**Міністерство освіти і науки України**

**Національний університет «Запорізька політехніка»**

Кафедра програмних засобів

**ЗВІТ**

Дисципліна «Операційні системи»

Робота №1

Тема «Функції операційної системи MS-DOS: визначення версії ОС»

**Виконав варіант 19**

Студент КНТ-122 Онищенко О. А.

**Прийняли**

Викладач Качан О. І.

2024

[Основи програмування мовою Асемблер 3](#_Toc179967574)

[Робота простих функцій вводу з клавіатури та виводу тексту 4](#_Toc179967575)

[Програма Асемблеру для визначення версії MS DOS 4](#_Toc179967576)

[Склад картки пам’яті – файлу типу МАР 5](#_Toc179967577)

[Робота функції 3306h INT 21h 5](#_Toc179967578)

[Структура PSP 5](#_Toc179967579)

[Структура та образ пам’яті програми ЕХЕ 6](#_Toc179967580)

[Структура програми СОМ 7](#_Toc179967581)

[Використання утиліти MS DOS EXE2BIN 8](#_Toc179967582)

[Поняття операційної системи 9](#_Toc179967583)

[Структура ОС MS DOS 9](#_Toc179967584)

[Завантаження операційної системи 11](#_Toc179967585)

[*Запуск операційної системи MS DOS* 11](#_Toc179967586)

Основи програмування мовою Асемблер

Асемблер – низькорівнева мова. Чим вижче рівень мови програмування, тим більший рівень абстракції. Абстракція – наскільки багато функцій мова «ховає» або виконує за лаштунками без відома програміста. Асемблер вважається низькорівневою мовою бо його інструкції майже точно відповідають вихідному машинному коду, який розуміє комп’ютер.

Оскільки команди мовою асемблер керують процесором, корисно розуміти як комп’ютер працює зсередини.

Запис чисел:

* Шістнадцяткові: суфікс H
* Двійкові: суфікс B
* Десяткові: суфікс D або без суфіксу

При написанні шістнадцяткових констант потрібно дописати 0 на початку аби система не сприйняла число як змінну.

Арифметичні оператори за ступенем прецеденту:

* ( ) дужки
* +. - унарний плюс та мінус
* \*, / множення та ділення
* MOD модуль
* +, - додавання та віднімання

Запис символів відбувається беручи їх до лапок. Можна використовувати як одинарні ('a'), так і подвійні ("T").

Рядки записуються так само у лапках: "32", 'ІСУС ГОСПОДЬ'.

Асемблер має кілька зарезервованих слів:

* Інструкції: MOV, ADD, MUL
* Атрибути: BYTE, WORD

[Джерело Українською](https://elib.lntu.edu.ua/sites/default/files/elib_upload/16,05/other/tema_9__osnovi_programuvannya_movoyu_asembler.pdf), [Джерело Англійською](https://www.cs.emory.edu/~cheung/Courses/255/Syllabus/9-Intel/Resources/Book01-partial/chapt_03.pdf)

Робота простих функцій вводу з клавіатури та виводу тексту

Для введення числа у консоль потрібен наступний код:

MOV AH,1

INT 21H

Для виведення символа до консолі потрібен наступний код:

MOV AH,2

MOV DL, 'T'

INT 21H

[Джерело відповіді](https://stackoverflow.com/questions/2718332/how-can-i-do-input-output-on-a-console-with-masm#:~:text=example%20of%20how%20to%20read%20a%20character%20and%20write%20a%20character)

Програма Асемблеру для визначення версії MS DOS

Програмний код мовою Assembly:

.model small

.stack 100h

.data

vns db 'DOS Version $'

.code

main proc

lea dx,vns

mov ah,9

int 21h

mov ah,30h

int 21h

main endp

end main

Склад картки пам’яті – файлу типу МАР

Структура файлу МАР складається з таких елементів:

1. Inputs: змінні для вхідних даних асемблювання мапи
2. Outputs: змінні з вихідними даними асемблювання мапи
3. Action: масив дій які треба виконати по порядку. Кожна дія має директиву set або create – визначає вихідну змінну.
4. Options: вибір опцій, як от «Включати порожні елементи XML» чи «Обробка порожніх масивів JSON»

[Джерело відповіді](https://www.ibm.com/docs/en/datapower-gateway/10.6.x?topic=action-map-file-structure#assemblymap_structure__structure__title__1:~:text=Accessing%20other%20contexts-,Structure,-In%20addition%20to)

Робота функції 3306h INT 21h

Призначення для отримання версії DOS. Параметри:

* Вхід: ah – 30h
* Вихід: ax – номер версії, більша версія у нижньому байті; bl:cx – серійний номер користувача; bh – номер ОЕМ, або прапорці
* Помилка: al – 00h

[Джерело відповіді](https://www.geoffchappell.com/notes/dos/interrupts/21h/30h/index.htm)

