

Основи програмування

Заняття №1

1. Будова ПК.

Основними вузлами системного блоку є:

- **електричні плати**, що керують роботою комп'ютера (мікропроцесор, оперативна пам'ять, контролери пристроїв тощо);
- **накопичувач на жорсткому диску (вінчестер)**, призначений для читання або запису інформації;
- **накопичувачі (дисководи)** для гнучких магнітних дисків (дискет).

Основною платою ПК є **материнська плата** (MotherBoard). На ній розташовані:

- **процесор** - основна мікросхема, що виконує математичні та логічні операції;
- **чіпсет (мікропроцесорний комплект)** - набір мікросхем, що керують роботою внутрішніх пристроїв ПК і визначають основні функціональні можливості материнської плати;
- **шини** - набір провідників, по яких відбувається обмін сигналами між внутрішніми пристроями комп'ютера;
- **оперативний запам'ятовуючий пристрій (ОЗП)** - набір мікросхем, що призначені для тимчасового зберігання даних, поки включений комп'ютер;
- **постійний запам'ятовуючий пристрій (ПЗП)** - мікросхема, призначена для довготривалого зберігання даних, навіть при вимкненому комп'ютері;
- роз'єми для під'єднання додаткових пристроїв (**слоти**).

Процесор

Процесор - головна мікросхема комп'ютера, його "мозок". Він дозволяє виконувати програмний код, що знаходиться у пам'яті і керує роботою всіх пристроїв комп'ютера. Швидкість його роботи визначає швидкодію комп'ютера. Конструктивно, процесор - це кристал кремнію дуже маленьких розмірів. Процесор має спеціальні комірки, які називаються регістрами.

Саме в цих регістрах містяться команди, які виконуються процесором, а також дані, якими оперують ці команди. Робота процесора полягає у вибиранні з пам'яті у певній послідовності команд та даних і виконанні їх. На цьому і базується виконання

програм. У ПК обов'язково має бути присутній центральний процесор (Central Rprocessing Unit - CPU), який виконує всі основні операції. Часто ПК оснащений додатковими сопроцесорами, орієнтованими на ефективне виконання специфічних функцій, такими як, математичний сопроцесор для обробки числових даних у форматі з плаваючою точкою, графічний сопроцесор для обробки графічних зображень, сопроцесор введення/виведення для виконання операції взаємодії з периферійними пристроями.

Основними параметрами процесорів є:

- тактова частота,
- розрядність,
- робоча напруга,
- коефіцієнт внутрішнього домноження тактової частоти,
- розмір кеш пам'яті.

Тактова частота визначає кількість елементарних операцій (тактів), що виконуються процесором за одиницю часу. Тактова частота сучасних процесорів вимірюється у МГц (1 Гц відповідає виконанню однієї операції за одну секунду, 1 МГц=106 Гц). Чим більша тактова частота, тим більше команд може виконати процесор, і тим більша його продуктивність. Перші процесори, що використовувалися в ПК працювали на частоті 4,77 МГц, а сьогодні робочі частоти найсучасніших процесорів досягли позначки в 2 ГГц (1 ГГц = 103 МГц).

Розрядність процесора показує, скільки біт даних він може прийняти і обробити в свої регістрах за один такт. Розрядність процесора визначається розрядністю командної шини, тобто кількістю провідників у шині, по якій передаються команди. Сучасні процесори сімейства Intel є 32-розрядними.

Робоча напруга процесора забезпечується материнською платою, тому різним маркам процесорів відповідають різні материнські плати. Зараз робоча напруга процесорів не перевищує 3 В. Пониження робочої напруги дозволяє зменшити розміри процесорів, а також зменшити тепловиділення в процесорі, що дозволяє збільшити його продуктивність без загрози перегріву.

