

Структура и функции
программного обеспечения.
Ввод и вывод информации в
микроконтроллерную систему

Структура и функции системного программного обеспечения

Системное ПО, в состав которого входят *операционная система*, трансляторы языков и обслуживающие программы, управляет доступом к аппаратному обеспечению. *Прикладное ПО*, такое как языки программирования и различные пользовательские приложения, работает с аппаратным обеспечением через слой системного ПО. Пользователи, в свою очередь, взаимодействуют с прикладным программным обеспечением.

Программные системы можно классифицировать по различным признакам. Рассмотрим классификацию, в которой основополагающим признаком является сфера (область) использования программных продуктов:

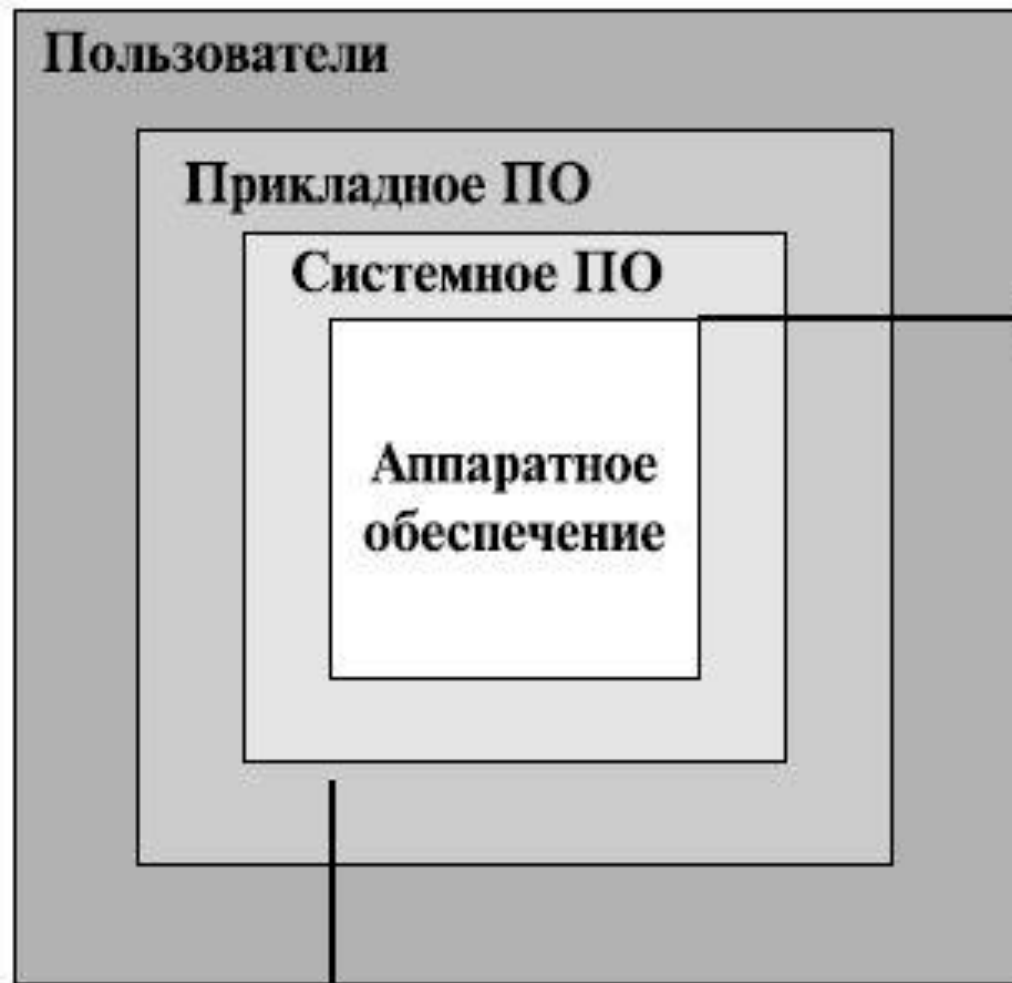
- ▶ аппаратная часть автономных компьютеров и сетей ЭВМ;
- ▶ функциональные задачи различных предметных областей;
- ▶ технология разработки программ.

