



СИСТЕМНЕ ПРОГРАМУВАННЯ. Частина 1. Мова асемблера

Робоча програма навчальної дисципліни (Силабус)

Реквізити навчальної дисципліни

Рівень вищої освіти	<i>Перший (бакалаврський)</i>
Галузь знань	<i>12 Інформаційні технології</i>
Спеціальність	<i>123 Комп'ютерна інженерія</i>
Освітня програма	<i>Системне програмування та спеціалізовані комп'ютерні системи</i>
Статус дисципліни	<i>Нормативна</i>
Форма навчання	<i>очна(денна)</i>
Рік підготовки, семестр	<i>2 курс, осінній семестр</i>
Обсяг дисципліни	<i>3 кредити</i>
Семестровий контроль/ контрольні заходи	<i>Залік/МКР</i>
Розклад занять	<i>http://rozklad.kpi.ua/</i>
Мова викладання	<i>Українська</i>
Інформація про керівника курсу / викладачів	Лектор: к.т.н, с.н.с. Тесленко Олександр Кирилович, akirill766@ukr.net teslenko@scs.kpi.ua Лабораторні: к.т.н, с.н.с. Тесленко Олександр Кирилович, Ст. викладач Дробязко Ірина Павлівна
Розміщення курсу	На кафедральній системі дистанційного навчання https://scs-kpi.pp.ua/

Програма навчальної дисципліни

1. Опис навчальної дисципліни, її мета, предмет вивчення та результати навчання

Предмет дисципліни. Вивчення кредитного модуля "Системне програмування Частина 1. " дозволяє сформуванню у студентів компетенції, необхідні для розв'язання практичних задач професійної діяльності, пов'язаної з аналізом, використанням та розробкою програмного забезпечення мовою Асемблера для сучасних комп'ютерних пристроїв різного призначення.

Метою навчальної дисципліни є формування у студентів компетенцій:

ЗК 1 Здатність до абстрактного мислення, аналізу та синтезу

ЗК 3 Здатність застосовувати знання у практичних ситуаціях

ЗК 7 Вміння виявляти, ставити та вирішувати проблеми

ФК 2 Здатність використовувати сучасні методи і мови програмування для розроблення алгоритмічного та програмного забезпечення

ФК 3 Здатність створювати системне та прикладне програмне забезпечення комп'ютерних систем та мереж

ФК 5 Здатність використовувати засоби і системи автоматизації проєктування до розроблення компонентів комп'ютерних систем та мереж, Інтернет-додатків, кіберфізичних систем тощо.

ФК 6 Здатність проєктувати, впроваджувати та обслуговувати комп'ютерні систем та мережі різного виду та призначення

ФК 11 Здатність оформляти отримані робочі результати у вигляді презентацій, науко-во-технічних звітів

Згідно з вимогами освітньо-професійної програми студенти після засвоєння навчальної дисципліни мають продемонструвати такі **програмні результати навчання**:

ПРН 6 Вміти застосовувати знання для ідентифікації, формулювання і розв'язування технічних задач спеціальності, використовуючи методи, що є найбільш придатними для досягнення поставлених цілей.

ПРН 8 Вміти системно мислити та застосовувати творчі здібності до формування нових ідей

ПРН 10 Вміти розробляти програмне забезпечення для вбудованих і розподілених застосунків, мобільних і гібридних систем, розраховувати, експлуатувати типове для спеціальності обладнання.

ПРН 11 Вміти здійснювати пошук інформації в різних джерелах для розв'язання задач комп'ютерної інженерії.

2. Пререквізити та постреквізити дисципліни (місце в структурно-логічній схемі навчання за відповідною освітньою програмою)

Кредитний модуль "Системне програмування. Частина 1." базується на кредитних модулях - 3О 9.1 Програмування. Частина 1, 3О 9.2 Програмування. Частина 2.

