Зображення, що містить текст

Автоматично згенерований опис

Міністерство освіти і науки України

Національний технічний університет України

“Київський політехнічний інститут імені Ігоря Сікорського”

Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

**Комп’ютерний практикум №1**

Системне програмне забезпечення

**Тема:** Створення програм на асемблері

Виконав Перевірив:

студент групи ІП-11: Лісовиченко О. І

Панченко С. В.

Київ 2023

Зміст

[1 Мета комп’ютерного практикуму 6](#__RefHeading___Toc3319_204808571)

[2 Завдання 7](#__RefHeading___Toc5937_3437555080)

[3 Текст програми 8](#__RefHeading___Toc4957_2404284615)

[4 Вміст .lst файлу 9](#__RefHeading___Toc4959_2404284615)

[5 Вміст .map файлу 10](#__RefHeading___Toc4961_2404284615)

[6 Схема функціонування програми 16](#__RefHeading___Toc4963_2404284615)

[7 Вікно DUMP 19](#__RefHeading___Toc4965_2404284615)

[7.1 Зображення вікно DUMP до виконання 19](#__RefHeading___Toc4967_2404284615)

[7.2 Зображення вікно DUMP після виконання 19](#__RefHeading___Toc4969_2404284615)

[8 Висновок 20](#__RefHeading___Toc4971_2404284615)

# Мета комп’ютерного практикуму

Навчитись створювати програми на асемблері.

# Завдання

1. Для програми, наведеної вище, створити файл типу .asm. Ця програма не має засобів виводу даних, тому правильність її виконання треба перевірити за допомогою td.exe.
2. Скомпілювати програму, включивши потрібні опції для налагоджувача та створення файлу лістингу типу .lst.
3. Ознайомитись зі структурою файлу .lst. За вказівкою викладача, для певної команди асемблера розглянути структуру машинної команди і навести її у звіті.
4. Скомпонувати .obj-файл програми. Включити опції для налагодження та створення .map-файлу.
5. Занести до звіту адреси початку та кінця всіх сегментів з .map-файлу.
6. Завантажити до налагоджувача td.exe одержаний .exe-файл програми.
7. У вікні CPU у полі DUMP знайти початкову адресу сегмента даних та записати його до звіту. Знайти масиви SOURCE та DEST. Дані у масиві SOURCE подаються у шістнадцятковій системі.
8. У покроковому режимі за допомогою клавіші F7 виконати програму. Одержані результати у масиві DEST показати викладачеві.

# Текст програми

%include "io64.inc"

bits 64

segment .data ; a.k.a DS - Data Segment

SOURCE: db 10, 20, 30, 40

segment .bss

DEST: resb 4

segment .text ; a.k.a CS - Code Segment

global CMAIN

CMAIN:

mov rbp, rsp; for correct debugging

xor rax, rax

mov byte [DEST], 0

mov byte [DEST + 1], 0

mov byte [DEST + 2], 0

mov byte [DEST + 3], 0

mov al, byte [SOURCE]

mov byte [DEST + 3], al

mov al, byte [SOURCE + 1]

mov byte [DEST + 2], al

mov al, byte [SOURCE + 2]

mov byte [DEST + 1], al

mov al, byte [SOURCE + 3]

mov byte [DEST], al

mov rax, 60 ; system call for exit

xor rdi, rdi ; exit code 0

syscall

# Вміст .lst файлу

1

2 bits 64

3 segment .data ; a.k.a DS - Data Segment

4 00000000 0A141E28 SOURCE: db 10, 20, 30, 40

5 segment .bss

6 00000000 ???????? DEST: resb 4

7 segment .text ; a.k.a CS - Code Segment

8 global \_start

9 \_start:

10 00000000 4889E5 mov rbp, rsp; for correct debugging

11

12 00000003 4831C0 xor rax, rax

13

14 00000006 C60425[00000000]00 mov byte [DEST], 0

15 0000000E C60425[01000000]00 mov byte [DEST + 1], 0

16 00000016 C60425[02000000]00 mov byte [DEST + 2], 0

17 0000001E C60425[03000000]00 mov byte [DEST + 3], 0

18

19 00000026 8A0425[00000000] mov al, byte [SOURCE]