Для поддержки информационной технологии в этих областях выделяют соответственно три класса программных продуктов:

- ▶ системное программное обеспечение;
- ▶ прикладное программное обеспечение;
- ▶ инструментальное программное обеспечение.

Системное программное обеспечение (System Software) - совокупность программ и программных комплексов, предназначенная для обеспечения работы компьютера и сетей ЭВМ. *Системное программное обеспечение* выполняет следующие задачи:

- ▶ создание операционной среды функционирования других программ;
- ▶ обеспечение надежной и эффективной работы самого компьютера и вычислительной сети;
- ▶ проведение диагностики, локализации сбоев, ошибок и отказов и профилактики аппаратуры компьютера и вычислительных сетей;
- ▶ выполнение вспомогательных технологических процессов (копирование, архивирование, восстановление файлов программ и баз данных и т.д.).



Прикладное ПО

Языки программирования

Системное ПО

Операционная система

- Распределяет ресурсы компьютера
- Планирует использование ресурсов
- Контролирует работу компьютера

Языковые трансляторы

- Интерпретатор
- Компилятор

Утилиты

Обслуживание компьютера и периферийных устройств

Системное программное обеспечение



Прикладное программное обеспечение

Прикладные программы предназначены для того, чтобы обеспечить применение вычислительной техники в различных сферах деятельности человека. Помимо создания новых программных продуктов, разработчики прикладных программ большие усилия тратят на совершенствование и модернизацию популярных систем, создание их новых версий. Новые версии, как правило, поддерживают старые, сохраняя преемственность, и включают в себя базовый *минимум* (стандарт) возможностей.

Один из возможных вариантов классификации программных средств (ПС), составляющих прикладное *программное обеспечение* (ППО). Как и почти всякая классификация, приведенная на рисунке, не является единственно возможной. В ней представлены даже не все виды прикладных программ. Тем не менее, использование классификации полезно для создания общего представления о ППО.

Несмотря на широкие возможности использования компьютеров для обработки самой разной информации, самыми популярными являются программы, предназначенные для работы с текстами - текстовые редакторы и издательские системы. Текстовыми редакторами называют программы для ввода, обработки, хранения и печатания текстовой информации в удобном для пользователя виде. Эксперты оценивают использование компьютера в качестве печатающей машинки в 80% всего времени задействования техники.

Большую популярность приобрели программы обработки графической информации. Компьютерная *графика* в настоящее время является одной из самых динамично развивающихся областей программного обеспечения. Она включает в себя ввод, обработку и *вывод* графической информации - чертежей, рисунков, фотографий, картин, текстов и т. д. - средствами компьютерной техники. Различные типы графических систем позволяют быстро строить изображения, вводить иллюстрации с помощью сканера или видеокамеры, создавать анимационные ролики. Графические редакторы позволяют пользоваться различным инструментарием художника, стандартными библиотеками изображений, наборами стандартных шрифтов, редактированием изображений, копированием и перемещением фрагментов *по* страницам экрана и др.

ППО

ПС общего
назначения

ПС специального
назначения

ПС профессионального
уровня

Текстовые редакторы
Графические редакторы
Электронные таблицы
СУБД
Пакеты презентаций
Почтовые пакеты
Органайзеры
Издательские системы

Авторские системы
Экспертные системы
Системы поддержки
принятия решений
Гипертекстовые
системы
Системы аудио-
и видеоконференций

САПР
АСНИ
АСУ ОУ
АСУ ТП
Педагогические
комплексы
Системы управления
предприятием
Бухгалтерские системы

Задачи и функции инструментального программного обеспечения

Для инструментального программного обеспечения, как особой разновидности программного обеспечения, характерны общие и частные функции, как и для всего программного обеспечения в целом. Общие функции рассмотрены нами выше, а специализированными функциями, присущими только данному типу программ, являются:

