Міністерство освіти і науки України Національний університет "Запорізька політехніка" Кафедра програмних засобів

3BIT

З лабораторної роботи №2 З дисципліни "Архітектура Комп'ютера та Низькорівневе Програмування" З теми "Структури програм на мові асемблера" Варіант №20

Виконав:

Студент групи КНТ-122 О.А. Онищенко

Прийняли:

Ст. Викладач О. І. Качан

Доцент КТН А. Є. Казурова

1 Завдання	3
1.1 Текст завдання	3
2 Хід роботи	4
2.1 ЕХЕ Програма	4
2.1.1 Код програми	4
2.1.2 Дизасемблювання	5
2.1.3 Вказівки	6
2.1.3.1 PSP	6
2.1.3.2 Segments	6
2.1.3.3 Символ	6
2.1.3.4 Код помилки	7
2.1.4 Результат виконання	7
2.2 COM Програма без модульних процедур	8
2.2.1 Код програми	8
2.2.2 Дизасемблювання	10
2.2.3 Вказівки	12
2.2.3.1 PSP	12
2.2.3.2 Segments	13
2.2.3.3 Символ	15
2.2.3.4 Код помилки	15
2.2.4 Результат виконання	15
2.3 СОМ Програма з модульними процедурами	15
2.3.1 Код програми	15
2.3.2 Дизасемблювання	18
2.3.3 Вказівки	19
2.3.3.1 PSP	19
2.3.4.2 Segments	20
2.3.3.3 Символ	21
2.3.3.4 Код помилки	21
2.3.4 Результат виконання	21
2.4 Виведення кодів помилок	21
2.5 ВАТ файл	22
3 Висновки	22

1 Завдання

1.1 Текст завдання

- 1. Провести практичне ознайомлення з компілятором TASM та лінкувальником TLINK і ознайомитися з їх аргументами, параметрами та способом використання.
- 2. У текстовому редакторі створити початковий код на мові асемблера для ехе-програм з наступними вимогами до коду:
 - реалізувати виведення до консолі будь-якого друкованого символу;
- на етапі завершення виконання програми реалізувати повернення до операційної системи коду помилки зі значенням згідно варіанту як у ЛР №1 (регістр ВХ);
 - обов'язково у потрібних місцях коду прописувати коментарі;
- у якості імен сегментів та точки входу використовувати зрозумілі, не довгі але оригінальні назви.
- 3. У текстовому редакторі створити початковий код на мові асемблера для сотпрограм з наступними вимогами до коду:
 - створити дві версіі коду сот-програм:
 - а) без використання модульних процедур;
 - б) з використанням модульної процедури;
 - реалізувати виведення до консолі будь-якого друкованого символу;
- на етапі завершення виконання програми реалізувати повернення до операційної системи коду помилки зі значенням згідно варіанту як у ЛР №1 (регістр ВХ);
 - обов'язково у потрібних місцях коду прописувати коментарі;
- у якості імен сегментів та точки входу використовувати зрозумілі, не довгі але оригінальні назви.
- 4. Над створеними початковими кодами програм виконати процеси компіляції, лінкування та позбавитись від помилок (якщо вони є).
 - 5. Виконати демонстрацію структури програми у відлагоджувачі DEBUG:
 - дизасемблювати виконуваний код програми;
- вказати на місця початку/кінця наступних частин програми: PSP, CODE Segment, DATA Segment, STACK Segment;
- знайти у виконуваному коді символ, який повинен бути виведено до консолі;
- знайти у виконуваному коді значення, яке має бути передано до змінної коду помилки операційної системи.
- 6. Виконати запуск програм у консолі з демонстрацією виведеного символу та кодом помилки. Перевірити коректність значення коду помилки згідно варіанту.
- 7. Оформити звіт згідно ДСТУ 3008:2015 та завантажити поряд з ZIP-архівом власноруч створених початкових кодів програм. Документ звіту та архів повинні мати у назві Група_Прізвище_НомерЛР.

