

**Національний технічний університет України
«Київський політехнічний інститут»**

**Факультет прикладної математики
Кафедра системного програмування і спеціалізованих
комп'ютерних систем**

КОНСПЕКТ

лекцій

з дисципліни

«ІНЖЕНЕРІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ -1. ОСНОВИ ПРОЕКТУВАННЯ ТРАНСЛЯТОРІВ»

Автор: к.т.н., доцент Марченко О.І.

**Copyright © 2006 – 2013, Марченко О.І.
All rights reserved**

Київ – 2013

Лекція №1

Вступ

Основні визначення

Транслятор – це програма чи комплекс програм, що здійснюють переклад тексту, написаного однією мовою програмування (вхідна мова), в текст, поданий іншою мовою (вихідна мова).

Розрізняють транслятори двох видів:

- 1) компілюючого типу;
- 2) інтерпретуючого типу;

Компілятор – це транслятор, для якого вхідною мовою є мова високого рівня (наприклад C, Pascal, Algol, та інші), а вихідною – мова асемблера чи мова машинних команд. До того ж переклад вхідної програми вихідною мовою виконується одразу цілком.

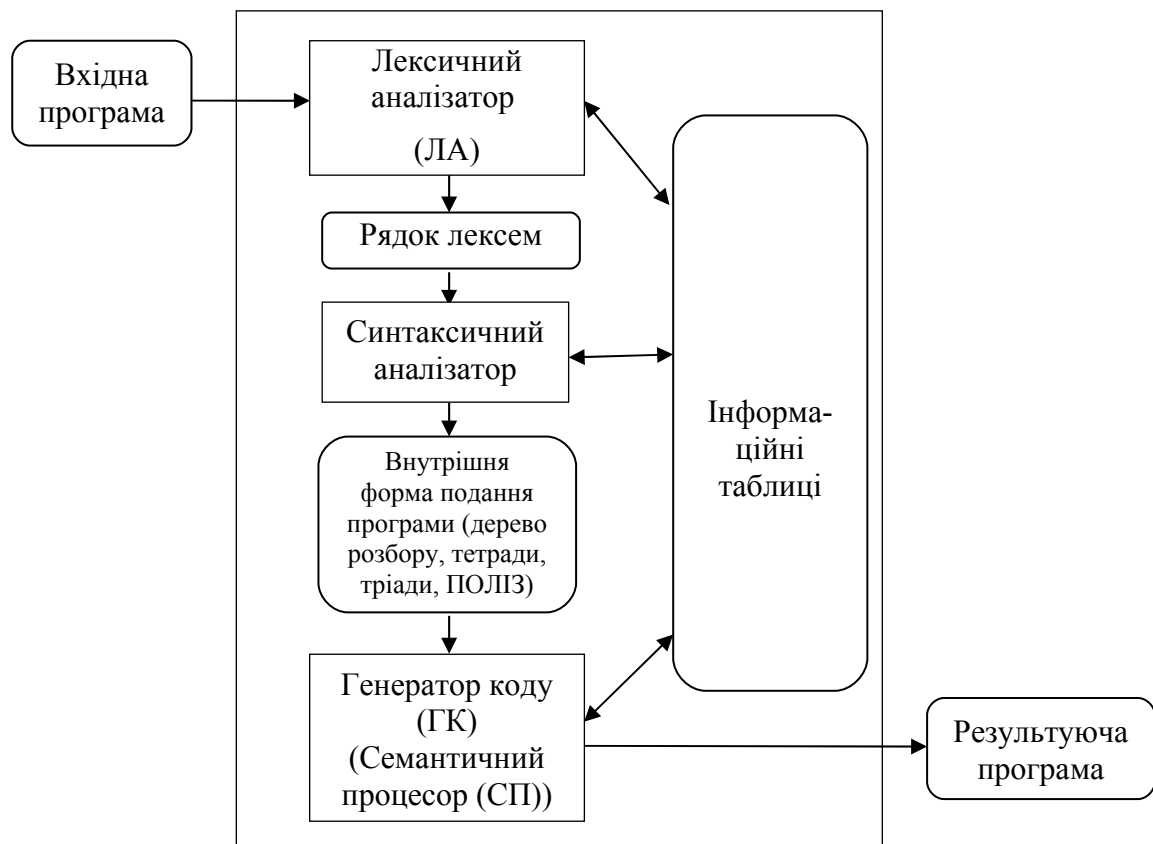
Асемблер – це компілятор, в якому вхідною мовою є мова асемблера, а вихідною – мова машинних команд.

Варто зауважити, що вхідною та вихідною програмами компілятора завжди є текст.

Інтерпретатор – це транслятор, що здійснює пооператорний переклад тексту програми вихідною мовою з одночасним виконанням цих операторів.

На виході інтерпретатора отримуємо результат роботи вхідної програми.

Структура транслятора (компілятора)



Семантичний процесор – це інша назва генератора коду програми.

Лексичний аналізатор (scanner, сканер) здійснює перетворення вхідного тексту програми, що подана рядком символів, в рядок лексем, поданий в цифровій формі, а також знаходить лексичні помилки.