Кредитний модуль "Системне програмування Частина 1" забезпечує вивчення кредитних модулів ПО 7.1 Об'єктно-орієнтоване програмування. Частина 1. та ПО 7.2 Об'єктно-орієнтоване програмування. Частина 2. ОСВІТНЬО ПРОФЕСІЙНОЇ ПРОГРАМИ першого (бакалаврського рівня вищої освіти) .

3. Зміст навчальної дисципліни

Тема 1. Предмет дисципліни .

Тема 2. Основи Асемблера.

Тема 3. Архітектура ЕОМ на рівні програмно доступних елементів.

Тема 4. Склад та адресація програмних об'єктів.

Тема 5. Основи модульного програмування

4. Навчальні матеріали та ресурси

Базова література

1. Дробязко І.П.,Тесленко О.К. Програмування мовою Асемблера. Лабораторний практикум з дисципліни «Системне програмування» [Електронне видання] К. : ФПМ, НТУУ «КПІ», 2013
Протокол №6 засідання вченої ради ФПМ від 12.01.2013

2. О.К. Тесленко, І.П. Дробязко СИСТЕМНЕ ПРОГРАМУВАННЯ КУРСОВА РОБОТА
Рекомендовано Методичною радою КПІ ім. Ігоря Сікорського як навчальний посібник для здобувачів ступеня бакалавра за освітньою програмою «Системне програмування і спеціалізовані комп'ютерні системи» спеціальності 123 «Комп'ютерна інженерія» «КПІ ім. Ігоря Сікорського» 2021Гриф надано Методичною радою КПІ ім. Ігоря Сікорського (протокол № 5 від 14.01.2021 р.) за поданням Вченої ради факультету прикладної математики (протокол № 5 від 14.12.2020 р.)
<https://ela.kpi.ua/handle/123456789/39044> 2021. – 162 с

3. Мельник А.О. Архітектура комп'ютера. Наукове видання. – Луцьк: Волинська обласна друкарня, 2008. – 470

4. Тарасенко В.П. Корнійчук В.И. Основи комп'ютерної арифметики. ВПП "Корнійчук", Київ, 2002, - 176 с

Додаткова література

1. Intel Architecture Developer's Manual. Volume 2. Instruction Set Reference. Order Number 243191.

2. Intel Architecture Developer's Manual. Volume 3. System Programming Guide. Order Number 243192.
3. Intel® 64 and IA-32 Architectures Software Developer's Manual. Volume 1: Basic Architecture

Навчальний контент

5. Методика опанування навчальної дисципліни (освітнього компонента)

Навчальна дисципліна охоплює 36 годин лекцій та 36 годин лабораторних занять, а також виконання модульної контрольної роботи

Лекційні заняття

1 Тема 1. Предмет дисципліни

Лекція 1.1. Рівні деталізації процесів та структур в обчислювальних системах. Роль і місце системного програмування, призначення і особливості системного програмування. Визначення мови Асемблера та її роль в системному програмуванні. Основні риси та особливості мови Асемблера в порівнянні з мовами високого рівня.

Тема 2. Основи Асемблера.

Лекція 2.1. Алфавіт та терми (лексеми) мови Асемблера. Структура програм на мовою Асемблера. Структура речень мови Асемблера. Машинні інструкції та директиви. Вимоги до оформлення програм. Технологія програмування мовою Асемблера. Характеристика компіляторів та редакторів зв'язку. Вхідні та вихідні файли. Налаштовувачі та їх роль в технології програмування мовою Асемблера.

Завдання на CPC: Вивчення основних параметрів запуску трансляторів MASM та TASM, редакторів зв'язків LINK та TLINK, основних можливостей та команд налаштовувачів AFD та TD.

Тема 3. Архітектура ЕОМ на рівні програмно доступних елементів.

Лекція 3.1. Формальне визначення алгоритму. Машина Тьюрінга. Реалізація алгоритму програмного управління. Потік даних та потік команд. Управління потоком команд, команди передачі управління. Гарвардська архітектура та архітектура фон-Неймана, їх переваги та недоліки.