Коефіцієнт внутрішнього домноження тактової частоти - це коефіцієнт, на який слід помножити тактову частоту материнської плати, для досягнення частоти процесора. Тактові сигнали процесор отримує з материнської плати, яка з чисто фізичних причин не може працювати на таких високих частотах, як процесор. На сьогодні тактова

частота материнських плат складає 100-133 МГц. Для отримання більш високих частот у процесорі відбувається внутрішнє домноження на коефіцієнт 4, 4.5, 5 і більше

Кеш-пам'ять. Обмін даними всередині процесора відбувається набагато швидше ніж обмін даними між процесором і оперативною пам'яттю. Тому, для того щоб зменшити кількість звертань до оперативної пам'яті, всередині процесора створюють так звану надоперативну або кеш-пам'ять. Коли процесору потрібні дані, він спочатку звертається до кеш-пам'яті, і тільки якщо там потрібні дані відсутні, відбувається звертання до оперативної пам'яті. Чим більший розмір кеш-пам'яті, тим більша ймовірність, що необхідні дані знаходяться там. Тому високопродуктивні процесори оснащуються підвищеними обсягами кеш-пам'яті. Розрізняють кеш-пам'ять *першого рівня* (виконується на одному кристалі з процесором і має об'єм порядку декілька десятків Кбайт), *другого рівня* (виконується на окремому кристалі, але в межах процесора, з об'ємом в сто і більше Кбайт) та *третього рівня* (виконується на окремих швидкодійних мікросхемах із розташуванням на материнській платі і має обсяг один і більше Мбайт).

У процесі роботи процесор обробляє дані, що знаходяться в його регістрах, оперативній пам'яті та зовнішніх портах процесора. Частина даних інтерпретується як *власне дані*, частина даних - як *адресні дані*, а частина - як *команди*. Сукупність різноманітних команд, які може виконати процесор над даними, утворює так звану **систему команд процесора**. Чим більший набір команд процесора, тим складніша його архітектура, тим довший запис команд у байтах і тим довша середня тривалість виконання команд.

Так, процесори Intel, які використовуються в IBM-сумісних ПК, нараховують більше тисячі команд і відносяться до так званих процесорів із розширеною системою команд - CISC-процесорів (CISC - Complex Instruction Set Computing). На противагу CISC-процесорам розроблено процесори архітектури RISC із скороченою системою команд (RISC - Reduced Instruction Set Computing). При такій архітектурі кількість команд набагато менша, і кожна команда виконується швидше. Таким чином, програми, що складаються з простих команд виконуються набагато швидше на RISC-процесорах.

Зворотна сторона скороченої системи команд полягає в тому, що складні операції доводиться *емалювати* (відтворення програмними або апаратними засобами або їх комбінацією роботи інших програм або пристроїв) далеко не завжди ефективною послідовністю простіших команд. Тому CISC-процесори використовуються в універсальних комп'ютерних системах, а RISC-процесори - у спеціалізованих. Для ПК платформи IBM PC домінуючими є CISC-процесори фірми Intel, хоча останнім часом компанія AMD виготовляє процесори сімейства AMD-K6, які мають гібридну

архітектуру (внутрішнє ядро цих процесорів виконане по RISC-архітектурі, а зовнішня структура - по архітектурі CISC).

В комп'ютерах IBM PC використовують процесори, розроблені фірмою Intel, або сумісні з ними процесори інших фірм, що відносяться до так званого сімейства x86. Родоначальником цього сімейства був 16-розрядний процесор Intel 8086. В подальшому випускалися процесори Intel 80286, Intel 80386, Intel 80486 із модифікаціями, різні моделі Intel Pentium, Pentium MMX, Pentium Pro, Pentium II, Celeron, Pentium III. Найновішою моделлю фірми Intel є процесор Pentium IV. Серед інших фірм-виробників процесорів слід відзначити AMD із моделями AMD-K6, Athlon, Duron та Cyrix.

Шини

З іншими пристроями, і в першу чергу з *оперативною пам'яттю, процесор* зв'язаний групами провідників, які називаються **шинами**. Основних шин три:

- шина даних,
- адресна шина,
- командна шина.

Адресна шина. Дані, що передаються по цій шині трактуються як адреси комірок оперативної пам'яті. Саме з цієї шини процесор зчитує адреси команд, які необхідно виконати, а також дані, із якими оперують команди. У сучасних процесорах адресна шина 32-розрядна, тобто вона складається з 32 паралельних провідників.

Шина даних. По цій шині відбувається копіювання даних з оперативної пам'яті в регістри процесора і навпаки. У ПК на базі процесорів Intel Pentium шина даних 64-розрядна. Це означає, що за один такт на обробку поступає відразу 8 байт даних.

Командна шина. По цій шині з оперативної пам'яті поступають команди, які виконуються процесором. Команди представлені у вигляді байтів. Прості команди вкладаються в один байт, але є й такі команди, для яких потрібно два, три і більше байтів. Більшість сучасних процесорів мають 32-розрядну командну шину, хоча існують 64-розрядні процесори з командною шиною.