Лексема – найменша одиниця інформації, яка обробляється синтаксичним аналізатором. Приклади лексем:

- 1) односимвольні роздільники (, ; .), знаки операцій (+, -, *, / тощо), що є односимвольними роздільниками;
- 2) багатосимвольні роздільники (<=, <>, тощо);
- 3) ідентифікатори;
- 4) константи;
- 5) ключові слова (**for**, **while** та інші).

Синтаксичний аналізатор (parser, парсер) здійснює декомпозицію вхідної програми, поданої рядком лексем, у структурні одиниці мови (оператори, описи, декларації тощо) згідно граматики вхідної мови, а також знаходить синтаксичні помилки.

Дерево розбору є однією із внутрішніх форм подання вхідної програми, що містить структурні одиниці мови, а також зв'язки між ними.

Генератор коду (семантичний процесор) перетворює вхідну програму, подану одною із внутрішніх форм, в текст (оператори, команди) вихідною мовою на основі семантики вхідної мови.

Для опису синтаксису і семантики мови використовуються так звані **метамови**, тобто мови, за допомогою яких можна описати інші мови.

Для опису синтаксису використовується метасинтаксична мова.

Для опису семантики використовується метасемантична мова.

Побудова лексичного і синтаксичного аналізаторів базується на теорії формальних граматик.

Для побудови генератора коду використовуються мови та методи опису семантики мов програмування.

Основні поняття теорії формальних граматик

Алфавіт мови (T) – попередньо визначена довільна непуста скінченна множина символів (так званих **термінальних** символів) для побудови речень (рядків) мови.

T^* – множина всіх рядків алфавіту T (тобто тих, що складаються лише із термінальних символів), включаючи порожній рядок ϵ .

T^+ – множина всіх рядків в алфавіті T, не включаючи порожній рядок ϵ .

$T^* = T^+ \cup \epsilon$.

Позначимо довільну формальну граматику літерою **G**.

Формальна граMATика **G** є сукупністю чотирьох понять

$$G = (T, N, P, S),$$

де **T** – множина термінальних символів граматики (терміналів), тобто алфавіт мови;

N – множина нетермінальних символів граматики (нетерміналів).

Множина $V = T \cup N$ – це об'єднання множин термінальних і нетермінальних символів.

P – множина правил граматики виду $\alpha \rightarrow \beta$,

де $\alpha \in V^+$, $\beta \in V^*$

' \rightarrow ' означає «є за визначенням».

S – аксіома граматики, один із нетермінальних символів граматики ($S \in N$), з якого починається опис граматики.

Термінали (одиничні символи алфавіту вхідної мови) позначатимемо **маленькими латинськими** літерами.

Нетермінали (одиничні нетермінальні символи мови) позначатимемо **великими латинськими** літерами чи **рядками символів у кутових дужках** **< i >** (наприклад: <буква>, <оператор>).

Рядки, що складаються в загальному випадку із термінальних і нетермінальних символів, тобто ті, що належать множині V^+ , позначатимемо маленькими грецькими літерами.

Особливі випадки, коли рядок належить тільки T^+ або N^+ , будуть зазначені окремо.

Приклад. Розглянемо граматику, що породжує цілі двійкові константи, наприклад, 0, 10, 110 та інші.

Множина правил **P** може бути, наприклад, такою:

1. <рядок> \rightarrow <двійковий рядок>
2. <двійковий рядок> \rightarrow <цифра>
3. <двійковий рядок> \rightarrow <двійковий рядок><цифра> – рекурсивне правило
4. <цифра> \rightarrow 0
5. <цифра> \rightarrow 1

$$T = \{0, 1\}$$

$$N = \{<рядок>, <двійковий рядок>, <цифра>\}$$

$$S = <рядок>$$

Правило називається **рекурсивним**, якщо один і той самий нетермінальний символ знаходиться зліва і справа від стрілки.

Ліва рекурсія: $<X> \rightarrow <X>\alpha$

Права рекурсія: $<X> \rightarrow \alpha<X>$

Центральна рекурсія: $<X> \rightarrow \alpha<X>\beta$

Множинна рекурсія: $<X> \rightarrow <X><X>\alpha$ і т.п.

Непряма рекурсія: $<X> \rightarrow \alpha<Y>$

$$<Y> \rightarrow <X>\beta$$

Де α і β – рядки, що належать V^+ ($\alpha, \beta \in V^+$)

Завдання. Написати послідовність виводу рядка 011 за вищевказаною граматикою.

$\langle \text{рядок} \rangle \Rightarrow^1 \langle \text{двійковий рядок} \rangle \Rightarrow^3 \langle \text{двійковий рядок} \rangle \langle \text{цифра} \rangle \Rightarrow^3$
 $\langle \text{двійковий рядок} \rangle \langle \text{цифра} \rangle \langle \text{цифра} \rangle \Rightarrow^2 \langle \text{цифра} \rangle \langle \text{цифра} \rangle \langle \text{цифра} \rangle \Rightarrow^4$
 $0 \langle \text{цифра} \rangle \langle \text{цифра} \rangle \Rightarrow^5$
 $01 \langle \text{цифра} \rangle \Rightarrow^5 011$

Можна використовувати також графічне подання процесу виводу – **дерево розбору (виводу)**, де

- корінь дерева – аксіома граматики;
- вершини дерева – нетермінали;
- листки – термінальні символи;

В загальному випадку вершини дерева можуть бути простими (складатися з одного нетерміналу) і багатокомпонентними (включати декілька нетерміналів).