- ▶ 1. Создание текста разрабатываемой программы с использованием специально установленных кодовых слов (языка программирования), а также определенного набора символов и их расположения в созданном файле - синтаксис программы.
- ▶ 2. Перевод текста создаваемой программы в машинно-ориентированный код, доступный для распознавания ЭВМ. В случае значительного объема создаваемой программы, она разбивается на отдельные модули и каждый из модулей переводится отдельно.
- ▶ 3. Соединение отдельных модулей в единый исполняемый код, с соблюдением необходимой структуры, обеспечение координации взаимодействия отдельных частей между собой.
- ▶ 4. Тестирование и контроль созданной программы, выявление и устранение формальных, логических и синтаксических ошибок, проверка программ на наличие запрещенных кодов, а также оценка работоспособности и потенциала созданной программы.

Стадии разработки программного обеспечения

Разработка любой программы, будь то небольшая процедура по обработке поступающей на консоль информации или комплексный программный продукт, состоит из нескольких этапов, грамотная реализация которых является обязательным условием для получения хорошего результата. Четкое следование выверенным временем этапам разработки программного обеспечения становится основополагающим критерием для занимающихся созданием ПО компаний и их заказчиков, заинтересованных в получении превосходно выполняющей свои функции программы. Подробно рассмотрим каждую стадию общепризнанной методологии разработки ПО, чтобы оценить их высокую значимость для достижения поставленной перед исполнителями цели.

► Анализ требований

Самым первым этапом разработки программного обеспечения по праву называется процедура проведения всестороннего анализа выдвинутых заказчиком требований к создаваемому ПО, чтобы определить ключевые цели и задачи конечного продукта. В рамках этой стадии происходит максимально эффективное взаимодействие нуждающегося в программном решении клиента и сотрудников компании-разработчика, в ходе обсуждения деталей проекта помогающих более четко сформулировать предъявляемые к ПО требования. Результатом проведенного анализа становится формирование основного регламента, на который будет опираться исполнитель в своей работе – технического задания на разработку программного обеспечения. ТЗ должно полностью описывать поставленные перед разработчиком задачи и охарактеризовать конечную цель проекта в понимании заказчика.

► Проектирование

Следующий ключевой этап в разработке программного обеспечения – стадия проектирования, то есть моделирования теоретической основы будущего продукта. Самые современные средства программирования позволяют частично объединить этапы проектирования и кодирования, то есть технической реализации проекта, будучи основанными на объектно-ориентированном подходе, но полноценное планирование требует более тщательного и скрупулезного моделирования. Качественный анализ перспектив и возможностей создаваемого продукта станет основой для его полноценного функционирования и выполнения всего комплекса возлагаемых на ПО задач. Одной из составных частей этапа проектирования, к примеру, является выбор инструментальных средств и операционной системы, которых сегодня на рынке присутствует очень большое количество.

В рамках данного этапа стороны должны осуществить:

- ▶ оценку результатов проведенного первоначально анализа и выявленных ограничений;
- ▶ поиск критических участков проекта;
- ▶ формирование окончательной архитектуры создаваемой системы;
- ▶ анализ необходимости использования программных модулей или готовых решений сторонних разработчиков;
- ▶ проектирование основных элементов продукта – модели базы данных, процессов и кода;
- ▶ выбор среды программирования и инструментов разработки, утверждение интерфейса программы, включая элементы графического отображения данных;
- ▶ определение основных требований к безопасности разрабатываемого ПО.

Кодирование

Следующим шагом становится непосредственная работа с кодом, опираясь на выбранный в процессе подготовки язык программирования. Описывать особенности и тонкости самого трудоемкого и сложного этапа вряд ли стоит, достаточно указать, что успех реализации любого проекта напрямую зависит от качества предварительного анализа и оценки конкурирующих решений, с которыми создаваемой программе предстоит “бороться” за право называться лучшей в своей нише. Если речь идет о написании кода для выполнения узкоспециализированных задач в рамках конкретного предприятия, то от грамотного подхода к этапу кодирования зависит эффективность работы компании, заказавшей разработку. Кодирование может происходить параллельно со следующим этапом разработки – тестированием программного обеспечения, что помогает вносить изменения непосредственно по ходу написания кода. Уровень и эффективность взаимодействия всех элементов, задействованных для выполнения сформулированных задач компанией-разработчиком, на текущем этапе является самым важным – от слаженности действий программистов, тестировщиков и проектировщиков зависит качество реализации проекта.

