



МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені
ІГОРЯ СІКОРСЬКОГО»

Кафедра обчислювальної техніки

МЕТОДИЧНІ ВКАЗІВКИ

до виконання лабораторних робіт
з дисципліни
"Системне програмування"

Розробник: доцент, канд. техн. наук, доцент Павлов В.Г.
(посада, вчена ступінь та звання П.І.Б.)

Затверджено на засіданні кафедри
Протокол № 10 від 25 травня 2022 р.

Завідувач кафедри ОТ
Стіренко С.Г.
(прізвище, ініціали)

(підпис)

Київ – 2022

ЗМІСТ

ВСТУП.....	2
1 МЕТОДИЧНІ ВКАЗІВКИ ЩОДО ВИКОНАННЯ ЛАБОРАТОРНИХ РОБІТ.....	3
1.1 Мета циклу лабораторних робіт.....	3
1.2 Зміст та оформлення лабораторних робіт.....	3
2 ЛАБОРАТОРНІ РОБОТИ.....	4
2.1 Лабораторна робота 1	4
2.2 Лабораторна робота 2	7
2.3 Лабораторна робота 3	9
2.4 Лабораторна робота 4	11
2.5 Лабораторна робота 5	13
2.5 Лабораторна робота 6	15
2.5 Лабораторна робота 7	17
2.5 Лабораторна робота 8	19

ВСТУП

Дисципліна «Системне програмування» відноситься до нормативних освітніх компонент циклу професійної підготовки фахівців рівня магістр з напрямку 121 «Інженерія програмного забезпечення». Для успішного вивчення курсу «Системне програмування» студенти повинні засвоїти матеріал та мати певні знання, вміння та навички з таких дисциплін, як «Історія науки і техніки», «Комп'ютерна дискретна математика», « Основи комп'ютерних систем і мереж», «Основи програмування», «Об'єктно-орієнтоване програмування». Знання та навички, можуть бути використані при опануванні наступних курсів: «Компоненти програмної інженерії», «Операційні системи», «Безпека програмного забезпечення»,

Цикл лабораторних робіт призначений для проведення комп'ютерного практикуму дисципліни «Системне програмування» студентів денної форми навчання. Роботи дозволяють отримати практичні навички написання програм у процесорних системах з використанням механізмів сучасних мов та бібліотек системного програмування.

Методичні вказівки включають для кожної роботи:

- інструкція з порядку підготовки до роботи та її виконання у формі доповнення шаблонів програмних проектів;
- варіанти індивідуальних завдань для виконання кожної лабораторної роботи з циклу.;
- питання для самоконтролю;
- література, що рекомендується.

1. МЕТОДИЧНІ ВКАЗІВКИ ЩО ДО ВИКОНАННЯ ЛАБОРАТОРНИХ РОБІТ

1.1 Мета циклу лабораторних робіт

Метою проведення лабораторних занять з кредитного модулю "Системне програмування" є практичне закріплення знань, набутих під час лекційного курсу в частині вивчення базових архітектур та програмне оточення процесорів Intel 64 та IA-32, структури форматів шістнадцяткових та двійкових чисел для цілих та дробових значень, систему команд процесору та їхню структуру, основи нізкорівневого програмування на мові Assembler.

1.2 Зміст протоколу та оформлення лабораторних робіт

Протокол виконання кожної лабораторної роботи має містити:

- титульний лист;
- завдання на лабораторну роботу згідно варіанта;
- лістинг програми з коментаріями;
- необхідні розрахунки згідно завданню на лабораторну роботу;
- розрахунки контрольних прикладів з проміжними результатами у необхідній кількості
- результати експериментального виконання роботи;
- висновки по роботі (аналіз одержаних результатів).

2. ЛАБОРАТОРНІ РОБОТИ

Цикл лабораторних робіт включає 8 робіт. Варіанти завдань видаються викладачем на підставі таблиць в описах робіт.