Лекція 3.2. Характеристика програмно доступних елементів ЕОМ. Поняття тригера (розряд, біт, ознака, прапорець), поняття регістра, регістри з додатковими ланцюгами читання-записування. Оперативні запам'ятовуючі пристрої, поняття фізичної адреси, адресний простір ЕОМ. Коротка історія розвитку процесорів 80x86 ф. Інтел. Загальна характеристика реального та захищеного режиму, 64-розрядні платформи.

Лекція 3.3 Системна архітектура процесорів 80x86 в реальному режимі. Типова структура команд процесорів 80x86 та її зв'язок з структурою машинних інструкцій мови Асемблера. Сегментна організація пам'яті. Сегментні регістри. Логічна адреса та формування фізичної адреси. Суть переваг сегментної організації пам'яті.

Лекція 3.4 Загальна класифікація команд процесорів. Регістр ознак. Огляд системи команд процесорів 80x86 по функціональних групах. Підгрупа команд внутрішньо сегментної безумовної передачі управління та передачі управління по умові. Поняття прямої адресації (Асемблер) та відносної адресації (процесор). Правила формування мнемонік команд передачі управління по умові в залежності від типів даних та комбінацій ознак.

Завдання на CPC: Переглянути всі команди процесорів 80x86. Проаналізувати особливості використання в командах кожного із регістрів даних та адрес. Ознайомитись зі складом та

функціями додаткових реєстрів в процесорах 80x86 – реєстрів налагодження та системних реєстрів.

Тема 4. Склад та адресація програмних об'єктів

Лекція 4.1. Операнди в командах процесора та їх відповідність операндам машинних інструкцій мови Асемблера. Поняття типу даних в мові Асемблера та методи їх визначення, директива `=`. Поняття адресного виразу.

Лекція 4.2. Організація адресації даних в структурі команд процесорів 80x86 з використанням 16-розрядних реєстрів адрес та 16-розрядних зміщень в команді (байт `modr\m`) та їх відображення мовою Асемблера. Поняття виконавчої (ефективної) адреси, багатокомпонентна адреса, адресна арифметика. Використання сегментних реєстрів для завдання сегментних складових логічних адрес.

Лекція 4.3. Проблеми сумісності програм при використанні 32 розрядних реєстрів та зміщень в команді. Префікси зміни розрядності даних та адрес. Особливості байту `modr\m`. Байт `Sib`. Використання багатокомпонентної адреси для адресації елементів складних структур даних. Особливості використання сегментного реєстра `SS` по замовчуванню.

Лекція 4.4. Організація адресації команд в процесорах 80x86. Внутрішньосегментні та між сегментні команди безумовної передачі управління. Суть не прямої (опосередкованої) адресації в командах безумовної передачі управління. Роль не прямої адресації в системних програмах.

Лекція 4.5. Автоінкрементні та автодекрементні режими адресації даних. Група команд для обробки рядків (масивів, ланцюжків). Префікси повторення (апаратні цикли). Аналіз послідовності виконання процесів в однопроцесорних ЕОМ. Проблема локальних даних. Організація та використання стека, використання автоіндексації. Команди запису в стек та читання зі стеку. Команди виклику процедур та повернення з процедур.

Завдання на СРС: Створення в мові Асемблера багатовимірних масивів, структур та масивів структур. Використання режимів адресації в машинних інструкціях для посилання на елементи складних структур даних. Використання команди `Lea`. Використання команд обробки масивів для типових системних функцій – пересилання, конкатенація, порівняння масивів, пошук під масиву в масиві, заповнення масиву константою.

Тема 5. Основи модульного програмування.

Лекція 5.1. Суть, причини та рівні застосування модульного програмування. Поняття програмного модуля та програмного інтерфейсу. Роль системних засобів для забезпечення зв'язку між модулями на рівні початкових, об'єктних та виконавчих модулів. Модульне програмування як предтеча об'єктно орієнтованому програмуванню.

