Міністерство освіти і науки України  
Національний технічний університет України

«Київський політехнічний інститут ім. І. Сікорського»

Кафедра автоматизації проектування енергетичних процесів і систем

Лабораторна робота №4

з дисципліни «Операційні системи»

«Організація програм \*.ЕХЕ та \*.СОМ»

Варіант №1

**Виконав:**  
студент 1-го курсу, ТЕФ

групи ТР-15

Руденко В.І.

**Перевірила:**

д.т.н. Левченко Л.О.

КИЇВ-2022

**Мета роботи**

Ознайомитися зі структурою виконуваних програм \*.ЕХЕ

та \*.СОМ та їх образом в пам’яті.

**Теоретична частина**

Операційна система MS DOS передбачає два типи виконуваних програм, які мають розширення \*.СОМ та \*.ЕХЕ .

Операційна система MS DOS передбачає два типи виконуваних програм, які мають розширення \*.СОМ та \*.ЕХЕ . Перед завантаженням в оперативну пам’ять СОМ- та ЕХЕ-програм DOS визначає спеціальну область оперативної пам’яті розміром 256 (100h) байт - сегментну адресу, яка називається префіксом програмного сегменту (PSP – Program Segment Prefics). PSP може використовуватися в програмі для визначення імен файлів та параметрів з командного рядка, які вводяться при запуску програми на виконання, обсягу допустимої пам'яті, змінних оточення системи та ін.

При завантаженні програми в оперативну пам’ять DOS ініціалізує як мінімум три сегментних регістри: CS, DS, SS (додатковим може бути ES). Код та дані переміщуються з файлу на диску в оперативну пам’ять, а адреси цих сегментів заносяться у CS та DS відповідно. Сегмент стека або виділяється в області, що вказана в програмі, або співпадає (якщо він явно не описаний в програмі) з самим першим сегментом програми. Адреса сегменту стека розташовується в регістрі SS. Програма може мати декілька кодових сегментів та сегментів даних і в процесі виконання за допомогою спеціальних команд здійснюється переключення між ними.

Для того, щоб адресувати одночасно два сегменти даних, наприклад, для виконання операції пересилання з однієї області пам’яті в іншу, можна використовувати регістр додаткового сегмента ES. Кодовий сегмент та сегмент стеку завжди визначається вмістом своїх регістрів (CS та SS), в кожний момент виконання програми завжди використовується якийсь один кодовий сегмент та один сегмент стека.

Усі сегменти можуть використовувати різні області пам’яті, а можуть частково або повністю перекриватися. Кодовий сегмент повинен обов’язково описуватися в програмі, усі інші сегменти можуть бути відсутніми. У цьому випадку при завантаженні програми в оперативну пам’ять DOS ініціює регістри DS та ES значенням адреси префікса програмного сегмента PSP. Регістр SS при цьому ініціалізується значенням сегменту, що знаходиться одразу за PSP, тобто першого сегменту програми. При цьому слід враховувати, що стек «зростає вниз» (при розміщенні у стек вміст регістра SP, що вказує на вершину стека, зменшується, а при зчитуванні зі стека - збільшується). Тому при розміщенні у стек будь-яких значень вони можуть затерти PSP, в зв’язку з цим слід завжди сегмент стеку описувати в програмі явно та задавати для нього розмір, достатній для нормальної роботи. У вказівник команд ІР завантажується зсув точки входу в програму (вибирається з операнда директиви END), у вказівник стека SP – зсув кінця сегмента стека.

Основні відмінності цих програм полягають у наступному:

програми типу \*.СОМ (двійковий образ коду та даних програми)

• складаються тільки з одного сегменту, в якому розміщується і код програми, і дані, і стек;

• файл СОМ-формату не містить переміщуваних адрес;

• СОМ-файл завантажується, починається з адреси PSP:0100h;

• CS, DS, ES, SS вказують на PSP;

• SP вказує на кінець сегмента PSP (зазвичай 0FFFEh);

• ІР містить 100h (перший байт модуля) в результаті команди JMP PSP:100h;

• розмір програми типу \*.СОМ не може перевищувати 64 Кбайт;

• використовується модель пам’яті TINY;

• образ СОМ-файла зчитується з диску та розташовується в пам’яті, починаючи PSP:0100h;

• програма  генерує стек автоматично, тому в самій асемблерній програмі стек має бути відсутнім; усі дані повинні бути визначені у сегменті коду; в програмах типу \*.ЕХЕ міститься спеціальний заголовок, за допомогою якого завантажувач виконує налаштування посилань на сегменти у завантаженому модулі

• використовуються окремі сегменти і для коду, і для даних, і для стеку;

• розмір програми типу \*.ЕХЕ не має жорстких обмежень, тому що програми може мати будь-яку кількість сегментів команд та даних;

• ЕХЕ-файл завантажується, починається з адреси PSP:0100h;

• DS та ES вказують на початок PSP;

• CS, IP, SS, SP ініціалізуються значеннями, зазначеними в заголовку ЕХЕ;

• використовується модель пам’яті SMALL, яка передбачає розташування структурних частин програми у двох сегментах: сегменту кода програми (64 Кбайт) та сегменту даних і стека , що належать одній групі Dgroup (64 Кбайт).