Для надсилання звітів та програмних файлів до лабораторних робіт використовується наступний шаблон для назв:

<номер ЛР>-<номер варіанту>--<номер групи>-<прізвище>

Наприклад: **4-22-ІМ-12-Petrov** (усі літери латинські)

2.1 Лабораторна робота 1

Внутрішнє представлення цілих і дійсних даних у процесорі ix86

Мета роботи:

Вивчення форматів внутрішнього представлення цілих і дійсних чисел шляхом порівняння результатів, отриманих теоретично, з результатами, що містяться в отриманому розширеному лістингу програми.

Порядок виконання роботи

1. Сформувані наступні початкові числа на основі дати народження студента, записаної у форматі **dd.mm.yyyy**:

- **A** – ціле двозначне число, що дорівнює **dd**;
- **B** – ціле чотиризначне число, яке дорівнює **ddmm**, записаним без роздільника;
- **C** – ціле восьмизначне число, яке дорівнює **ddmmyyyy**, записаним без роздільника;
- **D** – число, отримане шляхом ділення числа **A** на число **N**, де **N** - чотиризначний номер залікової книжки;
- **E** – число, отримане шляхом ділення числа **B** на число **N**;
- **F** - число, отримане шляхом ділення числа **C** на число **N**;

(при діленні округлювати результат до 3 знаків після коми)

2. Представити отримані числа в двійковій системі числення (*дробову частину округлювати до 10 знаків після коми*).

3. За допомогою розрядної сітки показати в звіті представлення цілих чисел в наступних форматах:

- “**ddmmuyyy**” у вигляді символічного рядка;
- числа **A** і **-A** у однобайтовому форматі **Byte**;
- числа **A, B, -A** і **-B** у двобайтовому форматі **Word**;
- числа **A, B, C, -A, -B** і **-C** у чотирьохбайтовому форматі **ShortInt**;
- числа **A, B, C, -A, -B** і **-C** у восьмибайтовому форматі **LongInt**;

4. Представити числа **D, E** і **F** у нормалізованому вигляді.

5. За допомогою розрядної сітки показати в звіті представлення дійсних чисел в наступних форматах:

- числа **D** і **-D** в чотирьохбайтовому форматі **Single** (float);
- числа **E** і **-E** у восьмибайтовому форматі **Double** (double);
- числа **F** і **-F** у десятибайтовому форматі **Extended** (long double).;

6. Остаточні результати по кожному пункту надати у звіті окрім двійкової ще і в шістнадцятковій системі числення.

7. Скласти програму на мові **Assembler**, в якій задати вказані символи і усі варіанти констант **A, B, ... F**, які присутні у пунктах 3 та 5. Значення констант **A, B, ... F** вивести у віконній формі на екран.

8. У режимі консолі виконати компіляцію і виконання програми. Привести скриншот вікна виведення значень змінних в звіті.

9. У режимі консолі виконати формування розширеного лістингу програми у вигляді файлу з розширенням “**lst**” за допомогою опції **/FL** компілятора **ML**.

10. Знайти в лістингу фрагмент, що містить коди команд і даних, і привести цей фрагмент в звіті з виконання лаб. роботи.

11. У наведеному фрагменті лістингу знайти і помітити кольоровим олівцем або фломастером шістнадцяткові коди усіх варіантів представлення чисел **A, B, ... F**.

12. Порівняти результати, що містяться в лістингу, з розрахунковими, отриманими при виконанні пунктів 3 – 6, і зробити висновки по лабораторній роботі.

Питання для самоконтролю

1. Як кількість розрядів у представленні числових даних впливають на точність?
2. Чим відрізняються представлення додатних та від'ємних цілих чисел?
3. Чому для обчислення порядку дійсних чисел використовуються додаткові константи?
4. Чим відрізняється представлення мантиси у десятибайтовому форматі **Extended** від інших форматів для дійсних чисел?
5. Чим відрізняються представлення додатних та від'ємних дійсних чисел?

Література, що рекомендується

1. IEEE Standard 754 for Binary Floating-Point Arithmetic

2.2 Лабораторна робота 2

Дослідження структури програм формату COM

Мета роботи

Вивчення прийомів написання, компіляції і відладки програм формату **COM** в середовищі **Masm32**. Здобуття навичок читання лістингу і розшифрування кодів команд.