Лекція 5.2. Особливості модульного програмування на рівні початкових програм. Директива `Include`. Суть та організація макросів в мові Асемблера. Поняття препроцесора та вбудованого макрогенератора. Директива `Equ`. Поняття макровизначення, макрокоманди та макророзширення.

Лекція 5.3. Директиви `Macro` та `Endm`. Формальні параметри в макровизначенні. Директива `Local`. Макрокоманди та варіанти задання фактичних параметрів. Учебні приклади макросів. Поняття трансляції по умові. Огляд директив трансляції по умові. Ефективність використання директив трансляції по умові в макросах. Стандартні макроси мови Асемблера. Особливості організації..

Лекція 5.4. Рекомендації по організації процедур в мові Асемблера. Директиви `Proc` та `Endp`. Практично не обмежені можливості передачі параметрів процедур в мові Асемблера. Приклади організації процедур з передачею параметрів в потоці команд та в реєстрі ознак.

Лекція 5.5. Стандарти передачі параметрів через стек, типові концепції в системному програмуванні. Організація локальних даних процедур.

Забезпечення модульного програмування на рівні об'єктних файлів. Параметри директиви Segment, проблеми та організація об'єднання логічних сегментів в один фізичний сегмент редакторами зв'язків. Директиви Public та Extern.

Лекція 5.6. Організація взаємозв'язку між модулями на мові Асемблера та модулями на мовах високого рівня. Недоліки статичного зв'язування програмних модулів. Суть динамічного зв'язування. Організація та використання DLL.

Завдання на СРС: Проаналізувати процеси об'єднання логічних сегментів у фізичні та створення зв'язків між модулями, проаналізувати алгоритми корекції зміщень в командах, проаналізувати особливості організації процедур компіляторами мов високого рівня.

ЛАБОРАТОРНІ ЗАНЯТТЯ

Основні завдання циклу лабораторних занять робіт є набуття студентами практичних навичок створювати програмні модулі мовою Асемблера та асемблерні вставки в мовах високого рівня

1. Вивчення технології програмування на мові Асемблера. Засоби трансляції, збирання, завантаження та налагоджування програм на мові Асемблера. Створення першої програми на мові Асемблера.
2. Аналіз результатів трансляції операторів мови С на мові Асемблера. Вивчення реалізації основних програмних конструкцій на мові Асемблера. Використання асемблерних вставок в програмах на мові C++.
3. Вивчення типових структур даних та методів адресації елементів цих структур
4. Розробка програм на мові Асемблера для обробки складних структур даних. Вивчення правил організації взаємозв'язку між програмами на мові Асемблера та мовах високого рівня.

6. Самостійна робота студента

1. Тема 2. Основи Асемблера.

Вивчення основних параметрів запуску трансляторів MASM та TASM, редакторів зв'язків LINK та TLINK, основних можливостей та команд налагоджувачів AFD та TD.

2. Тема 3. Архітектура EOM на рівні програмно доступних елементів.

Переглянути всі команди процесорів 80x86. Проаналізувати особливості використання в командах кожного із регістрів даних та адрес. Ознайомитись зі складом та функціями додаткових регістрів в процесорах 80x86 – регістрів налагодження та системних регістрів.

3. Тема 4. Склад та адресація програмних об'єктів

Створення в мові Асемблера багатовимірних масивів, структур та масивів структур. Використання режимів адресації в машинних інструкціях для посилання на елементи складних структур даних. Використання команди Lea. Використання команд обробки масивів для типових системних функцій – пересилання, конкатенація, порівняння масивів, пошук під масиву в масиві, заповнення масиву константою.

4. Тема 5. Основи модульного програмування.

Проаналізувати процеси об'єднання логічних сегментів у фізичні та створення зв'язків між модулями, проаналізувати алгоритми корекції зміщень в командах, проаналізувати особливості організації процедур компіляторами мов високого рівня.

