы использовали стек. Для работы со стеком мы использовали операции push (положить данные в стек) и pop (взять данные из стека)*.*