Порядок виконання роботи

1. Вивчити структуру програм формату **COM** і застосування переривань **BIOS** та **MS-DOS** при роботі з консоллю.

2. Розробити програму на мові Асемблер, за допомогою якої на екран системної консолі по введеному паролю виводяться персональні дані студента – ПІБ, дата народження, номер залікової книжки тощо (див. лаб. роботу 1) кожне з нового рядка з попереднім очищенням вікна системної консолі. Для правильного відображення при виведенні символів кирилиці підключити відповідну кодову сторінку.

3. Вивчити опції компілятора і лінковщика і сформувати **BAT**-файл, в якому передбачити завдання назви вихідного файлу **.asm**, як параметра. Шлях до файлу має бути визначений в результаті сканування логічного диску і служити для вказівки розміщення відповідних йому об'єктного і виконуваного файлів.

4. Виконати компіляцію розробленого файлу у формат **COM**.

5. Перевірити роботу програми шляхом введення як правильного, так і невірного паролів.

6. Отриманий виконуваний файл дослідити за допомогою програми **HEX**-редактору **HIEW32** або **HIEW**. У останньому випадку для запуску програми сформувати ярлик, де виконати налаштування параметрів сумісності з використовуваною операційною системою. Демоверсія **HEX**-редактору **HIEW32**, наприклад, доступна на **Internet**-ресурсі:

<http://soft.mydiv.net/win/download-Hiew.html>.

7. Перемикаючи послідовно режими перегляду (**Text – Hex – Decode**), зняти три відповідних скріншоти програми і привести їх в звіті по лабораторній роботі.

8. Переконалися, що текст оригінала пароля, який міститься в тексті програми, може бути легко виявлений за допомогою **HEX**-редактора.

9. Виконати шифрування пароля за допомогою функції **XOR**, знову скопіювати **COM**-файл і переконалися, що тепер вони не виявляються явним чином в тексті виконуваного **COM**-файлу. Привести скріншоти цієї програми в режимах «**Text**» та «**Decode**» у звіті по лабораторній роботі.

10. Порівняти текст програми, який набирався в редакторі, з текстом програми в скопійованому вигляді, який формує **HEX**-редактор **HIEW**. Виявити розбіжності і відобразити їх в звіті по лабораторній роботі.

11. На отриманому у п. 9 в режимі «**Decode**» скріншоті знайти всі команди **MOV**, вписати їх коди і розібрати по окремих полях, відповідно їх формату. Результати привести в звіті по лабораторній роботі.

12. Зробити висновки по лабораторній роботі.

Надсилаються: асемблерні файли для двох версій програм (без шифрування та зі шифруванням), обидва скопійованих **COM**-файли, **VAT**-файл та файл звіту.

Питання для самоконтролю

1. Які існують обмеження на обсяг **COM**-файлів?
2. Чому потрібно використовувати шифрування для паролів, що зберігаються у **COM**-файлах?
3. Чим відрізняється компіляція **COM** та інших виконуваних файлів?
4. Чому саме команда **MOV** вибрана для аналізу у цієї роботи?
5. Як перемикаються режими перегляду змістів файлів у **HEX**-редакторі **HIEW32**?

Література, що рекомендується

1. **HIEW**. Матеріал з Вікіпедії. <https://uk.wikipedia.org/wiki/Hiew>

2.3 Лабораторна робота 3

Дослідження структури програм формату EXE.

Мета роботи

Ознайомлення із специфікацією **COFF** (Common Object File Format). Вивчення прийомів дослідження структури файлів **PE**-формату.