6. Політика навчальної дисципліни

- **Академічна доброчесність.** Політика та принципи академічної доброчесності визначені у розділі 3 Кодексу честі Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського». Детальніше: <https://kpi.ua/code>.
- **Норми етичної поведінки.** Норми етичної поведінки студентів і працівників визначені у розділі 2 Кодексу честі Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського». Детальніше: <https://kpi.ua/code>.
- **Навчання іноземною мовою.** У ході виконання завдань студентам може бути рекомендовано звернутися до англomовних джерел.
- **правила відвідування занять** – обов'язкове (як лекцій, так і лабораторних);
- **правила захисту лабораторних робіт;**
Приймання викладачем лабораторної роботи включає:
 - а) безпомилкову роботу програми під час перевірки викладачем або самостійне виправлення помилки на протязі поточного заняття;
 - б) самостійність студента при створенні програми, яка визначається знанням семантики будь якої директиви чи будь якої машинної інструкції пред'явленої програми та усвідомлення ролі в наданій програмі будь якої директиви чи будь якої машинної інструкції;
 - с) спроможність на протязі поточного заняття незначно модифікувати програму за вимогою викладача;

7. Види контролю та рейтингова система оцінювання результатів навчання (PCO)

Семестровий контроль проводиться у вигляді заліку. Для оцінювання результатів навчання застосовується 100-бальна рейтингова система та університетська шкала.

Поточний контроль: фронтальні опитування (тематичні контрольні роботи по попередніх лекціях), МКР.

Календарний контроль: проводиться двічі на семестр як моніторинг поточного стану виконання вимог силябусу.

Семестровий контроль: залік

ПОЛОЖЕННЯ

про рейтингову систему оцінки успішності студентів

з кредитного модуля І.2.10 “Системне програмування. Частина 1. ”
для спеціальності: 123 Комп'ютерна інженерія
факультету прикладної математики.

Рейтинг студента з кредитного модуля складається з:

- 1) балів за виконання та захист лабораторних робіт,
- 2) балів за експрес опитування по матеріалах попередньої лекції,
- 3) балів за модульний контроль

І. Система рейтингових (вагових) балів та критерії оцінювання

1. Лабораторні роботи

Ваговий бал r_1 – 16. Максимальна кількість балів за чотири лабораторні роботи дорівнює $16 \text{ балів} \times 4 = 64 \text{ бали}$.

Критерії оцінювання.

Приймання викладачем лабораторної роботи включає:

- a) безпомилкову роботу програми під час перевірки викладачем або самостійне виправлення помилки на протязі поточного заняття;*
- b) самостійність студента при створенні програми, яка визначається знанням семантики будь якої директиви чи будь якої машинної інструкції пред'явленої програми та усвідомлення ролі в наданій програмі будь якої директиви чи будь якої машинної інструкції; спроможність на протязі поточного заняття незначно модифікувати програму за вимогою викладача;*

Максимальний ваговий бал (16) студент одержує при умові вчасної здачі лабораторної роботи :

- лабораторна робота №1 – до 30.09.20 включно*
- лабораторна робота №2 – до 31.10.20 включно*
- лабораторна робота №3 – до 30.11.20 включно*
- лабораторна робота №4 – до 23.12.20 включно*

При здачі лабораторної роботи з запізненням на тиждень – 14 балів

При здачі лабораторної роботи з запізненням на 2 тижні – 12 балів

При здачі лабораторної роботи з запізненням на місяць – 10 балів

При здачі лабораторної роботи з запізненням на більше чим місяць – 8 балів

2. Експрес опитування по матеріалах попередньої лекції

Максимальний ваговий бал (12) студент одержує при умові вчасної (не пізніше наступної лекції) відповіді на задане йому запитання.

Критерії оцінювання.

