

## 9. Понятия интегрированной среды разработки (IDE), компилятора, транслятора, интерпретатора, отладчика

IDE:

[http://de.ifmo.ru/bk\\_netra/page.php?dir=3&tutindex=25&index=81&layer=1](http://de.ifmo.ru/bk_netra/page.php?dir=3&tutindex=25&index=81&layer=1)

IDE:

[http://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D0%B3%D1%80%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%B0%D1%8F\\_%D1%81%D1%80%D0%B5%D0%B4%D0%B0%D1%80%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B8](http://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D0%B3%D1%80%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D1%80%D0%B5%D0%B4%D0%B0%D1%80%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B8)

Транслятор:

<http://ru.wikipedia.org/wiki/%D0%A2%D1%80%D0%B0%D0%BD%D1%81%D0%BB%D1%8F%D1%82%D0%BE%D1%80>

Компилятор:

<http://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D0%B8%D0%BB%D1%8F%D1%82%D0%BE%D1%80>

Интерпретатор:

<http://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D0%BF%D1%80%D0%B5%D1%82%D0%B0%D1%82%D0%BE%D1%80>

Отладчик:

[http://de.ifmo.ru/bk\\_netra/page.php?index=76&layer=1&tutindex=25](http://de.ifmo.ru/bk_netra/page.php?index=76&layer=1&tutindex=25)

**Интегрированная среда разработки** - это совокупность программных средств, поддерживающая все этапы разработки программного обеспечения от написания исходного текста программы до ее компиляции и отладки, и обеспечивающая простое и быстрое взаимодействие с другими инструментальными средствами (например программным отладчиком-симулятором).

Строго говоря, интегрированные среды разработки не относятся к числу средств отладки, тем не менее обойти вниманием данный класс программных средств, существенно облегчающий и ускоряющий процесс разработки и отладки микропроцессорных систем было бы неправильно.

При традиционном подходе, начальный этап написания программы строится следующим образом:

1. Исходный текст набирается при помощи какого-либо текстового редактора. По завершении набора, работа с текстовым редактором прекращается и запускается кросс компилятор. Как правило, вновь написанная программа содержит синтаксические ошибки, и компилятор сообщает о них на консоль оператора.
2. Вновь запускается текстовый редактор, и оператор должен найти и устранить выявленные ошибки, при этом сообщения о характере ошибок выведенные компилятором уже не видны, так как экран занят текстовым редактором.

И этот цикл может повторяться не один раз. Если программа имеет большой объем, собирается из различных частей, и подвергается длительному редактированию или модернизации, то даже этот начальный этап может потребовать много сил и времени. После этого наступает этап отладки программы и к редактору с компилятором добавляется эмулятор или симулятор, за работой которого хотелось бы следить прямо по тексту программы в текстовом редакторе.

Избежать большого объема однообразных действий и тем самым существенно повысить эффективность процесса разработки и отладки позволяют т.н. интегрированные среды (оболочки) разработки (Integrated Development Environment, IDE).

Работа в интегрированной среде дает программисту:

- Возможность использования встроенного многофайлового текстового редактора, специально ориентированного на работу с исходными текстами программ;
- Диагностика выявленных при компиляции ошибок, и исходный текст программы, доступный редактированию, выводятся одновременно в многооконном режиме;

- Возможность организации и ведения параллельной работы над несколькими проектами. Менеджер проектов позволяет использовать любой проект в качестве шаблона для вновь создаваемого проекта;
- Перекомпиляции подвергаются только редактировавшиеся модули;
- Возможность загрузки отлаживаемой программы в имеющиеся средства отладки, и работы с ними без выхода из оболочки;
- Возможность подключения к оболочке практически любых программных средств.

**Транслятор** — программа или техническое средство, выполняющее преобразование программы, представленной на одном из языков программирования, в программу на другом языке и, в определённом смысле, равносильную первой.

Транслятор обычно выполняет также диагностику ошибок, формирует словари идентификаторов, выдаёт для печати тексты программы и т. д.

Язык, на котором представлена входная программа, называется *исходным языком*, а сама программа — *исходным кодом*. Выходной язык называется *целевым языком* или *объектным кодом*.

Цель трансляции — преобразовать текст с одного языка на другой, который понятен адресату текста. В случае программ-трансляторов, адресатом является техническое устройство (процессор) или программа - интерпретатор.

Язык процессоров (машинный код) обычно является низкоуровневым. Существуют платформы, использующие в качестве машинного язык высокого уровня (например, APX-432), но они являются исключением из правила в силу сложности и дороговизны. Транслятор, который преобразует программы в машинный язык, принимаемый и исполняемый непосредственно процессором, называется **компилятором**.

Процесс компиляции, как правило, состоит из нескольких этапов: лексического, синтаксического и семантического анализов, генерации на основе результатов анализов промежуточного кода, оптимизации промежуточного кода и генерации результирующего объектного кода, в данном случае машинного. Помимо этого, программа, как правило, имеет внешние инфраструктуры: сервисы, предоставляемые операционной системой и сторонние библиотеки (например, файловый ввод-вывод или графический интерфейс), для чего машинный код программы необходимо *связать* с этими сервисами и функциями библиотек. Связывание со статическими библиотеками выполняется *редактором связей* или компоновщиком (который может представлять собой отдельную программу или быть частью компилятора), а с операционной системой и динамическими библиотеками связывание выполняется при начале исполнения программы загрузчиком.

Достоинство компилятора: программа компилируется один раз и при каждом выполнении не требуется дополнительных преобразований. Соответственно, не требуется наличие компилятора на целевой машине, для которой компилируется программа. Недостаток: отдельный этап компиляции замедляет написание и отладку и затрудняет исполнение небольших, несложных или разовых программ.

Другой метод реализации — когда программа исполняется с помощью **интерпретатора** вообще без трансляции. Интерпретатор программно моделирует машину, цикл выборки-исполнения которой работает с командами на языках высокого уровня, а не с машинными командами. Такое программное моделирование создаёт виртуальную машину, реализующую язык. Этот подход называется чистой интерпретацией. Чистая интерпретация применяется, как правило, для языков с простой структурой (например, АПЛ или Лисп). Интерпретаторы командной строки обрабатывают команды в скриптах в UNIX или в пакетных файлах (.bat) в MS-DOS также как правило в режиме чистой интерпретации.

Достоинство чистого интерпретатора: отсутствие промежуточных действий для трансляции упрощает реализацию интерпретатора и делает его удобнее в

использовании, в том числе в диалоговом режиме. Недостаток — интерпретатор должен быть в наличии на целевой машине, где должна исполняться программа. Также, как правило, имеется более или менее значительный проигрыш в скорости. А свойство чистого интерпретатора, что ошибки в интерпретируемой программе обнаруживаются только при попытке выполнения команды (или строки) с ошибкой, можно признать как недостатком, так и достоинством.

**Отладчик** является модулем среды разработки или отдельным приложением, предназначенным для поиска ошибок в программе. Отладчик позволяет выполнять пошаговую трассировку, отслеживать, устанавливать или изменять значения переменных в процессе выполнения программы, устанавливать и удалять контрольные точки или условия остановки и т. д. В режиме трассировки разработчик видит последовательность выполнения команд и значения переменных на данном шаге выполнения программы, что позволяет легче обнаруживать ошибки. Трассировка может быть начата и окончена в любом месте программы, выполнение программы может останавливаться на каждой команде или на точках останова, трассировка может выполняться с заходом в процедуры и без заходов.