Порядок виконання роботи

1. Вивчити структуру програм формату **EXE** [1].
2. Розробити програму на мові Асемблер, за допомогою якої у віконному інтерфейсі по введеному паролю виводяться персональні дані студента – ПІБ, дата народження, номер залікової книжки тощо (див. лаб. роботу 1).
3. Виконати компіляцію розробленого файлу у формат **EXE**.
4. Перевірити роботу програми шляхом введення як правильного, так і невірного паролів.
5. Отриманий виконавчий файл дослідити за допомогою програми **HEX**-редактора **HEW32** (див. лаб. роботу 2) або **WinHex** (<http://rainbowsky.ru/system/winhex/> - trial версія*)[2].
6. На скріншоте перших 25 рядків вмісту файлу обвести кольоровим олівцем або фломастером області **MS-DOS** заголовка (**DOS_HEADER**), **PE** заголовка (**PE_HEADER**) і таблиці секцій (**SECTION_HEADERS**). Скріншот привести в звіті по лабораторній роботі.
7. Відповідно до опису секцій [1] скласти таблицю, в яку занести параметри свого файлу, вказані в розділах 3.3.1, 3.4.1 і 4 (перша таблиця).
8. У останньому стовпчику таблиці розшифрувати вписані значення полів заголовка файлу. Таблицю привести в звіті по лабораторній роботі.
9. Провести дослідження того ж файлу за допомогою меню "**PE Editor**" безкоштовної програми **PE Tools** (<http://soft.mydiv.net/win/download-PE-Tools.html>*). Все скріншоти вікон програми з даними, відповідними раніше побудованій таблиці, привести в звіті по лабораторній роботі.

10. Дослідити таблицю імпорту (**Import Directory**) даного файлу і визначити, які саме функції використовуються з бібліотек, що підключаються. Скріншоти вікон **Import Directory** з функціями, що імпортуються, з кожного бібліотечного файлу привести в звіті по лабораторній роботі.

11. Знайти в тексті файлу по зусу, узятому з побудованої таблиці, секцію з даними і переконатися, що текст оригінала пароля, що міститься в тексті програми, може бути легко виявлений за допомогою **HEX**-редактора. Привести скріншот цього фрагмента програми у вигляді **HEX** - коду в звіті по лабораторній роботі.

12. Виконати шифрування пароля за допомогою функції **XOR**, знову скомпілювати **EXE** -файл і переконатися, що тепер вони не виявляються явним чином в тексті виконуваного **EXE** - файлу. Привести скріншоти цієї програми в режимах «Hex» і «Text» в звіті по лабораторній роботі.

13. Зробити висновки по лабораторній роботі.

Питання для самоконтролю

1. Чим викликана поява стандарту **COFF**?
2. Чому **DOS_HEADER** присутній у всіх файлів незалежно від операційного середовища, де вони використовуються?
3. Чи є розмір заголовку **PE_HEADER** сталим?
4. Яка інформація міститься для кожної секції у **SECTION_HEADERS**?
5. Яку функцію виконує точка входу (**ENTRY POINT**)?

Література, що рекомендується

1. Microsoft Portable Executable and Common Object File Format Specification.

(<http://www.osdever.net/documents/PECOFF.pdf>)

2. PE Format . <https://docs.microsoft.com/en-us/windows/win32/debug/pe-format>

3. WinHex. Матеріал з Вікіпедії.

<https://ru.wikipedia.org/wiki/WinHex>.

** вказані ресурси є орієнтовними, студент має право самостійно вибрати який-небудь інший ресурс для завантаження.*

2.4 Лабораторна робота 4 Макровизначення і макроси в MASM32.

Мета роботи

Вивчити технології створення і вживання макросів. Дослідження результатів роботи макрогенератора MASM32.