- a) Відповідь правильна та прокоментована $r_2 = 12$*
- b) Відповідь правильна, але не прокоментована $r_2 = 10$ балів*
- c) Відповідь не правильна, але коментар вказує на механічну помилку (наприклад, помилку в термінології) $r_2 = 8$ балів*
- d) Відповідь не правильна, але коментар вказує на не значну логічну помилку – 6 балів*
- e) 0 балів в інших випадках.*

3. Модульний контроль

Ваговий бал одного запитання – $r_3 = 12$. Максимальна кількість балів за два запитання дорівнює $12 \text{ балів} \times 2 = 24 \text{ бали}$.

Критерії оцінювання.

- f) Відповідь правильна та прокоментована $r_2 = 12$*
- g) Відповідь правильна, але не прокоментована $r_2 = 10$ балів*
- h) Відповідь не правильна, але коментар вказує на механічну помилку (наприклад, помилку в обчисленнях) $r_2 = 8$ балів*
- i) Відповідь не правильна, але коментар вказує на не значну логічну помилку – 6 балів*
- j) 0 балів в інших випадках.*

5. Заохочувальні бали :

- За здані в строк підряд дві роботи - +1 бал;*
- За здачу всіх лабораторних робіт до початку залікової сесії, не залежно від строків здачі на протязі семестру - +4 бали;*
- Третя та четверта лабораторні роботи мають завдання підвищеної складності. Виконання цих робіт в строк по завданню підвищеної складності +2 бали за кожну роботу*

Розрахунок шкали (R) рейтингу:

Сума вагових балів контрольних заходів протягом семестру складає:

$R_c = r_1 + r_2 + r_3 = 64 + 12 + 24 = 100$ балів.

Сума балів для окремо взятого студента

$$rc = \min((r1c+r2c + rs),100)$$

де $r1c$ – сума балів за лабораторні роботи, $r2c$ – кількість балів за контрольну роботу, rs – сума заохочувальних балів.

Семестровий контроль: *залік*

Умови допуску до семестрового контролю: *зарахування усіх лабораторних робіт / семестровий рейтинг більше 60 балів.*

Таблиця відповідності рейтингових балів оцінкам за університетською шкалою:

Кількість балів	Оцінка
100-95	Відмінно
94-85	Дуже добре
84-75	Добре
74-65	Задовільно
64-60	Достатньо
Менше 60	Незадовільно
Не виконані умови допуску	Не допущено

8. Додаткова інформація з дисципліни (освітнього компонента)

- *перелік питань, які виносяться на семестровий модульний контроль в додатку 1 до силабусу;*
- *Під час навчання та для взаємодії зі студентами використовуються сучасні інформаційно-комунікаційні та мережеві технології ля вирішення навчальних завдань.*

Робочу програму навчальної дисципліни (силабус):

Складено доцентом кафедри СПіСКС., к.т.н., с.н.с Тесленком О.К.

Ухвалено кафедрою СПСКС (Протокол №11 від 12.06.2024).

Погоджено Методичною комісією ФПМ (Протокол №12 від 21.06.2024).

ДОДАТОК 1
до робочої програми
кредитного модуля (силабусу) ПО 3.1

Завдання на контрольну роботу

Гр. П.І.Б.

Завдання 1

- 1 Які дані знаходяться в регістрах CS та IP (EIP) під час виконання поточної команди
- 2 Реалізувати на мові Асемблера оператори присвоєння мови C++:

Якщо задано $\text{int } x, y, *px$.

Гр. П.І.Б.

Завдання 2

- 1 Які сегментні регістри і в яких випадках використовуються по замовчуванню
- 2 Реалізувати на мові Асемблера оператор присвоєння мови C++:

Якщо задано $\text{int } Y, *PX$.

Гр. П.І.Б.

Завдання 3

- 1 Яким чином програмісти при програмуванні мовою Асемблера відрізняють знакові та беззнакові цілі числа

2 Реалізувати на мові Асемблера оператор мови C++:
if (v1>=v5) *v2=v4+3; else *v2=v3-7;
де v1,v2,v3,v4,v5 мають тип integer.