Структура PSP

Program Segment Prefix або Префікс програмного сегмента призначений для збереження поточного вигляд програми. Має наступну структру:

|  |  |
| --- | --- |
| **Зсув** | **Розмір** |
| 00h-01h | 2 байти (код) |
| 02h-03h | 2 байти (слово) |
| 04h | байт |
| 05h-09h | 5 байтів (код) |
| 0Ah-0Dh | 4 байти (дслово) |
| 0Eh-11h | дслово |
| 12h-15h | дслово |
| 16h-17h | слово |
| 18h-2Bh | 20 байт |
| 2Ch-2Dh | слово |
| 2Eh-31h | дслово |
| 32h-33h | слово |
| 34h-37h | дслово |
| 38h-3Bh | дслово |
| 3Ch-3Fh | 4 байти |
| 40h-41h | слово |
| 42h-4Fh | 14 байт |
| 50h-52h | 3 байти (код) |
| 53h-54h | 2 байти |
| 55h-5Bh | 7 байт |
| 5Ch-6Bh | 16 байт |
| 6Ch-7Fh | 20 байт |
| 80h | 1 байт |
| 81h-FFh | 127 байт |

[Джерело відповіді](following%20structure)

Структура та образ пам’яті програми ЕХЕ

Файле ЕХЕ складається з кількох частин:

* Заголовок MS DOS: інформація про переміщення, довжиною 64 байти, належить до системи MS DOS. Призначення – показувати помилки
* Заголовок PE: завантажувач ОС бере інформацію з цього заголовку. Каже куди треба завантажити програму у пам’ять
* Секції: різні параметри файлу - .text, .code, .tests, .rdata, .data, .bss, .rsrc, .debug, .idata, .edata, .reloc

Образ пам’яті програми ЕХЕ виглядає наступним чином:

Зображення, що містить текст, знімок екрана, Шрифт, число

Автоматично згенерований опис

Рисунок 1.1 – Образ пам’яті файлу ЕХЕ

[Джерело](https://eshagalawatta.medium.com/analyzing-structure-of-executable-programs-exe-files-821b2d531a4)

Структура програми СОМ

СОМ файли містять машинний код: без заголовків, таблиць переадресації, тощо. Максимальний об’єм такого файлу 64 КБ.

Завантаження файлу СОМ відбувається так:

1. Відвести максимально можливий обсяг вільної пам’яті
2. Збудувати PSP на початку блоку пам’яті
3. Прочитати СОМ файл з префіксо, передати керування на перший байт прогарми

Структура СОМ файлу виглядає так:

Зображення, що містить текст, знімок екрана, Шрифт, число

Автоматично згенерований опис

Рисунок 1.2 – Структура СОМ файлу

Пояснення:

* ss: сегмент стеку
* sp: покажчик стеку
* cs: сегменту коду
* ip: покажчик команд
* min: мінімальна додаткова пам’ять для файлу
* max: максимальна додаткова пам’ять для файлу

[Джерело](https://antibotan.com/file.html?work_id=501286)

Використання утиліти MS DOS EXE2BIN

Утиліта EXE2BIN призначена для конвертування файлів з формату ЕХЕ у бінарний. Це зменшує розміри файлів та пришвидшує їх завантаження.

Синтаксис команди такий:

EXE2BIN [шлях]назва\_файлу [шлях]назва\_файлу

Перша назва файлу це початковий файл. Якщо ввести без розширення, за замовчуванням вважатиме що файл ЕХЕ.

Друга назва файлу це новий файл, який створить у форматі BIN. Якщо не ввести цю назву, використає однакове ім’я з початковим.

Початковий файл має бути правильним ЕХЕ, резидентна частина (код та дані) не мають перевищувати 64 КБ за обсягом. Не має мати сегменту Stack.

Приклад використання програми EXE2BIN:

exe2bin test.exe new.bin

[Джерело](https://www.upgradenrepair.com/DOScommands/E/dosexe2bin.htm)

Поняття операційної системи

Операціна система (ОС) – набір програм який керує роботою комп’ютера.

ОС виконує такі дії:

* керує роботою апаратного забезпечення
* організовує обмін даних між пристроями
* надає інтерфейс взаємодії користувачеві з комп’ютером
* організовує збережння даних на носіях
* дозволяє виконувати інші програми
* узгоджує та керує роботою програм

[Джерело](https://www.miyklas.com.ua/p/informatica/5-klas/informatciini-protcesi-ta-sistemi-39831/operatciina-sistema-ta-yiyi-interfeis-326915/re-2f181d19-b05b-4c2a-a973-2eb3ce635a9c#:~:text=%D1%82%D0%B0%20%D1%97%D1%97%20%D1%96%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81-,%D0%A2%D0%B5%D0%BE%D1%80%D1%96%D1%8F,-%3A)

Структура ОС MS DOS

ОС Microsoft Disk Operating System (MS DOS) створена для використання на персональних комп’ютерах і є прикладом системи з багаторівневою структурою.