Для створення ЕХЕ-програми необхідно виконати наступні команди:  
-TASM /LA /ZI ім’я файлу.asm  
-TLINK /X /V ім’я файлу.obj  
-ім’я файлу.exe

При створенні програми \*.СОМ необхідно виконання двох умов:  
• вхідний текст програми повинен мати відповідний формат звикористанням мінімальної моделі пам’яті;  
• необхідно після компоновки отримати виконуваний файл зрозширенням \*.СОМ.

При використанні пакета ТАSM при виклику компоновщика необхідно вказати ключ /Т:

TASM /Z /N ім’я файлу.asm  
Ключ /z дозволяє виведення на екран рядків вихідного тексту програми, в яких асемблер виявив помилки (без цього ключа пошук помилок довелося б проводити з лістингу трансляції)./n пригнічує виведення в лістинг таблицю ідентифікаторів (переліку символічних позначень в програмі), за рахунок чого трохи зменшується інформативність лістингу, але скорочується його розмір.

TLINK /X /T ім’я файлу.obj  
Ключ /х означає не створити файл карти map.файл.  
Ключ /t означає створити виконуваний com.файл.

ім’я файлу.com

**Завдання**

1.Ознайомитися зі структурою програми \*.ЕХЕ та образом такої програми у пам’яті.

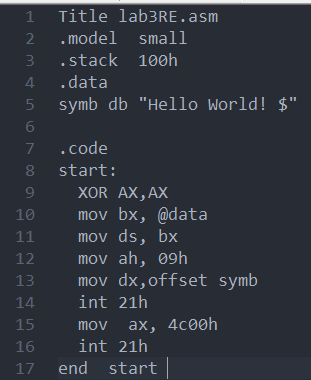
2. Ознайомитися зі структурою програми \*.СОМ та образом такої програми у пам’яті.

3. Засвоїти особливості створення виконуваних програм \*.ЕХЕ та \*.СОМ.  
 4. Підготувати .ЕХЕ та .СОМ програми для виведення на екран повідомлення HELLO WORD! з варіантом опису даних: msg db "Hello Word!", 0Dh, 0Ah, '$' 0Dh – символ повернення каретки (Carriage Return), 0Ah - символ переведення рядка

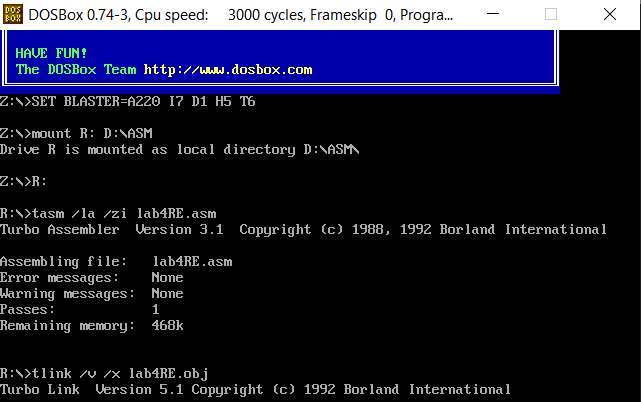
5. Продемонструвати роботу .ЕХЕ – та .СОМ-програм, пояcнити розподіл пам’яті для цих програм.

6. У DEBUG за допомогою команди D CS:0000 ви маєте можливість переглянути машинний код програми для ехе-програми.  
 7. У DEBUG за допомогою команди D DS:100 ви маєте можливість переглянути дані.

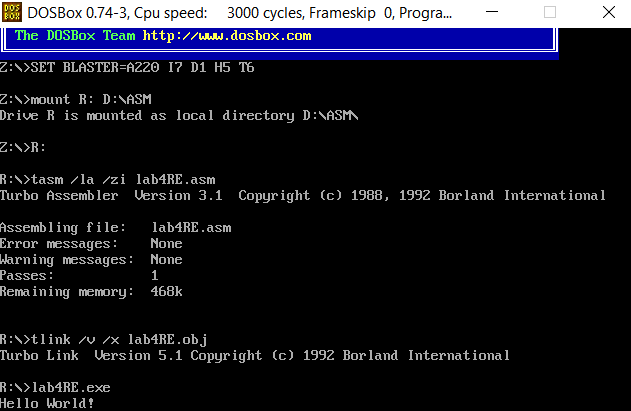
**Результати роботи**



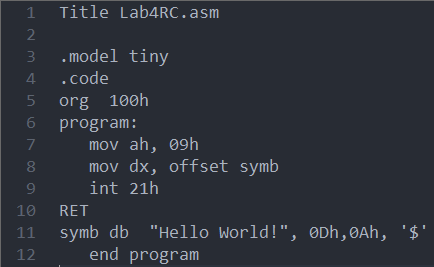
Результат 1 (Код програми для .exe)



Результат 2 (Підготовка та створення програми .exe)



Результат 3 (демонстрація коректної роботи програми .exe)



Результат 4 (код програми .com)



Результат 5 (створення та демонстрація програми .com)

**Висновок**

В результаті виконання лабораторної роботи №4 було ознайомлено зі структурою виконуваних програм .exe та .com. та їх образами в пам’яті. Ознайомлено з відмінностями структур та їхніми особливостями при проектуванні.

Створено два програмнийх продукти, які за допомогою різних структур програм виводять фразу “Hello World”.