Гр. П.І.Б.
Завдання 4

1 Які типові дії над рядками (масивами) можна виконати за допомогою префіксів повторення для команд обробки масивів (рядків)
2 Реалізувати на мові Асемблера оператор мови C++:
if (v1<v2) v2=v3-3; else v2=7+v4;
де v1,v2,v3, v4 мають тип unsigned.

Гр. П.І.Б.
Завдання 5

1 Чим відрізняється архітектура фон-Неймана від Гарвардської архітектури
2 Реалізувати на мові Асемблера оператор мови C++:
for (v1=v2,v1< v3, v1++) v4=v4+v1;
де v1,v2,v3,v4 мають тип integer.

Гр. П.І.Б.
Завдання 6

1 Як процесор виконує внутрішню сегментну команду jmp.
2 Реалізувати на мові Асемблера оператор мови C++:
for (v1=v3,v1> v4,v1--) v2=v2+v1;
де v1,v2,v3,v4 мають тип integer.

Гр. П.І.Б.
Завдання 7

1 Надати загальну структуру програм мовою Асемблера
2 Реалізувати на мові Асемблера оператор мови C++:
while (v1 < v2) v1+=v2;
де v1,v2 мають тип integer.

Гр. П.І.Б.
Завдання 8

1 Яку роль грають префікси заміни сегмента в командах процесорів x8086
2 Реалізувати на мові Асемблера оператор мови C++:
while (v1 < v3) v1+=v2;
де v1,v2,v3 мають тип unsigned.

Гр. П.І.Б.
Завдання 9

1 З яких частин складається логічна адреса та як процесор формує фізичну (лінійну) адресу
2 Реалізувати на мові Асемблера оператор мови C++:
do v1+=v2 while (v1>v3);

де v_1, v_2, v_3 мають тип `unsigned`.

Гр. П.І.Б.

Завдання 10

- 1 Надати структуру та класифікацію рядків програм мовою Асемблера
- 2 Реалізувати на мові Асемблера оператор мови C++:
`do v1=v1-v2 while (v1<v2);`
де v_1, v_2 мають тип `unsigned`.

Гр. П.І.Б.

Завдання 11

- 1 Визначити ідентифікатори в мові Асемблера та надати їх класифікацію.
- 2 Реалізувати на мові Асемблера оператор мови C++:
`for (v1=v2; v1<v3; v1++) v4=v4+5;`
де v_1, v_2, v_3, v_4 мають тип `integer`.

Гр. П.І.Б.

Завдання 12

- 1 Охарактеризувати константи в мові Асемблера, надати приклади можливих констант.
- 2 Реалізувати на мові Асемблера оператор мови C++:
`for (v1=v2; v1<v3; v1--) v4=*v4+10;`
де v_1, v_2, v_3, v_4 мають тип `integer`.

Гр. П.І.Б.

Завдання 13

- 1 Надати класифікацію операндів в машинних інструкціях мови Асемблера
- 2 Реалізувати на мові Асемблера оператор мови C++:
`if (v1>v3) v2=v1+*v2; else v2=v1-*v2;`
де v_1, v_2, v_3 мають тип `unsigned`.

Гр. П.І.Б.

Завдання 14

- 1 Які групи та підгрупи команд реалізовані в процесорах 80 x 86
- 2 Реалізувати на мові Асемблера оператор мови C++:
`do v1=v1-v2+1; while (v1>v2);`
де v_1, v_2 мають тип `integer`.

Гр. П.І.Б.

Завдання 15

- 1 Як процесор виконує команду `jmp dword ptr Addr1` в реальному режимі.
- 2 Реалізувати на мові Асемблера оператор мови C++:
`do v1= v2 -v1+1; while (v1<=v3);`
де v_1, v_2, v_3 мають тип `integer`.

Гр. П.І.Б.

Завдання 16

1 З яких частин складаються в загальному випадку команди процесорів 80x86

2 Реалізувати на мові Асемблера оператор мови C++:

while (v1 <= v2) v1=v1+v2;

де v1,v2 мають тип unsigned.

Гр. П.І.Б.