Порядок виконання роботи

1. Вивчити правила використання макросів в програмах [1].
2. Переробити програму, підготовлену в лабораторній роботі 3 так, щоб виведення кожного виду інформації (дата народження, номер залікової книжки і т. п.) виконувалося у вигляді багатократного звернення до макросу (макрос 1), в якому у віконному інтерфейсі здійснювалося б виведення тексту, який передається в макрос як параметр.
3. Оформити також у вигляді окремих макросів фрагменти програми, де здійснюється шифрування введеного рядка символів (макрос 2), а також порівняння її з хеш-кодом оригіналу пароля, що зберігається у програмі (макрос 3). Макровизначення для всіх макросів розмістити в тому ж файлі, що і програма. У всіх макросах обов'язково використовувати звичайні і приховані коментарі, а в останньому макросі також використовувати механізм оголошення локальних міток.
4. Виконати компіляцію і компоновку файлу програми.
5. Перевірити роботу програми шляхом введення як правильного, так і невірного паролів.
6. Отримати розширений лістинг програми за допомогою опції /FL компілятора ML.
7. Провести дослідження отриманого лістингу: визначити, яким чином компілятор виконав трансляцію кожній з команд макросів і помітити їх в тексті розширеного лістингу кольоровими олівцями або кольоровими фломастерами.
8. Повторити дослідження, які виконані в п. 4 – 7 для другої версії програми, де всі перераховані в п. 2 і 3 макровизначення для макросів повинні розташовуватися в окремому файлі.

9. Порівняти тексти розширених лістингів для обох варіантів розміщення макровизначень, знайти схожість і відмінності. Відобразити їх в звіті по лабораторній роботі.

10. Зробити висновки по лабораторній роботі.

Питання для самоконтролю

1. У яких IDE використовуються макроси?
2. Які переваги надає використання макросів?
3. Яку роль відіграє макровизначення?
4. Чи може одне макровизначення мати декілька макророзширень?
5. Чи може використання макросів впливати на швидкість виконання програми?

Література, що рекомендується

1. Рисований О.М. Системне програмування. – Харків, : НТУ «Харківський політехнічний інститут», 2012, 912 с.

2.5 Лабораторна робота 5

Арифметичні і логічні операції з цілими числами. Масиви.

Мета роботи

Вивчення арифметичних і логічних команд Асемблера і здобуття навиків виконання розрахунків з елементами масивів.

Порядок виконання роботи

1. Вивчити арифметичні і логічні команди цілочисельної арифметики [1, 2].

2. Розробити програму на мові Асемблер, в якій згідно з індивідуальним варіантом завдання (таблиця. 1) виконуються обчислення значення арифметичного виразу з подальшим виводом результату* у віконному інтерфейсі.

3. Для всіх варіантів: в разі парного результату він перед виводом додатково ділиться на 2, а в разі непарного – результат додатково множиться на 5.

4. Розрахунки (п. 2, 3) повторити в програмі для 5 значень змінних**, причому всі вихідні значення задати цілими числами у вигляді одновимірних масивів.

5. Для перевірки правильності виконання розрахунків і результатів, що виводяться, заздалегідь виконати контрольні розрахунки. Проміжні і остаточні результати контрольних розрахунків привести в звіті по лабораторній роботі.

6. Виконати відладку програми шляхом порівняння розрахованих програмою результатів з контрольними прикладами. Лістинг розробленої програми і скріншоти розрахунків по всіх контрольних прикладах привести в звіті по лабораторній роботі.

7. Зробити висновки по лабораторній роботі.

** Вхідні дані та результат у вікні виводу мають бути представлені у десятковій системі счислення.*

*** Змінні a, b, c і d повинні забезпечувати цілочисельне ділення, але не бути рівними 1 або 0.*

Табл. 1. Варіанти індивідуальних завдань

№ вар	Формула для розрахунку	№ вар	Формула для розрахунку
1	$(c - d/2 + 33)/(2*a*a-1);$	11	$(2*c + d/4 + 23)/(a*a - 1);$
2	$(-25/a + c - b*a)/(1 + c*b/2);$	12	$(41 - d/4 - 1)/(c/b + a*d);$
3	$(c/d + 3*a/2)/(c - a + 1);$	13	$(c/4 + 28*d)/(a/d - c - 1);$
4	$(8*b + 1 - c)/(a/2 + b*c);$	14	$(-15*a + b - a/4)/(b*a - 1);$
5	$(a*b/4 - 1)/(41-b*a + c);$	15	$(-2*c - d + 53)/(a/4 - 1);$
6	$(-2*c + d*82)/(a/4 - 1);$	16	$(b + c*b - a/4)/(a*b - 1);$
7	$(-53/a + d - 4*a)/(1 + a*b);$	17	$(c*d + 23)/(a/2 - 4*d - 1);$
8	$(4*b/c - 1)/(12*c + a - b);$	18	$(2*b - 38*c)/(b+ a/c + 1);$
9	$(21 - a*c/4)/(1 + c/a + b);$	19	$(12/c - d*4 + 73)/(a*a + 1);$
10	$(c/b - 24 + a)/(2*a*c - 1);$	20	$(c - 33 + b/4)/(a*c/b - 1);$