2 Хід роботи

2.1 EXE Програма

2.1.1 Код програми

```
.model small ; set program model as small
.stack 100h ; set stack size to 100h
sseg segment para stack 'stack' ; declare stack segment
   db 256 dup(?) ; reserve memory for stack
sseg ends ; end stack segment
dseg segment para public 'data' ; declare data segment
   symbol db 'X' ; declare symbol variable
dseg ends ; end data segment
cseg segment para public 'code' ; declare code segment
   assume cs:cseg, ss:sseg, es:nothing; set segment register to
corresponding ones
start: ; declare program entry point
   assume ds:dseg ; set data segment register
   mov bx, dseg ; add data segment to bx register
   mov ds, bx ; set ds register to bx register
   call main ; call main function
   mov ah, 4Ch ; exit to OS
   mov bl, 6Ch ; set error code to 108 in hex
   int 21h  ; call interrupt
main proc near ; declare main function
   mov dl, symbol ; load symbol into dl register
   mov ah, 02h ; output symbol to stdout
   int 21h ; call interrupt
```

```
ret ; stop function execution
main endp ; end main function

cseg ends ; end code segment
end start ; end program execution
```

2.1.2 Дизасемблювання

```
Z:\ASM\TASM>debug exe.exe
·u
0B4F:0000
           BB510B
                                    BX,0B51
                           MOV
0B4F:0003 8EDB
                                    DS, BX
                           MOV
0B4F:0005 F80600
                           CALL
                                    000E
0B4F : 0008
                                    AH,4C
          B44C
                           MOV
0B4F:000A B36C
                           MOV
                                    BL,60
                                    21
           CD21
0B4F:000C
                           TNT
           8A160000
                                    DL,[0000]
                           MOV
                                    AH, 02
0B4F:0012 B402
                           MOV
           CD21
                                    21
           0000
                                    [BX+SI], AL
           0000
                           ADD
                                    [BX+SI],AL
                                    [BX+SI1,AL
0B4F:001B
           0000
                           add
0B4F:001D
                           ADD
           0000
                                    [BX+SI],AL
                                    [BX+SI-01], BL
0B4F:001F
           0058FF
                           ADD
```

2.1.3 Вказівки

2.1.3.1 PSP

2.1.3.2 Segments

Z:\ASM\TAS	SM>debug e	exe.exe	
-U OD/F.0000	DDE10D	MOU	DV ADE1
0B4F:0000 0B4F:0003		MOV Mov	BX,0B51 DS,BX
0B4F:0005	E80600	CALL	000E
0B4F:0008		MOV	AH, 4C
0B4F:000A		MOV	BL,60
0B4F:000C 0B4F:000E		INT MOV	21 DL.[0000]
0B4F:0012		йŏv	AH,02
0B4F:0014		INT	21
0B4F:0016		RET	FRU OT LOI
0B4F:0017		ADD	[BX+SI],AL
0B4F:0019	0000	ADD	[BX+SI],AL
0B4F:001B		ADD	[BX+SI],AL
0B4F:001D 0B4F:001F		ADD ADD	[BX+SI],AL [BX+SI-01],BL

2.1.3.3 Символ

0B4F:000E	8A160000	MOV	DL.[0000]
-----------	----------	-----	-----------

0B4F:000A B36C MOV BL,6C

2.1.4 Результат виконання

```
Z:\ASM\TASM>TASM.EXE exe.asm
Turbo Assembler Version 3.2i Copyright (c) 1988, 1992 Borland International
Serial No: Tester:
Assembling file: exe.asm
Error messages: None
Warning messages: None
Passes: 1
Remaining memory: 455k

Z:\ASM\TASM>TLINK.EXE EXE.OBJ
Turbo Link Version 2.0 Copyright (c) 1987, 1988 Borland International
Z:\ASM\TASM>EXE.EXE
```