Шини на материнській платі використовуються не тільки для зв'язку з процесором. Усі інші внутрішні пристрої материнської плати, а також пристрої, що підключаються до неї, взаємодіють між собою за допомогою шин. Від архітектури цих елементів багато в чому залежить продуктивність ПК у цілому.

Внутрішня пам'ять

Під внутрішньою пам'яттю розуміють всі види запам'ятовуючих пристроїв, що розташовані на материнській платі. До них відносяться:

- оперативна пам'ять,
- постійна пам'ять,
- енергонезалежна пам'ять.

Оперативна пам'ять RAM (Random Access Memory).

Пам'ять RAM - це масив кристалічних комірок, що здатні зберігати дані. Вона використовується для оперативного обміну інформацією (командами та даними) між процесором, зовнішньою пам'яттю та периферійними системами. З неї процесор бере програми та дані для обробки, до неї записуються отримані результати. Назва "**оперативна**" походить від того, що вона працює дуже швидко і процесору не потрібно чекати при зчитуванні даних з пам'яті або запису. Однак, дані зберігаються лише тимчасово при включеному комп'ютері, інакше вони зникають.

За фізичним принципом дії розрізняють динамічну пам'ять **DRAM** і статичну пам'ять **SRAM**. Комірки динамічної пам'яті можна представити у вигляді мікроконденсаторів, здатних накопичувати електричний заряд. **Недоліки** пам'яті DRAM: повільніше відбувається запис і читання даних, потребує постійної підзарядки. **Переваги:** простота реалізації і низька вартість. Комірки статичної пам'яті можна представити як електронні мікроелементи - тригери, що складаються з транзисторів. У тригері зберігається не заряд, а стан (включений/виключений). Переваги пам'яті SRAM: значно більша швидкодія. Недоліки: технологічно складніший процес виготовлення, і відповідно, більша вартість. **Мікросхеми динамічної пам'яті використовуються як основна оперативна пам'ять, а мікросхеми статичної - для кеш-пам'яті.**

Кожна комірка пам'яті має свою адресу, яка виражається числом. В сучасних ПК на базі процесорів Intel Pentium використовується 32-розрядна адресація. Це означає, що всього незалежних адрес є 232, тобто можливий адресний простір складає 4,3 Гбайт. Однак, це ще не означає, що саме стільки оперативної пам'яті має бути в системі. Граничний розмір обсягу пам'яті визначається чіпсетом материнської плати і зазвичай складає декілька сот мегабайт.

Оперативна пам'ять у комп'ютері розміщена на стандартних панельках, що зветься **модулями**. Модулі оперативної пам'яті вставляють у відповідні роз'єми на материнській платі. Конструктивно модулі пам'яті мають два виконання - однорядні (SIMM (Single In-line **Memory** Module)- модулі) та дворядні (DIMM (Dual In-line **Memory** Module)- модулі). На комп'ютерах з процесорами Pentium однорядні модулі можна застосовувати лише парами (кількість роз'ємів для їх встановлення на

материнській платі завжди парне). DIMM - модулі можна встановлювати по одному. **Комбінувати на одній платі різні модулі не можна.** Основними характеристиками модулів оперативної пам'яті є: об'єм пам'яті та час доступу. SIMM- модулі є об'ємом 4, 8, 16, 32 мегабайти; DIMM - модулі - 16, 32, 64, 128, 256 Мбайт. Час доступу показує, скільки часу необхідно для звертання до комірок пам'яті, чим менше, тим краще. Вимірюється у наносекундах. SIMM - модулі - 50-70 нс, DIMM - модулі - 7-10 нс.

Постійна пам'ять ROM (Read Only Memory)

В момент включення комп'ютера в його оперативній пам'яті відсутні будь-які дані, оскільки оперативна пам'ять не може зберігати дані при вимкненому комп'ютері. Але процесору необхідні команди, в тому числі і відразу після включення. Тому процесор звертається за спеціальною стартовою адресою, яка йому завжди відома, за своєю першою командою. Ця адреса вказує на пам'ять, яку прийнято називати постійною пам'яттю ROM або постійним запам'ятовуючим пристроєм (ПЗП). Мікросхема ПЗП здатна тривалий час зберігати інформацію, навіть при вимкненому комп'ютері. Кажуть, що програми, які знаходяться в ПЗП, "зашиті" у ній - вони записуються туди на етапі виготовлення мікросхеми. Комплект програм, що знаходиться в ПЗП утворює базову систему введення/виведення BIOS (Basic Input Output System). Основне призначення цих програм полягає в тому, щоб перевірити склад та працездатність системи та забезпечити взаємодію з клавіатурою, монітором, жорсткими та гнучкими дисками.