20 0000002D 880425[03000000] mov byte [DEST + 3], al

21

22 00000034 8A0425[01000000] mov al, byte [SOURCE + 1]

23 0000003B 880425[02000000] mov byte [DEST + 2], al

24

25 00000042 8A0425[02000000] mov al, byte [SOURCE + 2]

26 00000049 880425[01000000] mov byte [DEST + 1], al

27

28 00000050 8A0425[03000000] mov al, byte [SOURCE + 3]

29 00000057 880425[00000000] mov byte [DEST], al

30

31 0000005E B83C000000 mov rax, 60 ; system call for exit

32 00000063 4831FF xor rdi, rdi ; exit code 0

33 00000066 0F05 syscall

# Вміст .map файлу

Memory Configuration

Name Origin Length Attributes

\*default\* 0x0000000000000000 0xffffffffffffffff

Linker script and memory map

LOAD release/release.o

[!provide] PROVIDE (\_\_executable\_start = SEGMENT\_START ("text-segment", 0x400000))

0x00000000004000e8 . = (SEGMENT\_START ("text-segment", 0x400000) + SIZEOF\_HEADERS)

.interp

\*(.interp)

.note.gnu.build-id

\*(.note.gnu.build-id)

.hash

\*(.hash)

.gnu.hash

\*(.gnu.hash)

.dynsym

\*(.dynsym)

.dynstr

\*(.dynstr)

.gnu.version

\*(.gnu.version)

.gnu.version\_d

\*(.gnu.version\_d)

.gnu.version\_r

\*(.gnu.version\_r)

.rela.dyn 0x00000000004000e8 0x0

\*(.rela.init)

\*(.rela.text .rela.text.\* .rela.gnu.linkonce.t.\*)

\*(.rela.fini)

\*(.rela.rodata .rela.rodata.\* .rela.gnu.linkonce.r.\*)

\*(.rela.data .rela.data.\* .rela.gnu.linkonce.d.\*)

\*(.rela.tdata .rela.tdata.\* .rela.gnu.linkonce.td.\*)

\*(.rela.tbss .rela.tbss.\* .rela.gnu.linkonce.tb.\*)

\*(.rela.ctors)

\*(.rela.dtors)

\*(.rela.got)

.rela.got 0x00000000004000e8 0x0 release/release.o

\*(.rela.bss .rela.bss.\* .rela.gnu.linkonce.b.\*)

\*(.rela.ldata .rela.ldata.\* .rela.gnu.linkonce.l.\*)

\*(.rela.lbss .rela.lbss.\* .rela.gnu.linkonce.lb.\*)

\*(.rela.lrodata .rela.lrodata.\* .rela.gnu.linkonce.lr.\*)

\*(.rela.ifunc)

.rela.plt 0x00000000004000e8 0x0

\*(.rela.plt)

[!provide] PROVIDE (\_\_rela\_iplt\_start = .)

\*(.rela.iplt)

.rela.iplt 0x00000000004000e8 0x0 release/release.o

[!provide] PROVIDE (\_\_rela\_iplt\_end = .)

.relr.dyn

\*(.relr.dyn)

0x0000000000401000 . = ALIGN (CONSTANT (MAXPAGESIZE))

.init

\*(SORT\_NONE(.init))

.plt 0x0000000000401000 0x0

\*(.plt)

\*(.iplt)

.iplt 0x0000000000401000 0x0 release/release.o

.plt.got

\*(.plt.got)

.plt.sec

\*(.plt.sec)

.text 0x0000000000401000 0x68

\*(.text.unlikely .text.\*\_unlikely .text.unlikely.\*)

\*(.text.exit .text.exit.\*)

\*(.text.startup .text.startup.\*)

\*(.text.hot .text.hot.\*)

\*(SORT\_BY\_NAME(.text.sorted.\*))

\*(.text .stub .text.\* .gnu.linkonce.t.\*)

.text 0x0000000000401000 0x68 release/release.o

0x0000000000401000 \_start

\*(.gnu.warning)

.fini

\*(SORT\_NONE(.fini))

[!provide] PROVIDE (\_\_etext = .)

[!provide] PROVIDE (\_etext = .)

[!provide] PROVIDE (etext = .)