2.2 COM Програма без модульних процедур

2.2.1 Код програми

```
.model tiny ; set program model as tiny
cseg segment para public 'code'; declare code segment
   assume cs:cseg, ds:cseg, ss:cseg, es:nothing ; set each code
segment to code seg as it is the only segment in the program
   org 100h; start loading the first instruction at 100h
start: ; declare program entry point
   symbol db 'U' ; declare a symbol to output
   xor ax, ax ; clear ax register
   mov dl, symbol ; load symbol to stdout
   mov ah, 02h ; output symbol to stdout
   int 21h ; call interrupt
   mov dl, 'k' ; load symbol to stdout
   mov ah, 02h ; output symbol to stdout
   int 21h  ; call interrupt
   int 21h  ; call interrupt
   mov dl, 'a' ; load symbol to stdout
mov ah, 02h ; output symbol to stdout
   int 21h ; call interrupt
   mov dl, 'i' ; load symbol to stdout
   mov ah, 02h ; output symbol to stdout
   int 21h ; call interrupt
```

```
mov ah, 02h ; output symbol to stdout
int 21h ; call interrupt

mov dl, 'e' ; load symbol to stdout
mov ah, 02h ; output symbol to stdout
int 21h ; call interrupt

mov dl, 10 ; set dl register to new line
mov ah, 02h ; output it to stdout
int 21h ; call interrupt
mov dl, 13 ; set dl register to carret return
mov ah, 02h ; output it to stdout
int 21h ; call interrupt

mov ah, 04h ; exit to 05
mov bl, 6Ch ; set error code to 108 in hex
int 21h ; call interrupt

cseg ends ; close segment
end start ; end program execution
```

2.2.2 Дизасемблювання

	SM>debug comno	. com	
-u 0B3F • 01 00	55	PHSH	BP
0B3F : 0101	55 3300	YOR.	ĂX, AX
0B3F : 0101	8A160001	MUN	DL,[0100]
0B3F:0107		MOV	
0B3F:0109			21
0B3F:010B	B26B	ΜÖΫ	DL,6B
0B3F:010D		ΜŎΫ	AH,02
0B3F:010F			21
0B3F:0111		MOV	DL, 72
0B3F:0113		MOV	ĀH, 02
0B3F:0115			21
0B3F:0117	B261	MOV	DL,61
0B3F:0119	B402	MOV	AH,02
0B3F:011B		INT	21
0B3F:011D	B269	MOV	DL,69
0B3F:011F	B402	MOV	AH,02
-u			
0B3F:0121		INT	21
0B3F:0123		MOV	DL,6E
0B3F:0125		MOA	AH,02
0B3F:0127		INT	21
0B3F:0129	B265	MOV	DL,65
0B3F:012B	B402	ΜΟΛ	AH,02
0B3F:012D	CD21	INT	21
0B3F:012F	B20A	MOV	DL, ØA
0B3F:0131	B402	MOV	AH, 02
0B3F:0133	CD21	INT	21 DL OD
0B3F:0135	B20D	MOV	DL, ØD
0B3F:0137 0B3F:0139	B402 CD21	MOV INT	AH,02 21
0B3F:013B	B44C	MOA	AH,4C
0B3F:013D	B36C	MOV	нп, 4С ВL,6С
0B3F:013F	CD21	INT	21
ACTA: JEGG	CDZI	THI	71

2.2.3 Вказівки

2.2.3.1 PSP

```
Z:\ASM\TASM>debug comno.com
-d 0 100
                                                                                        CD 20 FF
54 05 17
FF FF FF
14 0A 14
05 00 00
CD 21 CB
20 20 20
20 20 20
00 0D 63
3D 41 30
20 65 78
56 44 4D
69 6E 76
00 00 00
                                                                00 9A F0
54 05 43
FF FF FF
18 00 3F
00 00 00
20 20 20
20 20 20
6D 6E 6F
64 64 72
6D 70 6C
20 73 70
6C 69 64
00 00 00
0B3F:0000
0B3F:0010
                                                        9F
03
                              0B3F:0010
0B3F:0020
0B3F:0030
0B3F:0040
0B3F:0050
0B3F:0060
0B3F:0080
0B3F:0090
0B3F:00B0
0B3F:00B0
0B3F:00E0
0B3F:00E0
                                                                                                                                                             01
                                                                                                                                                             00
                                                                                                                                                             00
                                                                                                                                                             20
20
00
52
72
54
20
                                                                                                                                                                            ..comno.com.BJ.R
=A0.ddress. For
example:. on NT
VDM, specify an
invalid. only...
                                                                                                                                                              ōŏ
                                                                                                                                                             00
                                                                                                                                                             00
                                                                                                                                                                            U
                                                                                                                           00 00 00 00 00
 0B3F:0100
                                55
```