Питання для самоконтролю

1. Чому при операціях множення та ділення відіграє роль формат початкових даних?
2. Як визначити парність двійкового числа?
3. Яким чином перевіряється працездатність програми?
4. Чи може бути числове значення виведено безпосередньо у текстове вікно повідомлень MessageBox?
5. На що впливає формат елементів масиву?

Література, що рекомендується

1. Рисований О.М. Системне програмування. Підручник. В 2-х т. – Харків, : “Слово”, 2015.– 576 с.
2. Галісеєв. Г Системне програмування. – К., : Видавництво Університет "Україна", 2019. - 113 с.
3. Rudolf Marek. Učíme se programovat v jazyce Assembler pro PC. Computer Press, 2003. – 228 p.

2.6 Лабораторна робота 6

Обчислення арифметичних виразів і трансцендентних функцій.

Використання команд співпроцесора ix87.

Мета роботи

Вивчення команд Асемблера для арифметики з плаваючою комою і здобуття навичок виконання розрахунків з елементами масивів.

Порядок виконання роботи

1. Вивчити арифметичні команди з плаваючою комою [1].
2. Розробити програму на мові Асемблер, в якій згідно з індивідуальним варіантом завдання (табл. 1) виконуються обчислення значення арифметичного вираження із застосуванням команд співпроцесора **ix87** з подальшим виведенням результату* у віконному інтерфейсі.
3. Для всіх варіантів передбачити завдання значень вхідних змінних у форматі **double (DQ)**, проміжних результатів обчислень – у форматі **long double (DT)**, а кінцевих - знову у форматі **double**.
4. Розрахунки (п. 2, 3) повторити в програмі для 5 значень змінних**, причому всі вхідні значення задати дійсними числами у вигляді одновимірних масивів.
5. Для перевірки правильності виконання розрахунків і результатів, що виводяться, заздалегідь виконати контрольні розрахунки. Проміжні і остаточні результати контрольних розрахунків з точністю не гірше точності розрахунків у комп'ютері привести в звіті.
6. Виконати відладку програми шляхом порівняння розрахованих програмою результатів з контрольними прикладами. Лістинг розробленої програми і скріншоти розрахунків по всіх контрольних прикладах привести в звіті по лабораторній роботі.
7. Зробити висновки по лабораторній роботі.

* *Вхідні дані та результат у вікні виводу мають бути представлені у десятковій системі счислення.*

** *Значення змінних a , b , c та d повинні мати бути дійсними числами, не рівними 0 або 1, допустимими для обчислення трансцендентних функцій, а тригонометричні функції обчислюються від аргументів, заданих в радіанах.*