0x0000000000402000 . = ALIGN (CONSTANT (MAXPAGESIZE))

0x0000000000402000 . = SEGMENT\_START ("rodata-segment", (ALIGN (CONSTANT (MAXPAGESIZE)) + (. & (CONSTANT (MAXPAGESIZE) - 0x1))))

.rodata

\*(.rodata .rodata.\* .gnu.linkonce.r.\*)

.rodata1

\*(.rodata1)

.eh\_frame\_hdr

\*(.eh\_frame\_hdr)

\*(.eh\_frame\_entry .eh\_frame\_entry.\*)

.eh\_frame

\*(.eh\_frame)

\*(.eh\_frame.\*)

.gcc\_except\_table

\*(.gcc\_except\_table .gcc\_except\_table.\*)

.gnu\_extab

\*(.gnu\_extab\*)

.exception\_ranges

\*(.exception\_ranges\*)

0x0000000000402000 . = DATA\_SEGMENT\_ALIGN (CONSTANT (MAXPAGESIZE), CONSTANT (COMMONPAGESIZE))

.eh\_frame

\*(.eh\_frame)

\*(.eh\_frame.\*)

.gnu\_extab

\*(.gnu\_extab)

.gcc\_except\_table

\*(.gcc\_except\_table .gcc\_except\_table.\*)

.exception\_ranges

\*(.exception\_ranges\*)

.tdata 0x0000000000402000 0x0

[!provide] PROVIDE (\_\_tdata\_start = .)

\*(.tdata .tdata.\* .gnu.linkonce.td.\*)

.tbss

\*(.tbss .tbss.\* .gnu.linkonce.tb.\*)

\*(.tcommon)

.preinit\_array 0x0000000000402000 0x0

[!provide] PROVIDE (\_\_preinit\_array\_start = .)

\*(.preinit\_array)

[!provide] PROVIDE (\_\_preinit\_array\_end = .)

.init\_array 0x0000000000402000 0x0

[!provide] PROVIDE (\_\_init\_array\_start = .)

\*(SORT\_BY\_INIT\_PRIORITY(.init\_array.\*) SORT\_BY\_INIT\_PRIORITY(.ctors.\*))

\*(.init\_array EXCLUDE\_FILE(\*crtend?.o \*crtend.o \*crtbegin?.o \*crtbegin.o) .ctors)

[!provide] PROVIDE (\_\_init\_array\_end = .)

.fini\_array 0x0000000000402000 0x0

[!provide] PROVIDE (\_\_fini\_array\_start = .)

\*(SORT\_BY\_INIT\_PRIORITY(.fini\_array.\*) SORT\_BY\_INIT\_PRIORITY(.dtors.\*))

\*(.fini\_array EXCLUDE\_FILE(\*crtend?.o \*crtend.o \*crtbegin?.o \*crtbegin.o) .dtors)

[!provide] PROVIDE (\_\_fini\_array\_end = .)

.ctors

\*crtbegin.o(.ctors)

\*crtbegin?.o(.ctors)

\*(EXCLUDE\_FILE(\*crtend?.o \*crtend.o) .ctors)

\*(SORT\_BY\_NAME(.ctors.\*))

\*(.ctors)

.dtors

\*crtbegin.o(.dtors)

\*crtbegin?.o(.dtors)

\*(EXCLUDE\_FILE(\*crtend?.o \*crtend.o) .dtors)

\*(SORT\_BY\_NAME(.dtors.\*))

\*(.dtors)

.jcr

\*(.jcr)

.data.rel.ro

\*(.data.rel.ro.local\* .gnu.linkonce.d.rel.ro.local.\*)

\*(.data.rel.ro .data.rel.ro.\* .gnu.linkonce.d.rel.ro.\*)

.dynamic

\*(.dynamic)

.got 0x0000000000402000 0x0

\*(.got)

.got 0x0000000000402000 0x0 release/release.o

\*(.igot)

0x0000000000402000 . = DATA\_SEGMENT\_RELRO\_END (., (SIZEOF (.got.plt) >= 0x18)?0x18:0x0)

.got.plt 0x0000000000402000 0x0

\*(.got.plt)

.got.plt 0x0000000000402000 0x0 release/release.o

\*(.igot.plt)

.igot.plt 0x0000000000402000 0x0 release/release.o

.data 0x0000000000402000 0x4

\*(.data .data.\* .gnu.linkonce.d.\*)

.data 0x0000000000402000 0x4 release/release.o

.data1

\*(.data1)

0x0000000000402004 \_edata = .