	SM>debug comno	. com	
-u 0B3F • 01 00	55	PHSH	BP
0B3F : 0101	55 3300	YOR.	ĂX, AX
0B3F : 0101	8A160001	MUN	DL,[0100]
0B3F:0107		MOV	
0B3F:0109			21
0B3F:010B	B26B	ΜÖΫ	DL,6B
0B3F:010D		ΜŎΫ	AH,02
0B3F:010F			21
0B3F:0111		MOV	DL, 72
0B3F:0113		MOV	ĀH, 02
0B3F:0115			21
0B3F:0117	B261	MOV	DL,61
0B3F:0119	B402	MOV	AH,02
0B3F:011B		INT	21
0B3F:011D	B269	MOV	DL,69
0B3F:011F	B402	MOV	AH,02
-u			
0B3F:0121		INT	21
0B3F:0123		MOV	DL,6E
0B3F:0125		MOA	AH,02
0B3F:0127		INT	21
0B3F:0129	B265	MOV	DL,65
0B3F:012B	B402	ΜΟΛ	AH,02
0B3F:012D	CD21	INT	21
0B3F:012F	B20A	MOV	DL, ØA
0B3F:0131	B402	MOV	AH, 02
0B3F:0133	CD21	INT	21 DL OD
0B3F:0135	B20D	MOV	DL, ØD
0B3F:0137 0B3F:0139	B402 CD21	MOV INT	AH,02 21
0B3F:013B	B44C	MOA	AH,4C
0B3F:013D	B36C	MOV	нп, 4С ВL,6С
0B3F:013F	CD21	INT	21
ACTA: JEGG	CDZI	THI	71

0B3F:0103 8A160001 MOV DL,[0100]

2.2.3.4 Код помилки

0B3F:013D B36C MOV BL.6C

2.2.4 Результат виконання

```
Z:\ASM\TASM>TASM.EXE comNo.asm
Turbo Assembler Version 3.2i Copyright (c) 1988, 1992 Borland International Serial No: Tester:

Assembling file: comNo.asm
Error messages: None
Warning messages: None
Passes: 1
Remaining memory: 455k

Z:\ASM\TASM>TLINK.EXE /t COMNO.OBJ
Turbo Link Version 2.0 Copyright (c) 1987, 1988 Borland International

Z:\ASM\TASM>COMNO.COM
```

2.3 СОМ Програма з модульними процедурами

2.3.1 Код програми

```
.model tiny
              ; set program model as tiny
cseg segment para public 'code' ; Declare code segment
   assume cs:cseg, ds:cseg, ss:cseg, es:nothing
                                                 ; set each code
segment to code seg as it is the only segment in the program
   org 100h
               ; start loading the first instruction at 100h
start: ; declare program entry point
   call main ; call main function
   mov ah, 04Ch
                   ; exit to OS
   mov bl, 6Ch
                   ; set error code to 108 in hex
   int 21h  ; call interrupt
   symbol db 'U'
                   ; declare a symbol to output
```

```
main proc near ; declare main procedure
   call newline ; call newline function
   xor ax, ax ; clear ax register
   mov dl, symbol ; load symbol to stdout
   mov ah, 02h ; output symbol to stdout
   int 21h  ; call interrupt
   mov dl, 'k' ; load symbol to stdout
   mov ah, 02h ; output symbol to stdout
   int 21h ; call interrupt
   mov dl, 'r' ; load symbol to stdout
   mov ah, 02h     ; output symbol to stdout
   int 21h ; call interrupt
   mov ah, 02h ; output symbol to stdout
   int 21h  ; call interrupt
   mov dl, 'i' ; load symbol to stdout
   int 21h  ; call interrupt
   mov ah, 02h ; output symbol to stdout
   int 21h ; call interrupt
   int 21h  ; call interrupt
   call newline ; call newline function
   mov ah, 02h ; output symbol to stdout
```

```
int 21h ; call interrupt
   int 21h  ; call interrupt
   call newline ; call newline function
   ret ; end function execution
main endp ; end procedure
newline proc near ; declare modular procedure
   mov dl, 10 ; set dl register to new line
   mov ah, 02h ; output it to stdout
   int 21h  ; call interrupt
   mov dl, 13 ; set dl register to carret return
   mov ah, 02h ; output it to stdout
   int 21h ; call interrupt
   ret ; end function execution
newline endp ; end procedure
cseg ends ; close segment
end start ; end program execution
```