Завдання 17

1. Як процесор виконує команди push eax та pop eax

2. Реалізувати на мові Асемблера оператор мови C++:

if (*v1>=-4) v2=*v1-v2; else v2=*v1+v2;

де v1,v2 мають тип integer

Гр. П.І.Б.

Завдання 18

1 Охарактеризувати поняття типу в мові Асемблера

2 Реалізувати на мові Асемблера оператори присвоєння мови C++:

v1=&v2;

*v1+=5;

де v1,v2,*v1 мають тип int.

Гр. П.І.Б.

Завдання 19

1 Як в сегменті стека організовується область пам'яті для збереження тимчасових даних

2 Реалізувати на мові Асемблера оператор мови C++:

if (v1<=-4) *v2=v1-3; else *v2=-7;

де v1,v2 мають тип integer.

Гр. П.І.Б.

Завдання 20

1 Навести перелік типових ознак в регістрі flags

2 Реалізувати на мові Асемблера оператор мови C++:

if (v1!=-*v2) *v2=v3; else *v2=*v4;

де v1,v2,v3,v4 мають тип integer.

Гр. П.І.Б.

Завдання 21

1 Який з операндів у машинних інструкціях є джерелом даних, а який приймачем, навести приклади.

2 Реалізувати на мові Асемблера оператор мови C++:

for (v1=v2+3; v1<v3;v1++) v4=v2-v1;

де v1,v2,v3,v4 мають тип integer.

Гр. П.І.Б.

Завдання 22

1 Визначити поняття знакового розширення, навести приклади
2 Реалізувати на мові Асемблера оператор мови C++:
for (v1=v3; v1> v4;v1--) v2=v2-v1;
де v1,v2,v3,v4 мають тип integer.

Гр. П.І.Б.
Завдання 23

1 Як процесор виконує команду movsb
2 Реалізувати на мові Асемблера оператор мови C++:
while (v1 > -80) v1=v1-*v2;
де v1,v2 мають тип integer.

Гр. П.І.Б.
Завдання 24

1 Охарактеризувати директиви = та EQU мови Асемблера
2 Реалізувати на мові Асемблера оператор мови C++:
while (v1 <= v3) v1=v1-v2;
де v1,v2 v3 – тип unsigned.

Гр. П.І.Б.
Завдання 25

1 Охарактеризувати структуру макровизначень та визначити особливості задання формальних параметрів
2 Реалізувати на мові Асемблера оператор мови C++:
do v1=v1-v2; while (v1>v3);
де v1,v2 ,v3 – мають тип unsigned.

Гр. П.І.Б.
Завдання 26

1 Що визначають оператори offset, typ та seg в операндах пам'яті
2 Реалізувати на мові Асемблера оператор мови C++:
do v1=v1+v2; while (v1<=v3);
де v1,v2, v3 - мають тип int.

Гр. П.І.Б.
Завдання 27

1 Дати загальну характеристику організації процедур в мові Асемблера
2 Реалізувати на мові Асемблера оператор мови C++:
for (v1=v2; v1<v3+2;v1++) v4=v4+5;
де v1,v2,v3,v4 мають тип int.

Гр. П.І.Б.
Завдання 28

1 Охарактеризувати фактичні параметри в макрокомандах
2 Реалізувати на мові Асемблера оператор мови C++:

for (v1=v2+1;v1> v3;v1--) v4=v4-12;
де v1,v2,v3,v4 мають тип integer.

Гр. П.І.Б.
Завдання 29

1. Охарактеризувати директиви компіляції по умові
2. Реалізувати на мові Асемблера оператор мови C++:
if (v1>=-4) v2=v1-v2; else v2=v1+v2;
де v1,v2 мають тип integer.

Гр. П.І.Б.
Завдання 30

- 1 Охарактеризувати процесс макророзширення та заміни формальних параметрів на фактичні
- 2 Реалізувати на мові Асемблера оператор мови C++:
do v1=v1+v2-1; while (v1<=12);
де v1, v2 мають тип integer.