[!provide] PROVIDE (edata = .)

0x0000000000402004 . = .

0x0000000000402004 \_\_bss\_start = .

.bss 0x0000000000402004 0x4

\*(.dynbss)

\*(.bss .bss.\* .gnu.linkonce.b.\*)

.bss 0x0000000000402004 0x4 release/release.o

\*(COMMON)

0x0000000000402008 . = ALIGN ((. != 0x0)?0x8:0x1)

.lbss

\*(.dynlbss)

\*(.lbss .lbss.\* .gnu.linkonce.lb.\*)

\*(LARGE\_COMMON)

0x0000000000402008 . = ALIGN (0x8)

0x0000000000402008 . = SEGMENT\_START ("ldata-segment", .)

.lrodata

\*(.lrodata .lrodata.\* .gnu.linkonce.lr.\*)

.ldata 0x0000000000404008 0x0

\*(.ldata .ldata.\* .gnu.linkonce.l.\*)

0x0000000000404008 . = ALIGN ((. != 0x0)?0x8:0x1)

0x0000000000404008 . = ALIGN (0x8)

0x0000000000402008 \_end = .

[!provide] PROVIDE (end = .)

0x0000000000404008 . = DATA\_SEGMENT\_END (.)

.stab

\*(.stab)

.stabstr

\*(.stabstr)

.stab.excl

\*(.stab.excl)

.stab.exclstr

\*(.stab.exclstr)

.stab.index

\*(.stab.index)

.stab.indexstr

\*(.stab.indexstr)

.comment

\*(.comment)

.gnu.build.attributes

\*(.gnu.build.attributes .gnu.build.attributes.\*)

.debug

\*(.debug)

.line

\*(.line)

.debug\_srcinfo

\*(.debug\_srcinfo)

.debug\_sfnames

\*(.debug\_sfnames)

.debug\_aranges

\*(.debug\_aranges)

.debug\_pubnames

\*(.debug\_pubnames)

.debug\_info

\*(.debug\_info .gnu.linkonce.wi.\*)

.debug\_abbrev

\*(.debug\_abbrev)

.debug\_line

\*(.debug\_line .debug\_line.\* .debug\_line\_end)

.debug\_frame

\*(.debug\_frame)

.debug\_str

\*(.debug\_str)

.debug\_loc

\*(.debug\_loc)

.debug\_macinfo

\*(.debug\_macinfo)

.debug\_weaknames

\*(.debug\_weaknames)

.debug\_funcnames

\*(.debug\_funcnames)

.debug\_typenames

\*(.debug\_typenames)

.debug\_varnames

\*(.debug\_varnames)

.debug\_pubtypes

\*(.debug\_pubtypes)

.debug\_ranges

\*(.debug\_ranges)

.debug\_addr

\*(.debug\_addr)

.debug\_line\_str

\*(.debug\_line\_str)

.debug\_loclists

\*(.debug\_loclists)

.debug\_macro

\*(.debug\_macro)

.debug\_names

\*(.debug\_names)

.debug\_rnglists

\*(.debug\_rnglists)

.debug\_str\_offsets

\*(.debug\_str\_offsets)

.debug\_sup

\*(.debug\_sup)

.gnu.attributes

\*(.gnu.attributes)