Енергонезалежна пам'ять CMOS

Робота таких стандартних пристроїв, як клавіатура, може обслуговуватися програмами BIOS, але такими засобами неможливо забезпечити роботу з усіма можливими пристроями (у зв'язку з їх величезною різноманітністю та наявністю великої кількості різних параметрів). Але для своєї роботи програми BIOS вимагають всю інформацію про поточну конфігурацію системи. З очевидних причин цю інформацію не можна зберігати ні в оперативній пам'яті, ні в постійній.

Спеціально для цих цілей на материнській платі є мікросхема енергонезалежної пам'яті, яка по технології виготовлення називається CMOS. Від оперативної пам'яті вона відрізняється тим, що її вміст не зникає при вимкненні комп'ютера, а від постійної пам'яті вона відрізняється тим, що дані можна заносити туди і змінювати самостійно, у відповідності з тим, яке обладнання входить до складу системи. Мікросхема пам'яті CMOS постійно живиться від невеликої батарейки, що розташована на материнській платі. У цій пам'яті зберігаються дані про гнучкі та жорсткі диски, процесори і т.д. Той факт, що комп'ютер чітко відслідковує дату і час, також пов'язаний з тим, що ця інформація постійно зберігається (і оновлюється) у пам'яті CMOS. Таким чином, програми BIOS зчитують дані про склад комп'ютерної системи з мікросхеми CMOS, після чого вони можуть здійснювати звертання до жорсткого диска та інших пристроїв.

Висновок:

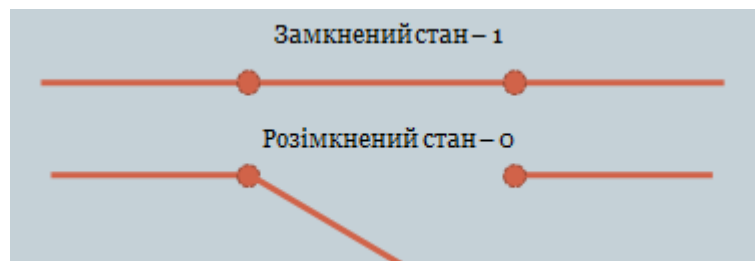
ROM - місце де зберігається інформація - всі програми і операційна система, в якій вони виконуються, зберігаються саме тут, ваші файли так само зберігаються тут; SD і CF карти по суті той же диск. Одним словом, ROM це SD карта, вбудована в ваш пристрій, яку ви просто не можете вийняти.

RAM - місце де дані виконуються. Коли ви включається ваш пристрій, воно завантажує програми з ROM в RAM і запускає їх уже з ОЗУ. Коли ви слухаєте музичний трек, пристрій завантажує частину пісні з ROM в RAM і програв цю частину, потім завантажує іншу частину і т. д. Коли ви читаєте електронну пошту, пристрій завантажує текст з ROM в RAM і тільки потім відображає його, коли ви відкриваєте інше електронної пошти повідомлення, пристрій знову завантажує його з ROM в RAM.

Отже, кількість пам'яті **RAM** визначає скільки програм, музики, фотографій і всього іншого ви можете переглядати (читати, слухати) одночасно; а кількість пам'яті **ROM** визначає скільки з усього перерахованого вище ви можете зберігати на пристрої.

2. Попадання даних у пам'ять комп'ютера

У комп'ютерних системах загальноприйнятим є кодування інформаційних повідомлень за допомогою лише двох символів, які називають двійковими цифрами й умовно позначають як "0" та "1", оскільки сучасні комп'ютери здатні обробляти сигнали, які можуть мати лише 2 стани.



Сприйняти й обробити дані та виконати програму комп'ютер зможе лише за умови, що їх подано у «зрозумілій» для нього формі. Сучасні комп'ютери — це пристрої, здатні обробляти сигнали, які мають два стани (їх позначено цифрами 0 та 1). Отже, щоб комп'ютер міг сприйняти й обробити числові значення, текст, зображення, звук чи відео, їх потрібно подати у вигляді послідовностей 0 та 1. Розглянемо, як це робиться.