2.3.2 Дизасемблювання

7. \ 0 eu \ T0	CM2 dabug aamua	G 66#	
	SM>debug comye	S.COM	
-u 0B3F:0100	E90700	CALL	010A
0B3F:0100		MOV	
0B3F:0105			BL,6C
0B3F:0103		INT	21
0B3F:0107		PUSH	BP
0D3F:0107	E84100	COLL	014E
0B3F:010D	8A160901	XOR	AX,AX
0B3F:0113		MOV	AH,02
0B3F: 0115		INT	21 DL CD
0B3F:0117		VOM	DL,6B
0B3F:0119		MOV	AH,02
0B3F:011B		INT	21
0B3F:011D		MOV	DL,72
0B3F:011F	D4VZ	MOV	AH,02
-u appr. a191	CD21	THE	21
0B3F: 0121		INT	21 DL 61
0B3F: 0123		MOV	DL,61
0B3F: 0125		MOV	AH, 02
0B3F: 0127		INT	21
0B3F:0129	B269	MOV	DL,69
0B3F: 012B		MOV	AH, 02
0B3F:012D 0B3F:012F	CD21 B26E	INT MOV	21 DL 65
0B3F:012F			DL,6E AH,02
0B3F:0131	B402 CD21	MOV	"
0B3F:0135		INT	21 DL 65
0B3F:0137	B265	MOV	DL,65 AH,02
0B3F:0139	B402 CD21	MOV INT	нп, од 21
0B3F:013B		CALL	014E
0B3F:013E		MOV	
			DL,3C ∩⊔ a2
0B3F:0140	B402	MOV	AH,02

CD21	INT	21
B233	MOV	DL,33
B402	MOV	AH, 02
CD21	TNT	21
E80100	CALL	014E
C3	RFT	0112
		DL,0A
B402		AH, 02
CD21	INT	21
B20D	MOV	DL, ØD
B402		AH,02
CD21		21
C3	RET	
3002	CMP	AL,02
7506	JNZ	0165
26	ES:	
895504	MOV	[DI+04],DX
	B233 B402 CD21 E80100 C3 B20A B402 CD21 B402 CD21 C3 3C02 7506	B233 MOV B402 MOV CD21 INT E80100 CALL C3 RET B20A MOV CD21 INT B20D MOV B402 MOV CD21 INT C3 RET 3C02 CMP 7506 JNZ ES:

2.3.3 Вказівки

2.3.3.1 PSP

```
-d 0 100
0B3F:0000
0B3F:0010
0B3F:0020
0B3F:0030
0B3F:0050
0B3F:0060
0B3F:0070
0B3F:0080
0B3F:0080
0B3F:0080
0B3F:0080
                                                                                                                                                                                           00 9A F0 FE-1D F0

54 05 43 05-01 01

FF FF FF FF-FF FF

18 00 3F 0B-FF FF

100 00 00 00-00 00

100 00 00 00-00 00

120 20 20 20-00 00

120 20 20 20-00 00

6D 79 65 73-2E 63

164 64 72 65-73 73

16D 70 6C 65-3A 0D

120 73 70 65-63 69

160 00 00 00-00 00

100 00 00 00-00 00

100 00 00 00-00 00
                                                                                              CD
54
                                                                                                                  20 FF 9F 05 17 03 FF FF FF FF 0A 14 00 00 21 CB 00 20 20 20 20 20 20 20 65 78 61 44 4D 2C 6E 76 61 00 00 00 00 00 00 00 00
                                                                                                                                                                                                                                                                                                                                                                                                           54
02
01
                                                                                                                                                                                                                                                                                                                                                                                                                                  05 8A 03
FF FF FF
08 4C 01
00 00 00
00 00 00
20 20 20
20 20 20
20 20 20
54 45 52
46 6F 72
20 4E 54
61 6E 20
2E 0D 00
00 00 00
00 00 00
                                                                                                                                                                                                                                                                                                                                                                                 03
00
FF
00
00
00
6D
6D
6C
00
00
                                                                                                                                                                                                                                                                                                                                                         4F
01
FF
00
00
00
                                                                                              FF
14
05
CD
20
20
3D
                                                                                                                                                                                                                                                                                                                                                                                                             ÕÕ
                                                                                                                                                                                                                                                                                                                                                                                                          00
00
00
00
20
6E
20
79
00
00
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                ..comyes.com.TER
=A0.ddress. For
example:. on NT
VDM, specify an
invalid. only...
                                                                                                                                                                                                                                                                                                                                                         6F
2E
20
66
6E
00
00
                                                                                              20
56
69
00
00
   0B3F:00E0
0B3F:00F0
                                                                                                                                                                                                                                                                                                                                                                                   00
   0B3F:0100
                                                                                              E8
```