/DISCARD/

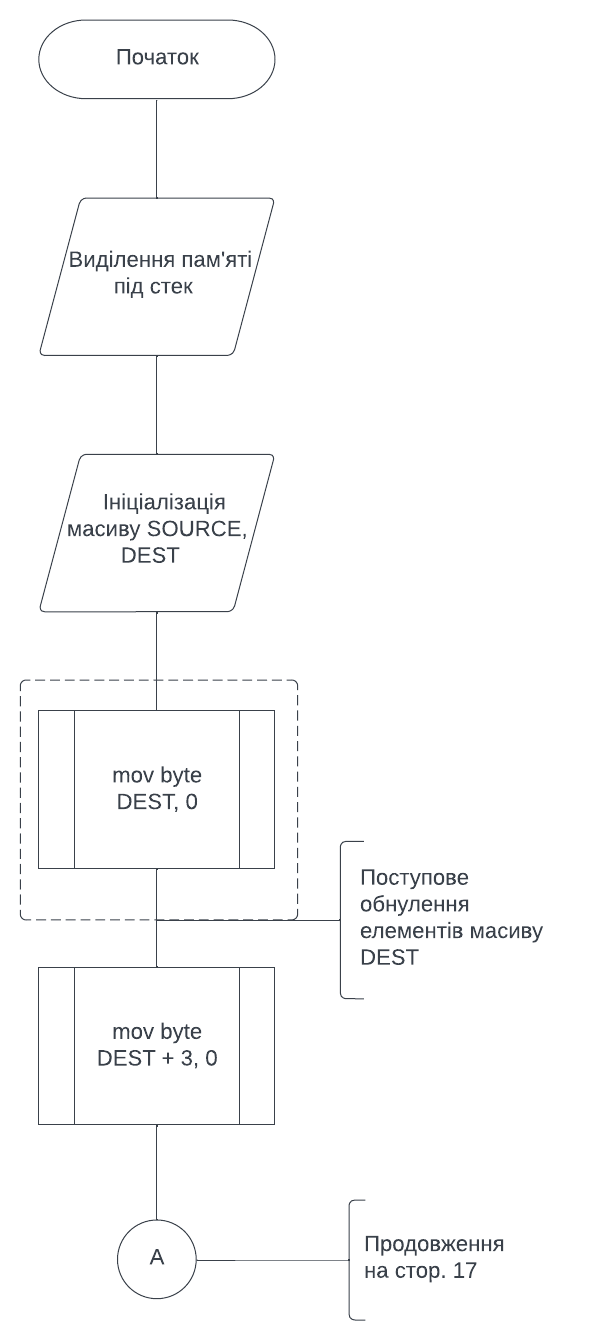
\*(.note.GNU-stack)

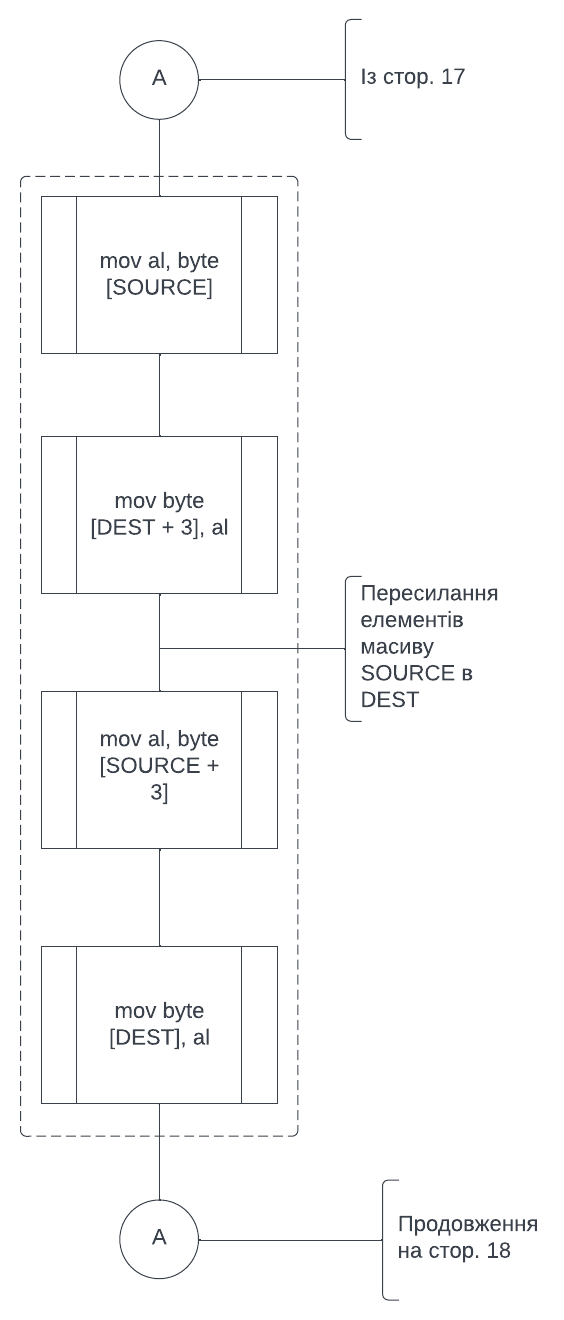
\*(.gnu\_debuglink)

