

## 4.3 Средства инструментального программирования

*Инструментальными* называются средства, предназначенные для разработки программных продуктов любого назначения.

К ним относятся языки программирования и среды программирования.

Языки программирования обычно классифицируются исторически, отражая эволюцию самих ЭВМ.

### ***Машинные языки (МЯ).***

Машинный язык представляет собой свод правил кодирования в числовом виде определенных действий.

Каждое действие – это команда.

*Команда определяет некоторую часть процесса вычисления (машинную операцию).*

## Средства инструментального программирования(2)

В команде указывается:

вид действия (код команды);

место хранения информации (адрес исходной информации);

адрес результата ;

следующая команда (счетчик адреса команд).

Составление программы на машинном языке носит характер решения сложной комбинаторной задачи, так как параллельно с кодированием программисту приходится решать проблемы с распределением памяти.

Распределение памяти – это размещение в ОП всей информации, относящейся к задаче (команд, исходных чисел, промежуточных данных и результатов).

Эти процессы сильно взаимосвязаны – пока не известны все команды, нельзя разместить данные, а пока не известны адреса данных – нельзя написать команду.

## Средства инструментального программирования(3)

Достоинство МЯ:

они обладают необходимой точностью;

достаточно формализованы для адекватного представления алгоритма задачи;

программы на МЯ работают очень быстро.

Недостатки:

1) Неудобство использования:

двоичная система счисления;

знание устройства машины и процессов, происходящих в ней;

знание организации памяти.

2) Сложность отладки и ее трудоемкость:

синтаксис (неправильные команды и адреса)

логические ошибки (многократный прогон и пошаговое вып.)

## Средства инструментального программирования(4)

### Машинно-ориентированные языки (МОЯ).

Это группа языков, в основе которых по-прежнему лежит система команд какой либо машины (ассемблер ЕС ЭВМ, ЯСК «Минск-32», автокод М-222).

Отличия от МЯ:

введение символических адресов (кодирование отделено от распределения памяти);

введение символических (мнемонических) имен команд (каждая группа команд имеет свои имена).

Достоинства:

1) Упрощение составление программ

за счет мнемонических команд и адресов;

за счет макрогенерации (сокращение длины команд);

удобных средств описания полей данных;

автоматического синтаксического анализа

## **Средства инструментального программирования(5)**

Недостатки:

синтаксис МОЯ далек от записи формул (чтобы запрограммировать формулу – много мелких операций);  
время написания и отладки программ по-прежнему велико;  
программа достаточно большая по объему.

### **Алгоритмические языки (ЯВУ).**

Создание ЯВУ позволило предоставить программистам средства значительно более гибкой и разнообразной структуры.

Применение этих языков приблизило описание алгоритмов для ЭВМ к общепринятой форме записи методов решения задач определенного класса.

## Средства инструментального программирования(6)

### *Свойства ЯВУ:*

позволяют удобно и экономично записывать формулы, определяющих вычислительную часть задачи;

позволяют определять и задавать структуру и формат данных различных видов;

позволяют статически и динамически выделять память под информацию;

достаточно точны и формализованы для автоматического перевода программ на машинный язык;

легко позволяет найти и устранить ошибку;

удобен для изучения.

### *ЯВУ служат:*

средством мышления;

средством общения;

посредником между человеком и ЭВМ.

## Средства инструментального программирования(7)

Однако:

программирование на ЯВУ требует от языка кодировать алгоритмы в привычном для человека виде – в виде предложений

автоматический же перевод на язык машинных команд требует, чтобы ЯВУ строились на основе использования строгих правил построения предложений (синтаксиса) и определенного набора стандартных символов и знаков из которых строились бы синтаксические конструкции (алфавит). Все это вместе определяет грамматику ЯВУ.

Поэтому, для перевода программы на язык, понятный машине, и проверки правильности программы нужны специальные программы, называемые компилирующими программами.

## Средства инструментального программирования(8)

### *Классификация компилирующих программ*

**Транслятор** – программа, которая переводит программу, написанную на одном языке, в эквивалентную ей программу, написанную на другом языке.

**Компилятор** – транслятор с языка высокого уровня на машинный язык или язык ассемблера.

**Ассемблер** – транслятор с языка Ассемблера на машинный язык.

**Интерпретатор** – программа, которая принимает исходную программу и выполняет ее, не создавая программы на другом языке.

**Макрогенератор** (для компиляторов – **препроцессор**) – программа, которая обрабатывает исходную программу, как текст, и выполняет в нем замены указанных символов на подстроки. Макрогенератор обрабатывает программу до трансляции.



## Средства инструментального программирования(9)

Существует множество алгоритмических языков программирования.

Это определяется множеством сфер применения компьютеров и, как следствие, множеством классов решаемых на компьютере задач.

Среди этого множества можно отметить такие языки, как:

Fortran (Formula translator), ориентированный на написание программ, решающих научные или вычислительные задачи;

COBOL, предназначенный для разработки бизнес-приложений;

Pascal, язык общего назначения;

C, являющийся стандартным процедурным языком программирования.

## **Средства инструментального программирования(10)**

Существенным упрощением разработки программ является наличие библиотек стандартных программ.

Эти библиотеки можно разделить на системные и прикладные.

К системным библиотекам относятся такие библиотеки, в которых содержатся стандартные программы для выполнения работы с устройствами и файлами, программы работы с памятью, т.е. те стандартные программы, которые работают с ресурсами системы.