Табл. 1. Варіанти індивідуальних завдань

№ вар.	Формула для розрахунку	№ вар.	Формула для розрахунку	№ вар.	Формула для розрахунку
1	$\frac{2 \bullet c - d + \sqrt{23 \bullet b}}{a/4 - 1}$	8	$\frac{tg(c) - d \bullet 23}{2 \bullet b - a}$	15	$\frac{a/2 - 4 \bullet d - 1}{c \bullet tg(b + 23)}$
2	$\frac{-2 \bullet c - d \bullet 82}{tg(a/4 - b)}$	9	$\frac{2 \bullet c - d/23}{\ln(b - a/4)}$	16	$\frac{c/d + \ln(3 \bullet a/2)}{c - b + 1}$
3	$\frac{c + 4 \bullet b - \sqrt{123 \bullet d}}{1 - a/2}$	10	$\frac{4 \bullet c + d - 1}{b - tg(a/2)}$	17	$\frac{2 \bullet c + \lg(b) \bullet 51}{d - a - 1}$
4	$\frac{\lg(2 \bullet c - a) + b - 152}{a/4 + c}$	11	$\frac{2 \bullet c - d \bullet \sqrt{42/b}}{c + a - 1}$	18	$\frac{2 \bullet c + \ln(d/4) + 23}{a \bullet a - b}$
5	$\frac{tg(a + c/4) - 12 \bullet d}{a \bullet b - 1}$	12	$\frac{\sqrt{25/c} - d + 2}{b + a - 1}$	19	$\frac{42 \bullet c - d/2 + 1}{a \bullet a - \ln(b - 5)}$
6	$\frac{-2 \bullet c - \sin(a/d) + 53}{a/4 - b}$	13	$\frac{2 \bullet a - c + b/3}{arctg(b - d/2)}$	20	$\frac{arctg(2 \bullet c)/d + 2}{b - a - 1}$
7	$\frac{2 \bullet c - \lg(d/4)}{a \bullet 2 - b}$	14	$\frac{4 \bullet \lg(c) - b/2 + 23}{d - a + 1}$	21	$\frac{\sqrt{53/a + d} - 4 \bullet b}{1 + b \bullet c}$

Питання для самоконтролю

1. У чому потреба операцій з плаваючою точкою?
2. Яким чином побудований стек для виконання операцій з плаваючою точкою?
3. Які вбудовані константи використовуються у арифметиці з плаваючою точкою?

Література, що рекомендується

1. Кравець В. О., Рисований О.М. Системне програмування. Програмування на мові асемблера: навч. посіб. – Харків, : НТУ «ХПІ», 2007. – 448 с.

2.7 Лабораторна робота 7

Модульне програмування. Використання процедур.

Мета роботи

Вивчення прийомів модульного програмування, методів звернення до процедур і передачі в них параметрів.

Порядок виконання роботи

1. Вивчити методи звернення до процедур і передачі в них параметрів [1].

2. Для свого варіанту індивідуального завдання до лабораторної роботи б розробити програму на мові Асемблер, в якій використовувати три процедури з різними способами передачі параметрів:

- через реєстри;
- через стек;
- за допомогою директив EXTRN та PUBLIC.

3. Для цього чисельник дробу зі свого варіанту індивідуального завдання до лабораторної роботи б розділити на два доданка, з яких для першого застосувати передачу параметрів і результату через реєстри, а для другого – через стек. Для знаменника використовувати метод оголошення загальних змінних директивами **public** і **extern**. Виведення результату* виконати в основній програмі.

4. Розрахунки (п. 3) повторити в програмі для 5 значень змінних**, причому всі вхідні значення задати дійсними числами у вигляді одновимірних масивів.

5. Для перевірки правильності виконання розрахунків і результатів, що виводяться, заздалегідь виконати контрольні розрахунки, які повинні охоплювати різноманітні сполучення вхідних даних, на які програма повинна надавати вірну відповідь***. Проміжні і остаточні результати контрольних розрахунків привести в звіті по лабораторній роботі. Точність розрахунків така ж, як і у лаб. роботі б.

6. Виконати відладку програми шляхом порівняння розрахованих програмою результатів з контрольними прикладами. Лістинг

розробленої програми і скріншоти розрахунків по всіх контрольних прикладах привести в звіті по лабораторній роботі.

7. У протоколі по лабораторній роботі для першого і другого способів передачі параметрів поруч з відповідними командами у лістингу відобразити в графічному вигляді стани стека при зверненні до процедур, виконання у них команд та повернення з процедур до основної програми.

8. Зробити висновки по лабораторній роботі.