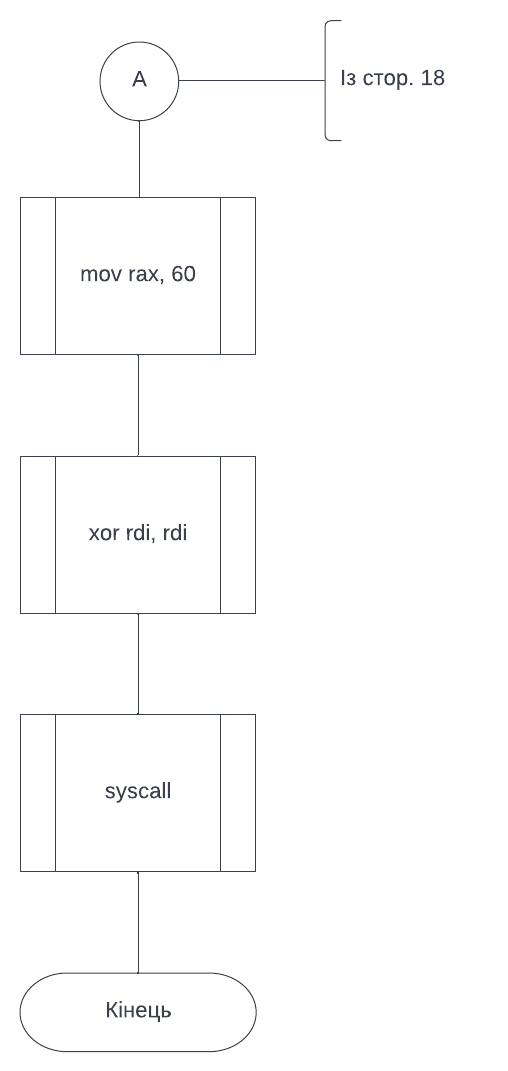
\*(.gnu.lto\_\*)

OUTPUT(release/release.out elf64-x86-64)

# Схема функціонування програми







# Вікно DUMP

## Зображення вікно DUMP до виконання

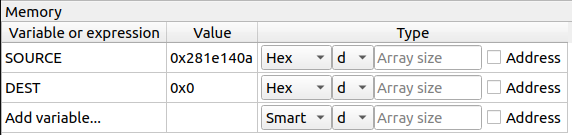
****

Рисунок 7.1 — Вікно DUMP до виконання

## Зображення вікно DUMP після виконання

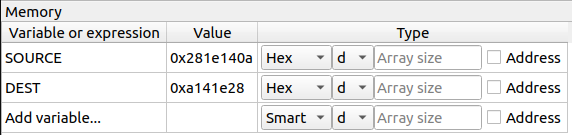
****

Рисунок 7.2 — Вікно DUMP після виконання

# Висновок

1. У IDE SASM було створено файл типу .asm

2. Скомпілював програму, включивши потрібні опції для налагоджувача та створення:

* nasm -felf64 release/release.asm
* файлу лістингу типу .lst.
* nasm -felf64 -l release/release.lst release/release.asm

3. Ознайомився зі структурою файлу .lst. Розглянув структури машинних команд.

4. Після усунення помилок, скомпонував .obj-файл програми, включивши опції для

налагодження та створення .map-файлу.

ld -M -o release/release.out release/release.o > release/release.map

5. Відкрив файл карти пам’яті (.map-файл) та подивився на адреси початку та кінця

всіх сегментів програми.

6. Завантажив програму налагоджувача td.exe та мій одержаний .exe-файл програми.

7. У вікні CPU у полі DUMP подивився на початкову адресу сегмента даних. В

сегменті даних знайшов масиви SOURCE та DEST. Дані у масиві SOURCE подаються

у шістнадцятковій системі.

8. У покроковому режимі за допомогою клавіші F7 виконав програму. Програма

коректно виконує поставлену задачу.