****

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ**

**“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ”**

**Факультет прикладної математики**

**Кафедра системного програмування і спеціалізованих комп’ютерних систем**

**Лабораторна робота №6**

**з дисципліни “Системне програмування”**

**Тема: «**Вивчення методики програмування переривань від зовнішніх пристроїв, ознайомлення з організацією мультипрограмних систем**»**

**Варіант 5**

**Виконав:**

**студент II-го курсу**

**групи КВ-41**

**Горпинич-Радуженко Іван**

**Київ 2016**

**Завдання на виконання роботи**

При натисненні на клавішу R зупинити всі задачі з парними номерами, при відпусканні клавіші Pause/Break всі задачі продовжують свою роботу.

Microsoft (R) Macro Assembler Version 5.00 4/13/16 10:09:22

Page 1-1

.386

;==============================================

========

; МОДЕЛЬ МУЛЬТИПРОГРАМНОЇ СИСТЕМИ

;==============================================

========

= 000E max\_prg equ 14

;максимальна кількість "одночасно"

; виконуваних задач

= FFFF time\_slice equ 65535;

кількість мікросекунд, виділених на один

; квант часу (максимальне значення 6553

5)

0000 \_ST SEGMENT WORD ST

ACK 'stack' use16

0000 7D00[ dw

32000 dup (?)

????

]

FA00 top label word

FA00 0064[ dw 100 dup

(?);резерв для помилок

????

]

; анти переповнення стека

FAC8 \_ST ENDS

0000 \_DATA SEGMENT WORD PUBLIC 'DATA' u

se16

0000 ???????? @ms\_dos\_busy dd (?) ; логічна адреса

ознаки зайнятості MS-DOS

0004 00 int8set db 0 ;

ознака перехоплення переривання від таймера

0005 00 int9set db 0 ;

ознака перехоплення переривання від клавіатури

= fon equ max\_prg

; ознака фонової задачі;

0006 fonsp label word

;адреса збереження SP фонової задачі

0006 FA00 ---- R sssp dd top

;логічна адреса стека фонової з

адачі

; масив значень SP для задач, (для стека кожн

ої задачі відведено 1000 слів)

Microsoft (R) Macro Assembler Version 5.00 4/13/16 10:09:22

Page 1-2

;задані початкові значення

000A 03E8 07D0 0BB8 0FA0 stp dw 1000,2

000,3000,4000

0012 1388 1770 1B58 1F40 dw 5000,6000,7000

,8000

001A 2328 2710 2AF8 2EE0 dw 9000,1

0000,11000,12000

0022 32C8 36B0 3A98 3E80 dw 13000,14000,15

000,16000

002A 0000 nprg dw 0

;номер активної задачі (від 0 д

о

;max\_prg-1)

; або ознака фо

нової задачі (fon)

; масив стану задач

002C 0010[ init db 16 dup (0)

00

]

; масив дозволеного числа квантів задач

003C 0010[ clock db 16 dup (1)

01

]

; масив лічильників квантів задач

004C 0010[ clockt db 16 dup (0)

00

]

005C 0010[ screen\_addr dw 16 dup (0) ; адреса (зм

іщення від початку відеосторінки)

0000

]

; області введення на екран значень задачі

; масив імен задач

007C names label word

007C 30 54 31 54 32 54 33 db '0T1T2T3T4T5T6T7T8T9

TATBTCTDTETFT'

54 34 54 35 54 36 54

37 54 38 54 39 54 41

54 42 54 43 54 44 54

45 54 46 54

009C 0000 clk dw 0 ;лічильни

Microsoft (R) Macro Assembler Version 5.00 4/13/16 10:09:22

Page 1-3

к переривань від таймера

009E \_DATA ENDS

0000 \_TEXT SEGMENT BYTE PUBLIC 'CODE' use

16

ASSUME CS:\_TEXT,DS:\_DATA

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

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

; процедура "перехоплення" переривання від тайм

ера (int8)

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

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

0000 setint8 PROC

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

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

0000 A0 0004 R mov al,int8set

0003 0A C0 or al,al

; контроль "перехоплення" перех

оплень

0005 0F 85 0037 R jnz zero\_8 ;

0009 B4 35 MOV AH,35H

; отримати вектор переривання

000B B0 08 MOV AL,8

; переривання від таймера (8)

000D CD 21 INT 21H

; значення що повертається:

; es:bx - логічна адреса систем

ної процедури

; обробки переривання від тайме

ра

000F 2E: 89 1E 0038 R mov cs:int8ptr,bx

; зберегти логічну адресу системної

0014 2E: 8C 06 003A R mov cs:int8ptr+2,es

; процедури в сегменті кодів

0019 BA 018B R mov dx,offset useri

nt8 ;формування в ds:dx логічної

001C 1E push ds

; адреси процедури користувача

001D 0E push cs

; для обробки переривань від та

ймера

001E 1F pop ds

001F B4 25 MOV AH,25H

; встановити вектор

0021 B0 08 MOV AL,8

; переривання від таймера

Microsoft (R) Macro Assembler Version 5.00 4/13/16 10:09:22

Page 1-4

0023 CD 21 INT 21H

; ds:dx - покажчик на користувацьку

; процедуру оброб. переривання від ;таймера

0025 B8 FFFF mov ax,time\_slice

; встановити задану величину кванту час

у

0028 E6 40 out 40h,al

; 40h - адреса 8-розрядного порта тайме

ра,

; через який задають період таймера

; спочатку молодший байт,

; а потім старший

002A EB 00 jmp $+2

; стандартний метод узгодження

швидкісного

; процесора з більш повільним з

овнішнім

; пристроєм. Припускаємо, що

; "безглузда" команда jmp очищу

є буфер

; попередньої вибірки команд і,

тим самим,

; уповільнює роботу процесора.

Тим часом

; зовнішній пристрій буде готов

ий

;прийняти наступний байт

002C 90 nop

002D 8A C4 mov al,ah

; (старший байт)

002F E6 40 out 40h,al

0031 1F pop ds

0032 C6 06 0004 R FF mov int8set,0ffh

; заборона повторних входжень

0037 zero\_8:

0037 C3 ret

0038 0002[ int8ptr dw 2 dup (?)

????

]

Microsoft (R) Macro Assembler Version 5.00 4/13/16 10:09:22

Page 1-5

003C setint8 ENDP

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

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

; Процедура відновлення вектора переривання ві

д таймера

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

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

003C retint8 PROC

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

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

003C 1E push ds

003D 52 push dx

003E B0 FF mov al,0ffh

; відновити нормальну роботу

0040 E6 40 out 40h,al

; системного таймера

0042 EB 00 jmp $+2

0044 90 nop

0045 E6 40 out 40h,al

0047 2E: 8B 16 0038 R mov dx,cs:int8ptr

004C 2E: 8E 1E 003A R mov ds,cs:int8ptr+2

0051 B4 25 MOV AH,25H

; відновити початковий вектор

0053 B0 08 MOV AL,8

; переривання від таймера

0055 CD 21 INT 21H

; ds:dx - вказівник (логічна адреса)

;на початкову (системну) процедуру

; оброб. переривання від таймера

0057 5A pop dx

0058 1F pop ds

0059 C6 06 0004 R 00 mov int8set,0h

; дозвіл наступних "перехоплень"

005E C3 ret

005F retint8 ENDP

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

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

005F setint9 PROC

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

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

; процедура "перехоплення" переривання від кла

віатури (int9)

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

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

005F A0 0005 R mov al,int9set

Microsoft (R) Macro Assembler Version 5.00 4/13/16 10:09:22

Page 1-6

0062 0A C0 or al,al

0064 0F 85 008A R jnz zero\_9

0068 B4 35 MOV AH,35H

; отримати вектор переривання

006A B0 09 MOV AL,9

; переривання від клавіатури (9)

006C CD 21 INT 21H

;значення що повертається:

; es:bx - вказівник на системну

процедуру

; обробки переривання від клаві

атури

006E 2E: 89 1E 008B R mov cs:int9ptr,bx

; зберегти в сегменті кодів вказівник

0073 2E: 8C 06 008D R mov cs:int9ptr+2,es

; на системну процедуру

0078 BA 00A9 R mov dx,offset useri

nt9

007B 1E push ds

007C 0E push cs

; ds:dx - вказівник на процедур

у користувача

007D 1F pop ds

; оброб. переривання від клавіа

тури

007E B4 25 MOV AH,25H

; встановити вектор "перехоплення"

0080 B0 09 MOV AL,9

; переривання від клавіатури (9)

0082 CD 21 INT 21H

;

0084 1F pop ds

0085 C6 06 0005 R FF mov int9set,0ffh

; заборона повторних входжень

008A zero\_9:

008A C3 ret

008B 0002[ int9ptr dw 2 dup (?)

????