Z.J.+.Z Ocginicitis			
Z:\ASM\TAS	SM>debug_comye:	S.COM	
-u			
0B3F:0100	E80700	CALL	010A
0B3F:0103	B44C	MOV	AH,4C
0B3F:0105	B36C	MOV	BL,6C
0B3F:0107	CD21	INT	21
0B3F:0109	55	PUSH	BP
0B3F:010A	E84100	CALL	014E
0B3F:010D		XOR	
0B3F:010F	8A160901		
0B3F:0113		MOV	AH,02
0B3F:0115		INT	21
0B3F:0117		MOV	DL,6B
0B3F:0119		MOV	ĀH, 02
0B3F:011B		INT	21
0B3F:011D			DL, 72
0B3F:011F		MÖV	ÃH, 02
-u	5.52		, 52
0B3F:0121	CD21	INT	21
0B3F:0123		MOV	
0B3F:0125		ЙŎV	ÃH, 02
0B3F:0127		INT	21
0B3F:0129		ΜÖΫ	DL, 69
0B3F:012B	B402	ΜŎŸ	AH, 02
0B3F:012D	CD21	INT	21
0B3F:012F	B26E	ΜÖŸ	DL,6E
0B3F:0131		ΜŎΫ	AH, 02
0B3F:0133	CD21	INT	21
0B3F:0135	B265	ΜÖV	DL, 65
0B3F:0137	B402	MOV	AH,02
0B3F:0139	CD21	INT	21
0B3F:013B	E81000	ĊALL	014E
0B3F:013E	B23C	MOV	DL,3C
0B3F:0140	B402	MOV	AH,02
ODDI . 0140	0402	MOA	nii, oz

0B3F:010F 8A160901 MOV DL,[0109]

2.3.3.4 Код помилки

0B3F:0105 B36C MOV BL,6C

2.3.4 Результат виконання

Z:\ASM\TASM>TASM.EXE comYes.asm
Turbo Assembler Version 3.2i Copyright (c) 1988, 1992 Borland International Serial No: Tester:

Assembling file: comYes.asm
Error messages: None
Warning messages: None
Passes: 1
Remaining memory: 455k

Z:\ASM\TASM>TLINK.EXE /t COMYES.OBJ
Turbo Link Version 2.0 Copyright (c) 1987, 1988 Borland International

Z:\ASM\TASM>COMYES.COM

Ukraine
<3

2.4 Виведення кодів помилок

```
Z:\ASM\TASM>test.bat
EXE Program:
X88
------
COM Program without Function:
Ukraine
13
-----
COM Program with Function:
Ukraine
COM Program with Function:
```

2.5 ВАТ файл

```
@echo off
echo EXE Program:
exe.exe
echo %ERRORLEVEL%
echo ————
echo COM Program without Function:
comNo.com
echo %ERRORLEVEL%
echo ————
echo COM Program with Function:
comYes.com
```

3 Висновки

Таким чином, ми ознайомились зі структурою програм на мові асемблера і навели приклад її застосування