У комп'ютерах та інших цифрових пристроях числові дані подаються у двійковій системі. Це позиційна система з основою 2, в якій для запису чисел використовують лише два знаки (цифри 0 та 1).

Наприклад, число 1001 у двійковій системі утворюється так:

$$1001_2 = 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 9_{10}$$

Як бачите, числу 9 десяткової системи відповідає двійкове число 1001. Ось ще кілька прикладів двійкових чисел:

$$11_2 = 1 \times 2^1 + 1 \times 2^0 = 3_{10}$$

Приклади перетворення чисел із десяткової системи числення у двійкову:

| | | | | | | | | | |
|----|-----|----|---|---|---|---|---|---|--|
| 59 | 2 | | | | | | | | |
| 58 | 29 | 2 | | | | | | | |
| 1 | 28 | 14 | 2 | | | | | | |
| | 1 | 14 | 7 | 2 | | | | | |
| | | 0 | 6 | 3 | 2 | | | | |
| | | | 1 | 2 | 1 | | | | |
| | | | | 1 | | | | | |
| 59 | ... | 1 | 1 | 1 | 0 | 1 | 1 | | |
| 10 | | | | | | | | 2 | |

Перетворення десяткового числа у двійкове

| | | | | | | | | | |
|-----|-----|-----|-----|----|----|----|---|---|---|
| 999 | 2 | | | | | | | | |
| 1 | 499 | 2 | | | | | | | |
| | 1 | 249 | 2 | | | | | | |
| | | 1 | 124 | 2 | | | | | |
| | | | 0 | 62 | 2 | | | | |
| | | | | 0 | 31 | 2 | | | |
| | | | | | 1 | 15 | 2 | | |
| | | | | | | 1 | 7 | 2 | |
| | | | | | | | 1 | 3 | 2 |
| | | | | | | | | 1 | 1 |

Перетворення десяткового числа у двійкове

Для подання текстових даних у комп'ютерах використовують так звані **набори символів**. Це таблиці для кодування певної кількості символів, де кожному з них відповідає двійковий код певної довжини. Найпоширенішими сьогодні є набори символів ASCII, Unicode (вимовляється як «юнікод») і сумісні з ними.

У цьому наборі символів для кодування застосовується 8 двійкових розрядів. Неважко здогадатися, що Фрагмент розширеного набору символів ASCII зі значеннями кодів.

| Символ | Двійковий код | Десяткове значення коду |
|--------|---------------|-------------------------|
| 0 | 00110000 | 48 |
| 1 | 00110001 | 49 |
| 2 | 00110010 | 50 |
| ... | ... | |
| A | 01000001 | 89 |
| B | 01000010 | 90 |
| C | 01000011 | 91 |

Як бачите, символ «0» кодується двійковим значенням 00110000 (48_{10}), символ «1» — значенням 00110001 (49_{10}) і чим більша довжина коду, тим більше символів можна закодувати з його використанням. За допомогою двійкового коду довжиною n біт можна закодувати 2^n різних повідомлень. Таким чином, набір символів **ASCII** може включати щонайбільше $2^8 = 256$ символів. Цього достатньо, щоб закодувати цифри, знаки пунктуації, латинські літери (великі й малі) та літери кирилиці (великі й малі).

У стандарті кодування **Unicode** для подання символів використовується 16 розрядів, тобто загальна кількість символів становить $2^{16} = 65\,536$. Цей набір включає в себе всі наявні алфавіти світу, багато математичних, музичних, хімічних символів. Для наукових цілей додано алфавіти історичних писемностей, зокрема давньогрецький та єгипетські ієрогліфи.

3. Біт, Байт

Послідовність двійкових цифр називають **двійковим кодом**. Оскільки двійковий код необхідно певним чином вимірювати тому придумали найменшу частинку двійкового коду – це **біт**.

Біт – один розряд двійкового коду

Байт – послідовність із восьми бітів.