]

008F setint9 ENDP

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

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

; Процедура відновлення попереднього (системно

го)

Microsoft (R) Macro Assembler Version 5.00 4/13/16 10:09:22

Page 1-7

; вектора переривання від клавіатури

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

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

008F retint9 PROC

008F 1E push ds

0090 52 push dx

0091 2E: 8B 16 008B R mov dx,cs:int9ptr

; ds:dx - покажчик на початкову (систем

ну)

0096 2E: 8E 1E 008D R mov ds,cs:int9ptr+2

; процедуру обробки переривання від

; клавіатури

009B B4 25 MOV AH,25H

; встановити вектор системної процедури

009D B0 09 MOV AL,9

; обробки переривання від клавіатури

009F CD 21 INT 21H

;

;

00A1 5A pop dx

00A2 1F pop ds

00A3 C6 06 0005 R 00 mov int9set,0h

; дозвіл наступних "перехоплень"

00A8 C3 ret

00A9 retint9 ENDP

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

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

--

; Процедура обробки переривань від клавіатури

,

; викликається при любому натисканні або відт

исканні клавіш клавіатури,

; здійснює повернення в MS-DOS після відтиска

ння клавіші Esc

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

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

---

00A9 userint9 proc far

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

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

= 0001 esc\_key equ 01h

; скан-код клавіші esc

= 0013 R\_key equ 13h

= 0014 T\_key equ 14h

00A9 60 pusha

00AA 06 push es

00AB E4 60 in al,60h

Microsoft (R) Macro Assembler Version 5.00 4/13/16 10:09:22

Page 1-8

; ввести скан-код - розряди 0-6

00AD 8A E0 mov ah,al

; 7-ий розряд дорівнює 0 при на

тисканні

00AF 24 7F and al,7fh

;клавіші, 1- при відтисканні

00B1 50 push ax

; al - "чистий" скан-код (без о

знаки

; натискання - відтискання)

00B2 68 0A28 push 2600

00B5 E8 02FF R call show

; відображення скан-коду на екр

ані

00B8 3C 01 cmp al,esc\_key

00BA 0F 84 0152 R je ui9010

;koristyvacka programa

00BE 80 FC 13 cmp ah, R\_key

00C1 0F 84 00CF R je prog

00C5 80 FC 14 cmp ah, T\_key

00C8 0F 84 00F5 R je progt

00CC EB 7D 90 jmp @notcontrolkey

00CF prog:

00CF 50 push ax

00D0 53 push bx

00D1 B8 ---- R mov ax,\_DATA

;DS<-data segment

00D4 8E D8 mov ds,ax

00D6 B0 10 mov al,100

00b

00D8 A2 002C R mov init[0

], al

00DB A2 002E R mov init[2

], al

00DE A2 0030 R mov init[4

], al

00E1 A2 0032 R mov init[6

], al

00E4 A2 0034 R mov init[8

], al

00E7 A2 0036 R mov init[1

Microsoft (R) Macro Assembler Version 5.00 4/13/16 10:09:22

Page 1-9

0], al

00EA A2 0038 R mov init[1

2], al

00ED A2 003A R mov init[1

4], al

; mov init[15],ready

00F0 5B pop bx

00F1 58 pop ax

00F2 EB 57 90 jmp @notcontrol

key

00F5 progt:

00F5 50 push ax

00F6 53 push bx

00F7 B8 ---- R mov ax,\_DATA

;DS<-data segment

00FA 8E D8 mov ds,ax

00FC B0 01 mov al,1b

00FE C6 06 002C R 00 mov init[0

], ready

0103 C6 06 002D R 00 mov init[1

], ready

0108 C6 06 002E R 00 mov init[2

], ready

010D C6 06 002F R 00 mov init[3

], ready

0112 C6 06 0030 R 00 mov init[4

], ready

0117 C6 06 0031 R 00 mov init[5

], ready

011C C6 06 0032 R 00 mov init[6

], ready

0121 C6 06 0033 R 00 mov init[7

], ready

0126 C6 06 0034 R 00 mov init[8

], ready

012B C6 06 0035 R 00 mov init[9

], ready

0130 C6 06 0036 R 00 mov init[1

0], ready

0135 C6 06 0037 R 00 mov init[1

1], ready

013A C6 06 0038 R 00 mov init[1

2], ready

013F C6 06 0039 R 00 mov init[1

3], ready

0144 C6 06 003A R 00 mov init[1

4], ready

Microsoft (R) Macro Assembler Version 5.00 4/13/16 10:09:22

Page 1-10

0149 5B pop bx

014A 58 pop ax

014B @notcontrolkey:

; (варіант 2)

014B 07 pop es

014C 61 popa

014D 2E: FF 2E 008B R jmp dword ptr cs:int9ptr

; перехід на системну

;процедуру обро

бки

;переривань від

клавіатури, яка

;виконає всі не

обхідні дії, включаючи

;повернення в п

ерервану програму

; (варіант 1)

; САМОСТІЙНЕ ВИРІШЕННЯ ПРОБЛЕМ З КЛАВІАТУРОЮ

0152 ui9010:

0152 8B D8 mov bx,ax

0154 E4 61 in al,61h

;біт 7 порта 61h призначений для введен

ня

; підтверджуючого імпульсу в кл

авіатуру ПЕОМ.