** Вхідні дані та результат у вікні виводу мають бути представлені у десятковій системі счислення.*

*** Значення змінних a , b , c та d повинні мати бути дійсними числами, не рівними 0 або 1, допустимими для обчислення трансцендентних функцій, а тригонометричні функції обчислюються від аргументів, заданих в радіанах.*

**** Якщо задані значення вхідних змінних не належать до області визначень певних функцій зі завдання, то програма повинна сповіщати про це.*

Питання для самоконтролю

1. Які переваги надає модульне програмування?
2. У чому використання процедур відрізняється від застосування макросів?
3. Де може бути розташований опис процедур у програмі?
4. У чому сутність прологу та епілогу процедури?
5. Навіщо потрібно вирівнювати стек?

Література, що рекомендується

1. Kip R. Irvine. Assembly Language for x86 Processors. – Florida International University: Pearson, Seventh Edition 2014, - 876 p.

2.8 Лабораторна робота 8

Розробка і використання динамічних бібліотек.

Мета роботи

Вивчення прийомів розробки і використання процедур, представлених у вигляді динамічних бібліотек.

Порядок виконання роботи

1. Вивчити методи розробки і використання **DLL**- бібліотек [1].

2. Для свого варіанту індивідуального завдання до лабораторної роботи 6 розробити процедуру на мові Асемблер, в якій реалізувати обчислення заданого арифметичного виразу* і виведення результатів у віконному інтерфейсі. Параметрами, які передаються до процедури, є значення змінних. Процедуру розробити в двох варіантах:

- окремо процедура без вказівки точки входу;
- окремо процедура з використанням точки входу.

3. Розробити файл установок модуля **DEF** і виконати компіляцію обох варіантів файлу бібліотек **DLL**. Лістинги обох варіантів написання процедур, файлу **DEF** і командних рядків компіляції представити в звіті по лабораторній роботі.

4. Написати і скомпілювати файл основної програми, в якому передбачити неявне завантаження розробленої **DLL**- бібліотеки з передачею в неї в циклі **5** значень змінних**, заданих дійсними числами у вигляді одновимірних масивів.

5. Перевірити роботу основної програми з підключенням по черзі спочатку одної, а потім іншої **DLL**- бібліотек і привести в звіті по лабораторній роботі лістинг основної програми і скріншоти вікон з результатами обчислень для обох варіантів.

6. Написати і скомпілювати файл основної програми, в якому передбачити явне завантаження розробленої **DLL**- бібліотеки з передачею до неї в циклі тих же **5** значень змінних.

7. Переконатися в працездатності основної програми при зверненні в ній по черзі спочатку одної, а потім іншої з **DLL**- біблі-

отек, розроблених в п. 3. Лістинг основної програми та скріпоти вікон з результатами обчислень за обома варіантами привести в звіті по лабораторній роботі.

8. Для перевірки правильності виконання розрахунків і результатів, що виводяться, заздалегідь виконати контрольні розрахунки. Проміжні і остаточні результати контрольних розрахунків привести в звіті по лабораторній роботі***.

9. Виконати відладку програми шляхом порівняння розрахованих програмою результатів з контрольними прикладами.

10. При підключенні у програмі зовнішніх бібліотек та макросів додати до нього коментар, де вказати, які функції з них потрібні для використання у даній програмі.

11. Зробити висновки по лабораторній роботі.

** Результат обчислення повинен представлятися в десятковій системі числення з точністю, достатньою для зв'язки з контрольними розрахунками.*

*** Значення змінних a , b , c та d повинні мати бути дійсними числами, не рівними 0 або 1, допустимими для обчислення трансцендентних функцій, а тригонометричні функції обчислюються від аргументів, заданих в радіанах.*

**** Контрольні розрахунки можуть бути узяті з лабораторної роботи 6.*

Питання для самоконтролю

1. Навіщо потрібні динамічні бібліотеки?
2. Які засоби визначення точки входу існують?
3. У чому відмінність явного та неявного завантаження бібліотек?
4. Як спосіб завантаження впливає на ефективність використання пам'яті?
5. Яку роль відіграє файл опису процедур при їх завантаженні?

Література, що рекомендується

1. Рисований О.М. Системне програмування. – Харків, : НТУ «Харківський політехнічний інститут», 2012, 912 с.