▶ **Тестирование и отладка**

После достижения задуманного программистами в написанном коде следуют не менее важные этапы разработки программного обеспечения, зачастую объединяемые в одну фазу – тестирование продукта и последующая отладка, позволяющая ликвидировать огрехи программирования и добиться конечной цели – полнофункциональной работы разработанной программы. Процесс тестирования позволяет смоделировать ситуации, при которых программный продукт перестает функционировать. Отдел отладки затем локализует и исправляет обнаруженные ошибки кода, “вылизывая” его до практически идеального состояния. Эти два этапа занимают не меньше 30% затрачиваемого на весь проект времени, так как от их качественного исполнения зависит судьба созданного силами программистов программного обеспечения. Нередко функции тестировщика и отладчика исполняет один отдел, однако самым оптимальным будет распределить эти обязанности между разными исполнителями, что позволит увеличить эффективность поиска имеющихся в программном коде ошибок.

▶ **Внедрение**

Процедура внедрения программного обеспечения в эксплуатацию является завершающей стадией разработки и нередко происходит совместно с отладкой системы. Как правило, ввод в эксплуатацию ПО осуществляется в три этапа:

- первоначальная загрузка данных;
- постепенное накопление информации;
- вывод созданного ПО на проектную мощность.

Ключевой целью поэтапного внедрения разработанной программы становится постепенное выявление не обнаруженных ранее ошибок и недочетов кода. В рамках этого этапа разработки программного обеспечения и заказчик, и исполнитель могут столкнуться с рядом достаточно узкого спектра ошибок, связанных с частичной рассогласованностью данных при их загрузке в БД, а также срывов выполнения программных процедур в связи с применением многопользовательского доступа. Именно на этой стадии выкристаллизовывается окончательная картина взаимодействия пользователя с программой, а также определяется степень лояльности последнего к разработанному интерфейсу. Если выход системы на проектную мощность после ряда проведенных доработок и улучшений произошел без особых осложнений, значит предварительная работа над проектом и реализация предыдущих стадий разработки осуществлялась правильно.

▶ **Заключение**

Создание даже небольшого и технически простого ПО зависит от четкого выполнения каждой фазы, то есть деятельности всех отделов, задействованных в процессе разработки. Четкий план выполнения необходимых мероприятий с указанием конечных целей становится неотъемлемой частью работы разработчиков, планирующих оставаться широко востребованными на рынке труда специалистами. Только правильно составленное техническое задание позволит добиться нужного результата и осуществить разработку по-настоящему качественного и конкурентного ПО для любой платформы – серверной, стационарной или мобильной. Неотъемлемой частью завершающего этапа разработки программного обеспечения также является последующая техническая поддержка созданного продукта в процессе его эксплуатации на предприятии заказчика. Грамотно организованная служба техподдержки зачастую становится ключевым фактором при выборе исполнителя в рамках достижения поставленной цели.

Ввод информации в микроконтроллерную систему

Функционирование *микропроцессорной системы*, особенно ориентированной на системы управления, во многом определяется взаимодействием *микропроцессора* с внешними устройствами. Такое взаимодействие может строиться по трем основным схемам:

- ▶ программно управляемый обмен;
- ▶ обмен по готовности внешнего устройства;
- ▶ обмен по прерыванию.

В первом случае *микроконтроллер* при возникновении необходимости обмена обращается к ВУ и производит *обмен информацией*. Однако при этом *программа МК* должна быть составлена таким образом, чтобы к моменту обмена *внешнее устройство* было безусловно к нему готово.

Второй механизм обмена (**обмен по готовности**) предполагает перед началом обмена проверку состояния готовности внешнего устройства.