; Клавіатура блокується поки не надійде

; підтверджуючий імпульс

;

0156 8A E0 mov ah,al

0158 0C 80 or al,80h

;

|

015A E6 61 out 61h,al

; виведення на клавіатуру L---¬

015C EB 00 jmp $+2

;

|

015E 8A C4 mov al,ah

Microsoft (R) Macro Assembler Version 5.00 4/13/16 10:09:22

Page 1-11

;

|

0160 E6 61 out 61h,al

; підтверджуючого імпульсу -----

0162 B0 20 mov al,20h ; розбл

окувати в контролері переривання

; проходження запитів на переривання

;поточного та меншого рівнів пріоритету,

0164 E6 20 out 20h,al

; що забезпечить можливість наступного

;переривання від клавіатури

0166 8B C3 mov ax,bx

0168 3A E0 cmp ah,al ; перев

ірка події переривання - від натискання

; чи від відтискання клавіші клавіатури

016A 0F 84 0188 R je ui9040

;відтискання клавіші

016E ui9020:

016E 06 push es

016F C4 1E 0000 R les bx, @ms\_dos\_bus

y ; es:bx - адреса ознаки

;зайнятості MS-DOS

0173 26: 8A 07 mov al,es:[bx]

; ax - ознака зайнятості MS-DOS

0176 07 pop es

0177 0A C0 or al,al

; перевірка

; якщо була перервана робота MS-DOS

;в "невдалий" момент

0179 0F 85 0188 R jnz ui9040

; то не можна від неї вимагати

; виконання ряду функцій

; (в за

гальному випадку MS-DOS

; не за

безпечує повторне входження)

017D E8 003C R call retint8

0180 E8 008F R call retint9

0183 B8 4C00 mov ax,4c00h

0186 CD 21 int 21h

; ЗАКІНЧИТИ РОБОТУ

; БАГАТОПРОГРАМНОЇ МОДЕЛІ

Microsoft (R) Macro Assembler Version 5.00 4/13/16 10:09:22

Page 1-12

0188 ui9040:

0188 07 pop es

; відновити стек перерваної програми

0189 61 popa

018A CF iret ; закін

чити обробку переривання

018B userint9 endp

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

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

; процедура обробки переривання від таймера

; (менеджер квантів)

; коди стану задач (використовуються в маси

ві init)

= 0000 ready equ 0 ; задача завант

ажена в пам’ять і

; готова до початкового запуску

; статус встановлюється поза менеджером

квантів

= 0001 execute equ 1 ; задача викону

ється

= 0002 hesitation equ 2 ; задача призуп

инена і чекає своєї черги

= 0004 close equ 4 ; виконання зад

ачі завершено

= 0008 stop equ 8 ; задач

а зупинена

; статус встановлюється і змінюється

; поза менеджера квантів

= 0010 absent equ 16 ; задач

а відсутня

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

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

018B userint8 PROC far

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

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

018B 66| 60 pushad

;збереження РОН в стеку перерваної зада

чі

018D 1E push ds

; (варіант 3)

018E 9C pushf

;програмна імітація апаратного перерива

ння

Microsoft (R) Macro Assembler Version 5.00 4/13/16 10:09:22

Page 1-13

;ВІДМІТИМО - ознака дозволу на переривання (if)

апаратурою скинута в 0.

018F 2E: FF 1E 0038 R call cs:dword ptr int8ptr

;виклик системної процедури обробки пер

еривання int8,

;яка, між іншим, розблокує 8-ме перерив

ання в контролері переривань

;але апаратні переривання не можливі, о

скільки if=0

0194 B8 ---- R mov ax,\_data

;в перерваній програмі вміст сегментног

о регістра

0197 8E D8 mov ds,ax

;ds в загальному випадку може бути люби

м

0199 FF 06 009C R inc clk

; програмний лічильник переривань від т

аймера

019D FF 36 009C R push clk

; може бути корисним при вивченні модел

і

01A1 68 0988 push 2440

01A4 E8 02FF R call show

; виведення на екран значення лічильник

а

01A7 66| 33 F6 xor esi,esi

01AA 8B 36 002A R mov si,nprg

01AE 83 FE 0E cmp si,fon ; перер

вана задача фонова ?