1 кілобайт (1 Кбайт) – 1024 байти;

1 мегабайт (1 Мбайт) – 1024 Кбайт;

1 гігабайт (1 Гбайт) – 1024 Мбайт;

1 терабайт (1 Тбайт) – 1024 Гбайт;

| Измерения в байтах | | | | | | | | |
|--------------------|--------|-----------|--------------|-----------|---------------|---------|----------|--|
| ГОСТ 8.417-2002 | | | Приставки СИ | | приставки МЭК | | | |
| Название | Символ | Степень | Название | Степень | Название | Символ | Степень | |
| байт | Б | 10^0 | - | 10^0 | байт | В Б | 2^0 | |
| килобайт | кБ | 10^3 | кило- | 10^3 | кибибайт | KiB КиБ | 2^{10} | |
| мегабайт | МБ | 10^6 | мега- | 10^6 | мебибайт | MiB МиБ | 2^{20} | |
| гигабайт | ГБ | 10^9 | гига- | 10^9 | гибибайт | GiB ГиБ | 2^{30} | |
| терабайт | ТБ | 10^{12} | тера- | 10^{12} | тебибайт | TiB ТиБ | 2^{40} | |
| петабайт | ПБ | 10^{15} | пета- | 10^{15} | пебибайт | PiB ПиБ | 2^{50} | |
| ексабайт | ЭБ | 10^{18} | екса- | 10^{18} | ексбибайт | EiB ЭиБ | 2^{60} | |
| зеттабайт | ЗБ | 10^{21} | зетта- | 10^{21} | зебибайт | ZiB ЗиБ | 2^{70} | |
| йоттабайт | ЙБ | 10^{24} | йотта- | 10^{24} | йобибайт | YiB ЙиБ | 2^{80} | |

4. Низькорівневі та високорівневі мови програмування.

Програма являє собою опис тієї послідовності дій, яку повинен виконати комп'ютер. Цей опис з точки зору машини виглядає як послідовність **машинних команд**-команд, які машина вміє виконувати, оскільки механізми їх виконання реалізовано в електронній схемі комп'ютера. Сукупність машинних команд складає так звану машинну мову, **мову низького рівня**. В епоху становлення обчислювальної техніки програмісти створювали програми саме в машинних кодах.

Однак машинна мова є зручною і рідною саме для машини і зовсім незручна для машини. Працюючи над програмою в машинних кодах, програміст змушений насильно

підлаштовувати своє мислення під технічні особливості машини. Внаслідок цього продуктивність праці програміста при роботі в машинних кодах виявляється дуже низькою, особливо при розробці великих за обсягом програм.

Очевидно що вигідніше машину підлаштовувати під людські потреби та уявлення про зручність. Для цього призначені *мови програмування високого рівня*, які являють собою посередника між людиною з притаманним їй способом мислення і машиною, яка розуміє лише машинні команди. З одного боку, мова високого рівня розробляється так , щоб текст програми ясно відображав сенс та задум програми. З іншого боку, мова високого рівня проектується таким чином, щоб будь-який текст можна було б перекласти з неї на мову машинних кодів. Такий переклад робить не людина, а спеціальна програма, яка називається *транслятором*.

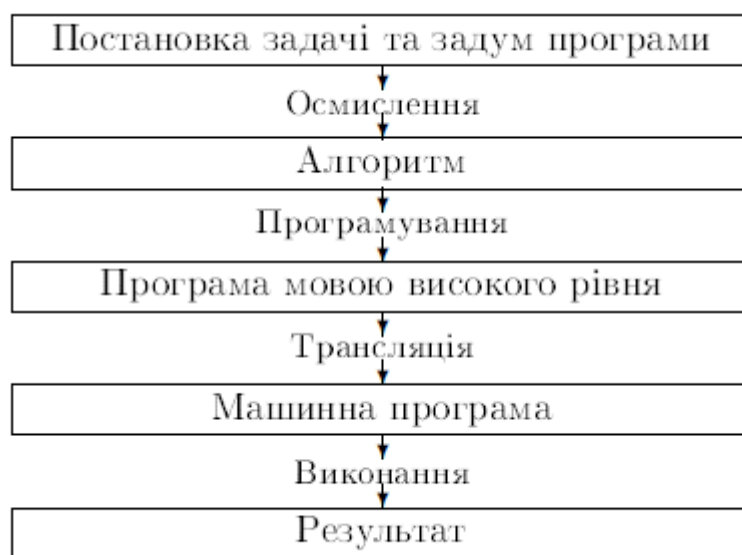


Схема процесу створення та використання програм