MS DOS розділена на різні шари, кожен з яких виконує свої функції. Такий підхід полегшує взаємодією з системою бо кожен з кожним рівнем можна працювати окремо.

Загальна структура виглядає так:

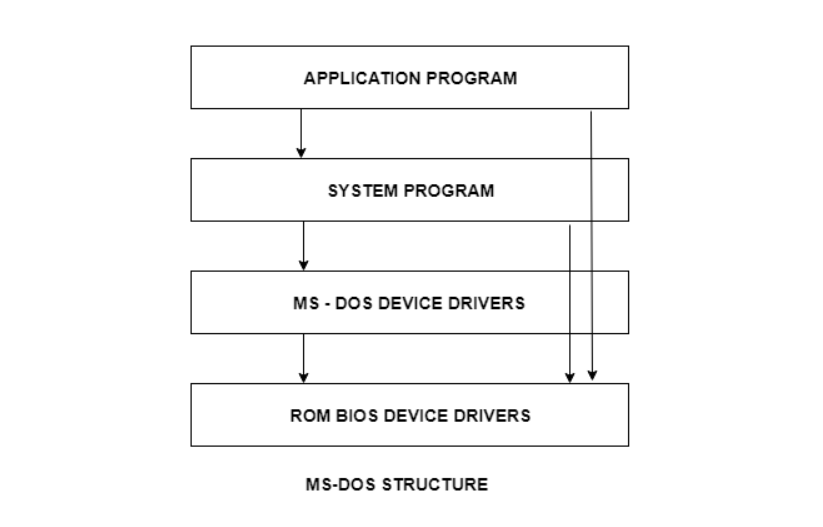


Рисунок 1.3 – Структура MS DOS

Нижче наведено пояснення кожного рівня:

*Application Program* або *Програмне забезпечення* містить програми корисні користувачам. Це рівень звичайних застосунків або прикладних програм.

*System Program* або *Системне забезпечення* надають програми для керування системою. Ці програми зазвичай не використовуються користувачем але допомагають системі взаємодіяти з іншими.

*MS-DOS Device Drivers* або *Драйвери пристроїв MS-DOS* є частиною операційної системи і надають різні драйвери, наприклад, клавіатурі, мишці, жорстким дискам, тощо: різним зовнішнім пристроям.

*ROM BIOS Device Drivers* або *Драйвери пристроїв ROM BIOS* зберігають драйвери на схемах EPROM або EEPROM на материнській плані. Це драйвери основні для запуску комп’ютера. *BIOS* – Basic Input Output System або Базова система вводу виводу.

[Джерело](https://www.tutorialspoint.com/ms-dos-layered-structure)

Завантаження операційної системи

Процес заваннтаження операційної системи на комп’ютері це перенесення операційної системи з вторинної пам’яті в основну. Цей процес виконується BIOSом.

Є два типи запуску системи:

* холодний: звичайний запуск коли комп’ютер не працював, а потім натиснули кнопку живлення
* теплий: виконується при перезавантаженні, коли комп’ютер вже працював і його перезавантажують

[Джерело](https://www.tutorialspoint.com/booting-process-in-dos-operating-system#:~:text=WindowsMicrosoft%20Technologies-,What%20is%20Booting%3F,-Booting%20is%20a)

*Запуск операційної системи MS DOS*

У MS DOS запуск складається з завантаження трьох файлів:

1. io.sys
2. msdos.sys
3. command.com

Процес запуску MS DOS виглядає так:

1. BIOS виконує Power-On Self Test (POST) – перевіряє конфігурації переферії, дивиться чи під’єднані усі пристрої, тощо. Робить це у Read Only Memory (ROM)
2. Шукає Master Boot Record (MBR) або Головний завантажувальний запис на фізичному диску. Якщо не знайдено на жорсткому диску, буде шукати на дисковводі. Якщо не знайдено на дисковводі, буде шукати на CD-ROM. Якщо не знайдено ніде, видасть повідомлення «No Boot Device Found» та зупинить роботу. Якщо завантажувальний запис знайдено, підвантажує Завантажувач операційної системи (Bootstrap Loader) до оперативної пам’яті.
3. Завантажувач підвантажує файл io.sys, потім msdos.sys – це основний файл ОС MS DOS.
4. Файл msdos.sys шукає командний інтерпретатор у файлі config.sys. Якщо знайдено, вантажить його до оперативної пам’яті. Якщо не знайдено, до пам’яті завантажує файл command.com як інтерпретатор команд за замовчуванням для MS DOS.
5. Завантажує і запускає файл autoexec.bat який містить послідовність команд DOS.
6. Показує командний рядок DOS на екрані комп’ютера, виводить літеру завантажувального диску. Якщо бачимо це, операційна система успішно завантажена з диску до оперативної пам’яті.

[Джерело](https://www.tutorialspoint.com/booting-process-in-dos-operating-system#:~:text=Booting%20Process%20in-,DOS%20Operating%20System,-In%20DOS%20(Disc)