Обмен производится лишь при положительном результате такой проверки. Этот способ используется в основном в тех случаях, когда *микроконтроллер* не занят выполнением каких-либо иных задач и его работа строится по циклу "считывание показаний датчика - обработка - выдача управляющих воздействий - считывание показаний датчика - ...".

Вывод информации из микроконтроллерной системы

Обмен по прерыванию предусматривает *прерывание* естественного порядка выполнения программы в случае запроса на обмен от внешнего устройства, который может поступить на МК в любой момент асинхронно по отношению к работе самого *микроконтроллера*. Появление запроса ведет к передаче управления специальной программе - **обработчику прерывания**, которая и выполняет все необходимые для обмена действия, после чего возвращает управление основной программе. Такой способ обмена может быть использован в тех случаях, когда запросы от внешних устройств поступают относительно редко, требуют быстрой реакции на них и в то же время на *микроконтроллер* возложено решение ряда дополнительных задач, выполнение которых может проходить не в реальном масштабе времени ("фоновые задачи").

Эти варианты взаимодействия можно проиллюстрировать бытовым примером. Предположим, необходимо вскипятить чайник. Для этого чайник ставится на плиту и зажигается огонь. **Программно управляемый обмен** соответствует ситуации, когда в определенный момент чайник снимается с плиты. При этом момент должен быть выбран таким образом, чтобы вода уже закипела, но еще не выкипела.

При **вводе по готовности** вы постоянно поднимаете крышку чайника, чтобы проверить, не кипит ли вода. Естественно, в это время никакой другой работой заниматься невозможно. Однако при этом момент закипания будет определен достаточно точно.

Использование **прерываний при обмене** информацией требует дополнительных программно-аппаратных средств. В данном случае это соответствует приобретению свистка для чайника. После включения плиты можно заниматься другой работой. О моменте закипания воды просигнализирует свисток, после чего следует прервать выполняемую работу и переключить свое внимание на чайник. Если при этом зазвонит, например, телефон, что можно рассматривать как еще один **запрос прерывания**, то в действие должна вступать уже система приоритетов и возможность обработки вложенных прерываний.

Элементы инструментального и системного программного обеспечения

В первом приближении все программы, работающие на компьютере, можно условно разделить на три категории:

- ▶ прикладные программы, непосредственно обеспечивающие выполнение необходимых пользователям работ;
- ▶ системные программы, выполняющие различные вспомогательные функции, например:
 - ▶ управление ресурсами компьютера;
 - ▶ создание копий используемой информации;
 - ▶ проверка работоспособности устройств компьютера;
 - ▶ выдача справочной информации о компьютере и др.;
- ▶ инструментальные программные системы, облегчающие процесс создания новых программ для компьютера.

Системное

Это часть системы, которая помогает следить за аппаратной стороной ПК и управлять ею. Сюда входят программы, контролирующие работу оперативной памяти, центрального процессора, видеокарты, устройств ввода и вывода информации, сетевые подпрограммы.

Таким ПО считается:

- ▶ Драйверы – утилиты небольшого размера, функционирование которых заключается в обеспечении корректной работы остальных элементов оборудования;
- ▶ ОС;
- ▶ Дополнения – языковые пакеты или настройки расширения экрана.

Основное отличие системной разновидности считается то, что она не рассчитана на выполнение конкретной поставленной задачи. Она необходима, чтобы обеспечивать бесперебойную работу остальных частей компьютера. Ее можно назвать посредником между оборудованием – «железом» и программным кодом.

► Инструментальное

Специфическое обеспечение любой компьютерной техники. Его можно было бы отнести к прикладному, но из-за специфики применения его выделили в отдельный вид. Основная функция – отладка, настройка, переписывание программного кода.

Сюда входят компиляторы, отладчики, переводчики высокого уровня, редакторы, интерпретаторы и другие средства. Они необходимы, потому что техника не понимает человеческих слов. Чтобы ей «объяснить», что надо сделать, требуется специальный «машинный язык».

Постоянно пользоваться этим кодом базовым пользователям довольно сложно, поэтому были разработаны системы, которые позволяют переводить обычную речь в двоичную, привычную для ПК.