01B1 0F 84 01C6 R je disp005

; перервана задача не фонова

01B5 80 BC 004C R 01 cmp clockt[si],1 ; є ще

не використані кванти ?

01BA 0F 82 01D6 R jc disp010

01BE FE 8C 004C R dec clockt[si] ; зменш

ити лічильник квантів

01C2 1F pop ds

01C3 66| 61 popad ; продо

вжити виконання перерваної задачі

01C5 CF iret

01C6 disp005: ; пере

рвана задача фонова

01C6 89 26 0006 R mov fonsp,sp

01CA C7 06 002A R 000D mov nprg,max\_prg-1

; забезпечити перегляд задач з 0-вої

Microsoft (R) Macro Assembler Version 5.00 4/13/16 10:09:22

Page 1-14

01D0 B9 000E mov cx,max\_prg

; max\_prg - max кількість задач

01D3 EB 11 90 jmp disp015

01D6 disp010:

; перервана задача не ф

онова

01D6 67| 89 2475 0000000A mov stp[esi\*2],sp

R

01DE C6 84 002C R 02 mov init[si],hesita

tion ; призупинити поточну задачу

01E3 B9 000E mov cx,max\_prg

01E6 disp015:

; визначення задачі, якій необхідно пер

едати управління

01E6 BF 000F mov di,max\_prg+1

01E9 2B F9 sub di,cx

01EB 03 3E 002A R add di,nprg

01EF 83 FF 0E cmp di,max\_prg

01F2 0F 82 01F9 R jc disp018

01F6 83 EF 0E sub di,max\_prg

01F9 disp018:

01F9 66| 33 DB xor ebx,ebx

01FC 8B DF mov bx,di

;push bx

;push 3220

;call show

; сх пр

обігає значення max\_prg,max\_prg-1,...,2,1

; bx пр

обігає значення nprg+1,nprg+2,...,max\_prg-

;1,0,...,nprg

;

01FE 80 BF 002C R 00 cmp init[bx],ready

0203 0F 84 023D R je disp100

; перехід на початковий запуск

задачі

0207 80 BF 002C R 02 cmp init[bx],hesitation

020C 0F 84 0220 R je disp020

; перехід на відновлення роботи

; наступної задачі

0210 E2 D4 loop disp015

; відсутні задачі, які

можна запустить

Microsoft (R) Macro Assembler Version 5.00 4/13/16 10:09:22

Page 1-15

; (перезапустити), тому

;

0212 8B 26 0006 R mov sp,fonsp

; установлюємо стек фонової зад

ачі

0216 C7 06 002A R 000E mov nprg,fon

021C 1F pop ds

; із стека фонової задачі відно

влюємо

021D 66| 61 popad

; вміст регістрів

021F CF iret

; повернення в фонову

задачу

0220 disp020:

; відновлення роботи на

ступної задачі

;push bx

;push 2480

;call show

0220 89 1E 002A R mov nprg,bx

0224 67| 8B 245D 0000000A mov sp,stp[ebx\*2]

R

022C 8A 87 003C R mov al,clock[bx]

0230 88 87 004C R mov clockt[bx],al

; встановити дозволену

; кількість квантів

0234 C6 87 002C R 01 mov init[bx],execute

; стан задачі - задача виконуєт

ься

0239 1F pop ds

023A 66| 61 popad

023C CF iret

023D disp100:

; першопочатковий запус

к задачі

023D 89 1E 002A R mov nprg,bx

0241 67| 8B 245D 0000000A mov sp,stp[ebx\*2]

R

0249 8A 87 003C R mov al,clock[bx]

024D 88 87 004C R mov clockt[bx],al

; встановити дозволену

; кількість квантів

0251 C6 87 002C R 01 mov init[bx],execut

Microsoft (R) Macro Assembler Version 5.00 4/13/16 10:09:22

Page 1-16

e

0256 67| FF 345D 0000007C push names[ebx\*2]

; ім'я задачі

R

025E 67| FF 345D 0000005C push screen\_addr[ebx

\*2] ; адреса "вікна" для задачі на екрані

R

0266 6A 1D push 29

; розрядність лічильник

а

0268 E8 0285 R call Vcount

; запуск

026B 66| 33 F6 xor esi,esi

026E 8B 36 002A R mov si,nprg

; на ax - номер задачі, яка

; завершила свою роботу

в межах

; чергового кванту часу

0272 C6 84 002C R 04 mov init[si],close

0277 8B 26 0006 R mov sp,fonsp

027B C7 06 002A R 000E mov nprg,fon

0281 1F pop ds

0282 66| 61 popad

0284 @ui8exit:

0284 CF iret

; повернення в фонову

задачу

0285 userint8 ENDP

;-

; Vcount - процедура для моделювання незалежних

задач

; вхідні параметри:

; 1-й - ім'я задачі (два символа) [bp+8]

; 2-й - зміщення в відеосторінці "вікна"

задачі [bp+6]

; 3-й - кількість двійкових розрядів лічи

льника [bp+4]

; Виконувані дії:

; при запуску:

; - дозволяє переривання

; - створює в стеку 10-байтну область для лока

льних даних

; - розміщує в цю область по адресі [bp-2] ста

ток від ділення

; 3-го параметра на 32 (фактична

розрядність лічильника -

; перестраховка від помилок в зав

Microsoft (R) Macro Assembler Version 5.00 4/13/16 10:09:22

Page 1-17

данні розрядності)

; - записує в цю область по адресу [bp-6] мас

ку з числом

; одиниць в молодших розрядів рів

ним фактичній

; розрядності лічильника

; - записує в нуль в 4-х байт ний лічильник по

адресу [bp-10]

; в подальшому в циклі:

; - виводить показники лічильника на

екран

; - збільшує значення лічильника на 1

; завершення задачі після переходу лічильника

; з стану "всі одиниці" в стан всі 0

0285 Vcount proc near

0285 55 push bp

0286 8B EC mov bp,sp

0288 83 EC 0A sub sp,10

;формування в стеку області для

;збереження даних

028B FB sti

028C 06 push es

028D B8 B800 mov ax,0b800h

0290 8E C0 mov es,ax

0292 8B 46 04 mov ax,[bp+4]

;ax = кількість розрядів лічильника

0295 25 001F and ax,31

;ax=ax mod 32 (для перестраховк

и)

0298 89 46 FE mov [bp-2],ax

;по [bp-2] кількість розр. лічи

льника

;<32

029B 8B C8 mov cx,ax

029D 66| B8 00000001 mov eax,001b

02A3 66| D3 E0 shl eax,cl

02A6 66| 48 dec eax

; eax - маска с числом 1 рівним

; кількості розрядів лічильника

02A8 66| 89 46 FA mov [bp-6],eax

02AC 66| C7 46 F6 00000000 mov dword ptr [bp-10],0 ; скида

Microsoft (R) Macro Assembler Version 5.00 4/13/16 10:09:22

Page 1-18

ння лічильника

02B4 8B 7E 06 mov di,[bp+6]

; вивід імені задачі

02B7 8B 56 08 mov dx,[bp+8]

02BA B4 0B mov ah,1011b

02BC 8A C6 mov al,dh

02BE FC cld

02BF AB stosw

02C0 8A C2 mov al,dl

02C2 AB stosw

02C3 FD std

;підготовка до виводу лічильника

02C4 03 F9 add di,cx

;починаючи с молодших розрядів

02C6 03 F9 add di,cx

02C8 8B DF mov bx,di

02CA 66| 33 D2 xor edx,edx

02CD l20:

;вивід показників лічильника в двоїчно

му

;форматі

02CD 8B FB mov di,bx

02CF 8B 4E FE mov cx,[bp-2]

02D2 B4 0A mov ah,1010b

; 1010b атрибут символу, атрибут фона -

0

; (чорний)

02D4 l40:

02D4 B0 30 mov al,'0'

02D6 66| D1 EA shr edx,1

02D9 0F 83 02DF R jnc l60

02DD B0 31 mov al,'1'

02DF l60:

02DF AB stosw

02E0 E2 F2 loop l40

02E2 66| FF 46 F6 inc dword ptr [bp-10]

; +1 в лічильник

02E6 66| 8B 56 F6 mov edx,dword ptr [bp-10]

02EA 66| 23 56 FA and edx,[bp-6]

; перевірка на 0

02EE 75 DD jnz l20

02F0 07 pop es

02F1 83 C4 0A add sp,10

02F4 8B 46 08 mov ax,[bp+8]

02F7 25 000F and ax,0fh

02FA FA cli

02FB 5D pop bp

Microsoft (R) Macro Assembler Version 5.00 4/13/16 10:09:22

Page 1-19

02FC C2 0006 ret 6

02FF Vcount endp

;=====

02FF show proc near

02FF 55 push bp

0300 8B EC mov bp,sp

0302 60 pusha

0303 06 push es

0304 B8 B800 mov ax,0b800h

0307 8E C0 mov es,ax

0309 FD std

030A ls20:

030A 8B 7E 04 mov di,[bp+4]

030D 8B 5E 06 mov bx,[bp+6]

0310 B9 0004 mov cx,4

0313 B4 0A mov ah,0ah

0315 ls40:

0315 8A C3 mov al,bl

0317 24 0F and al,00001111b

0319 3C 0A cmp al,10

031B 0F 8C 0321 R jl ls100

031F 04 07 add al,7

0321 ls100:

0321 04 30 add al,30h

0323 AB stosw

0324 C1 EB 04 shr bx,4

0327 E2 EC loop ls40

0329 07 pop es

032A 61 popa

032B 5D pop bp

032C C2 0004 ret 4

032F show endp

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

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

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

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

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

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

032F begin:

032F B8 ---- R mov ax,\_data

0332 8E D8 mov ds,ax

0334 B8 0003 mov ax,3

; задати текстовий режим 80 на

25

0337 CD 10 int 10h

0339 B4 10 mov ah,10h

; відключити режим миготіння

033B B0 03 mov al,3

033D B3 00 mov bl,0

Microsoft (R) Macro Assembler Version 5.00 4/13/16 10:09:22

Page 1-20

033F CD 10 int 10h

0341 B9 000E mov cx,max\_prg

0344 66| 33 F6 xor esi,esi

0347 BB 0004 mov bx,4

034A b10:

034A 67| 89 1C75 0000005C mov screen\_addr[esi\*2],bx

; заповнення таблиці

R

; адрес виводу для зада

ч

0352 C6 84 002C R 00 mov init[si],ready

; першопочаткове заповнення

; таблиці стану

задач

0357 83 C3 50 add bx,80

035A 46 inc si

035B E2 ED loop b10

;SETINT

035D FA cli

; заборона переривань

035E B4 34 mov ah,34h

0360 CD 21 int 21h ;es:bx

- адреса ознаки зайнятості MS-DOS

0362 89 1E 0000 R mov word ptr @ms\_dos\_busy,b

x

0366 8C 06 0002 R mov word ptr @ms\_dos\_busy+2

,es

036A E8 0000 R call setint8 ;"перех

оплення" int8

036D E8 005F R call setint9 ;"перех

оплення" int9

0370 0F B2 26 0006 R lss sp,sssp ; стек

фонової задачі

0375 C7 06 002A R 000E mov nprg,fon

037B 68 464E push 'FN'

037E 68 0708 push 1800

0381 6A 1E push 30

0383 E8 0285 R call Vcount

; запуск фонової задачі

; в про

цедурі Vcount установлюється дозвіл

;на пер

еривання і при чергових перериваннях

; від т

аймера менеджер квантів (userint8)

; буде

запускать інші задачі

Microsoft (R) Macro Assembler Version 5.00 4/13/16 10:09:22

Page 1-21

;

; управління в цю точку буде передано по команд

і RET по завершені фонової ; задачі, а це можли

во лише після завершення інших задач

0386 E8 003C R call retint8

; відновлення системних векторів

0389 E8 008F R call retint9

038C FB sti

038D B8 4C00 mov ax,4c00h

0390 CD 21 int 21h

0392 \_TEXT ENDS

end begin

Microsoft (R) Macro Assembler Version 5.00 4/13/16 10:09:22

Symbols-1

N a m e Size Length Align Combine Class

\_DATA . . . . . . . . . . . . . 16 Bit 009E WORD PUBLIC 'DATA'

\_ST . . . . . . . . . . . . . . 16 Bit FAC8 WORD STACK 'STACK'

\_TEXT . . . . . . . . . . . . . 16 Bit 0392 BYTE PUBLIC 'CODE'

Symbols:

N a m e Type Value Attr

ABSENT . . . . . . . . . . . . . NUMBER 0010

B10 . . . . . . . . . . . . . . L NEAR 034A \_TEXT

BEGIN . . . . . . . . . . . . . L NEAR 032F \_TEXT

CLK . . . . . . . . . . . . . . L WORD 009C \_DATA

CLOCK . . . . . . . . . . . . . L BYTE 003C \_DATA Length = 0010

CLOCKT . . . . . . . . . . . . . L BYTE 004C \_DATA Length = 0010

CLOSE . . . . . . . . . . . . . NUMBER 0004

DISP005 . . . . . . . . . . . . L NEAR 01C6 \_TEXT

DISP010 . . . . . . . . . . . . L NEAR 01D6 \_TEXT

DISP015 . . . . . . . . . . . . L NEAR 01E6 \_TEXT

DISP018 . . . . . . . . . . . . L NEAR 01F9 \_TEXT

DISP020 . . . . . . . . . . . . L NEAR 0220 \_TEXT

DISP100 . . . . . . . . . . . . L NEAR 023D \_TEXT

ESC\_KEY . . . . . . . . . . . . NUMBER 0001

EXECUTE . . . . . . . . . . . . NUMBER 0001

FON . . . . . . . . . . . . . . ALIAS MAX\_PRG

FONSP . . . . . . . . . . . . . L WORD 0006 \_DATA

HESITATION . . . . . . . . . . . NUMBER 0002

INIT . . . . . . . . . . . . . . L BYTE 002C \_DATA Length = 0010

INT8PTR . . . . . . . . . . . . L WORD 0038 \_TEXT Length = 0002

INT8SET . . . . . . . . . . . . L BYTE 0004 \_DATA

INT9PTR . . . . . . . . . . . . L WORD 008B \_TEXT Length = 0002

INT9SET . . . . . . . . . . . . L BYTE 0005 \_DATA

L20 . . . . . . . . . . . . . . L NEAR 02CD \_TEXT

L40 . . . . . . . . . . . . . . L NEAR 02D4 \_TEXT

L60 . . . . . . . . . . . . . . L NEAR 02DF \_TEXT

LS100 . . . . . . . . . . . . . L NEAR 0321 \_TEXT

LS20 . . . . . . . . . . . . . . L NEAR 030A \_TEXT

LS40 . . . . . . . . . . . . . . L NEAR 0315 \_TEXT

MAX\_PRG . . . . . . . . . . . . NUMBER 000E

NAMES . . . . . . . . . . . . . L WORD 007C \_DATA

NPRG . . . . . . . . . . . . . . L WORD 002A \_DATA

Microsoft (R) Macro Assembler Version 5.00 4/13/16 10:09:22

Symbols-2

PROG . . . . . . . . . . . . . . L NEAR 00CF \_TEXT

PROGT . . . . . . . . . . . . . L NEAR 00F5 \_TEXT

READY . . . . . . . . . . . . . NUMBER 0000

RETINT8 . . . . . . . . . . . . N PROC 003C \_TEXT Length = 0023

RETINT9 . . . . . . . . . . . . N PROC 008F \_TEXT Length = 001A

R\_KEY . . . . . . . . . . . . . NUMBER 0013

SCREEN\_ADDR . . . . . . . . . . L WORD 005C \_DATA Length = 0010

SETINT8 . . . . . . . . . . . . N PROC 0000 \_TEXT Length = 003C

SETINT9 . . . . . . . . . . . . N PROC 005F \_TEXT Length = 0030

SHOW . . . . . . . . . . . . . . N PROC 02FF \_TEXT Length = 0030

SSSP . . . . . . . . . . . . . . L DWORD 0006 \_DATA

STOP . . . . . . . . . . . . . . NUMBER 0008

STP . . . . . . . . . . . . . . L WORD 000A \_DATA

TIME\_SLICE . . . . . . . . . . . NUMBER FFFF

TOP . . . . . . . . . . . . . . L WORD FA00 \_ST

T\_KEY . . . . . . . . . . . . . NUMBER 0014

UI9010 . . . . . . . . . . . . . L NEAR 0152 \_TEXT

UI9020 . . . . . . . . . . . . . L NEAR 016E \_TEXT

UI9040 . . . . . . . . . . . . . L NEAR 0188 \_TEXT

USERINT8 . . . . . . . . . . . . F PROC 018B \_TEXT Length = 00FA

USERINT9 . . . . . . . . . . . . F PROC 00A9 \_TEXT Length = 00E2

VCOUNT . . . . . . . . . . . . . N PROC 0285 \_TEXT Length = 007A

ZERO\_8 . . . . . . . . . . . . . L NEAR 0037 \_TEXT

ZERO\_9 . . . . . . . . . . . . . L NEAR 008A \_TEXT

@FILENAME . . . . . . . . . . . TEXT l6v5

@MS\_DOS\_BUSY . . . . . . . . . . L DWORD 0000 \_DATA

@NOTCONTROLKEY . . . . . . . . . L NEAR 014B \_TEXT

@UI8EXIT . . . . . . . . . . . . L NEAR 0284 \_TEXT

692 Source Lines

692 Total Lines

67 Symbols

50980 + 416412 Bytes symbol space free

0 Warning Errors

0 Severe Errors