Прикладные библиотеки обычно включают в себя реализацию различных математических вычислений: интегрирование, статистическая обработка, различные преобразования и т.п.

## **Средства инструментального программирования(11)**

Особое место занимают библиотеки, в состав которых входят программы, предназначенные для организации интерфейса между программой и пользователем, а также библиотеки построения графических изображений.

Библиотеки могут быть встроены в состав компилятора или подключаться в процессе компиляции или связывания (линковки), а также подключаться к программе в процессе ее выполнения – динамические библиотеки.

Наличие библиотек позволяет сократить время разработки программ за счет повторного использования ранее написанных программ, реализующих необходимые нам функции.

## Средства инструментального программирования(12)

### Естественно-ориентированные языки (ЕОЯ).

Эти языки позволяют преодолеть психологический барьер между человеком и машиной.

ЕОЯ позволяют специалисту общаться с машиной в терминах его проблемной области, близких и понятных ему.

Эти языки, в отличие от ЯВУ, не являются универсальными, а ориентированы на применение в каких либо областях знаний.

Примерами естественно-ориентированных языков программирования могут служить

Утопист, Симула, Модула, ПРОЛОГ.

## Средства инструментального программирования(13)

### Среды программирования.

Разработка системного и прикладного программного обеспечения составляет существенную часть работ на ЭВМ.

Обязательным при этом является этап создания текста на каком-либо ЯВУ. Такой текст называют исходным.

Для создания его можно использовать любой текстовый редактор.

Созданный затем исходный текст транслируется на МЯ с помощью специальных программ.

Эти программы необходимо вызвать и передать им в качестве исходной информации набранный текст.

В процессе трансляции могут возникнуть ошибки. При их обнаружении процесс следует повторить, внося изменения в исходный текст.

## Средства инструментального программирования(14)

Процесс повторяется, пока все синтаксические ошибки не будут исправлены.

После этого компилятор создает объектный код программы на МЯ (\*.OBJ). Однако, к нему еще не присоединены библиотечные и стандартные программы.

Для их подключения используется специальная программа компоновщик (редактор связей). Ее нужно вызвать на выполнение и передать ей в качестве исходного файла объектный код.

Компоновщик подсоединит библиотеки и создаст исполнительный файл программы (\*.EXE).

Однако, если при компоновке возникли ошибки, повторяется весь процесс начиная от редактора текстов.

После исправления всех ошибок компоновки, программа запускается на выполнение.

## Средства инструментального программирования(15)

Однако, в процессе выполнения программы пользователь может получить не совсем верный результат или вообще не получить его.

Это говорит о наличие ошибок выполнения (логические ошибки).

Для поиска, локализации и исправления ошибки необходимо провести отладку программы.

Для этого служат специальные программы – отладчики.

Программу отладчик запускают, передав ей в качестве исходной информации исполняемый файл программы.

После нахождения ошибки весь процесс от исправления текста редактором до выполнения повторяется вновь.

Для упрощения этого процесса были разработаны специальные среды, интегрирующие в себе все необходимые средства разработки.

## Средства инструментального программирования(16)

Интегри́рованная среда́ разрабо́тки, ИСР (*IDE, Integrated development environment* или *integrated debugging environment*) — система программных средств, используемая программистами для разработки программного обеспечения (ПО).

Обычно среда разработки включает в себя:

- Текстовый редактор

- Компилятор и/или интерпретатор

- Средства автоматизации сборки

- Отладчик

- Информационную помощь.

Появление сред программирования дало новую жизнь многим популярным языкам программирования таким как C, Pascal.



## Средства инструментального программирования(17)

Интегрированные среды разработки были созданы для того, чтобы максимизировать производительность программиста благодаря тесно связанным компонентам с простыми пользовательскими интерфейсами.

Это позволяет разработчику сделать меньше действий для переключения различных режимов, в отличие от дискретных программ разработки.

Однако, так как IDE является сложным программным комплексом, то лишь после долгого процесса обучения среда разработки сможет качественно ускорить процесс разработки ПО.

Обычно IDE ориентирована на определенный язык программирования, предоставляя набор функций, который наиболее близко соответствует парадигмам этого языка программирования.

Однако, есть некоторые IDE с поддержкой нескольких языков, такие как Eclipse, ActiveState, Komodo, последние версии Microsoft Visual Studio, WinDev и Xcode

## **Средства инструментального программирования(18)**

Частный случай ИСП — **среды визуальной разработки**, которые включают в себя возможность Визуального редактирования интерфейса программы.

**Среда визуальной разработки** — среда разработки программного обеспечения, в которой наиболее распространенные блоки программного кода представлены в виде графических объектов.

Применяются в основном для создания прикладных программ и разработки графического интерфейса пользователя (GUI).

Некоторые визуальные среды разработки имеют собственный формат хранения проекта и при переходе на другую среду может возникнуть непереносимость свойств проекта и некоторых частей проекта, таких, как собственные библиотеки используемой среды разработки.

## Средства инструментального программирования(19)

Преимущества:

- быстрота разработки

- лёгкость освоения

- стандартизация внешнего вида программ.

Недостатки:

- как правило, привязка к конкретной среде разработки, связанное с проблематичностью перехода на другую среду разработки

- затруднённое использование нестандартных компонентов

- наличие недокументированных особенностей компонент.

Следует учитывать некоторые изменения, которые могут вноситься в язык программирования средой.

Например - среда разработки Delphi (за основу взят Pascal, но по сути это уже новый язык программирования).

Среду разработки как и язык программирования следует выбирать на этапе проектирования ПО.