

Операционные системы

Автор В.А.Серков

Интерфейсы операционных
систем

Основные функции, которые выполняются операционной системой по соответствующим запросам от задач

Управление процессами

- запуск, приостановка и снятие задачи с выполнения;
- задание или изменение приоритета задачи;
- взаимодействие задач между собой (механизмы сигналов, семафорные примитивы, очереди, конвейеры, почтовые ящики);
- вызов удаленных процедур (Remote Procedure Call, RPC).

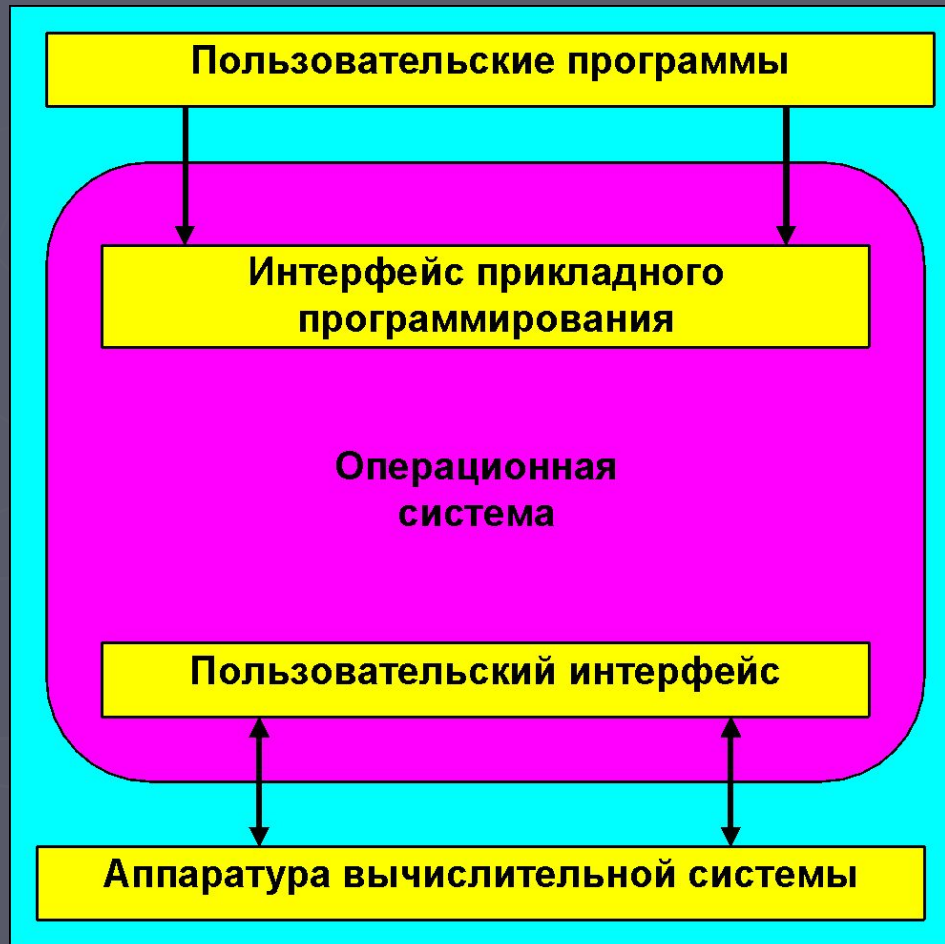
Управление памятью

- запрос на выделение блока памяти;
- освобождение памяти;
- изменение параметров блока памяти (например, память может быть заблокирована процессом либо предоставлена в общий доступ);
- отображение файлов на память (имеется не во всех системах).

Управление вводом-выводом

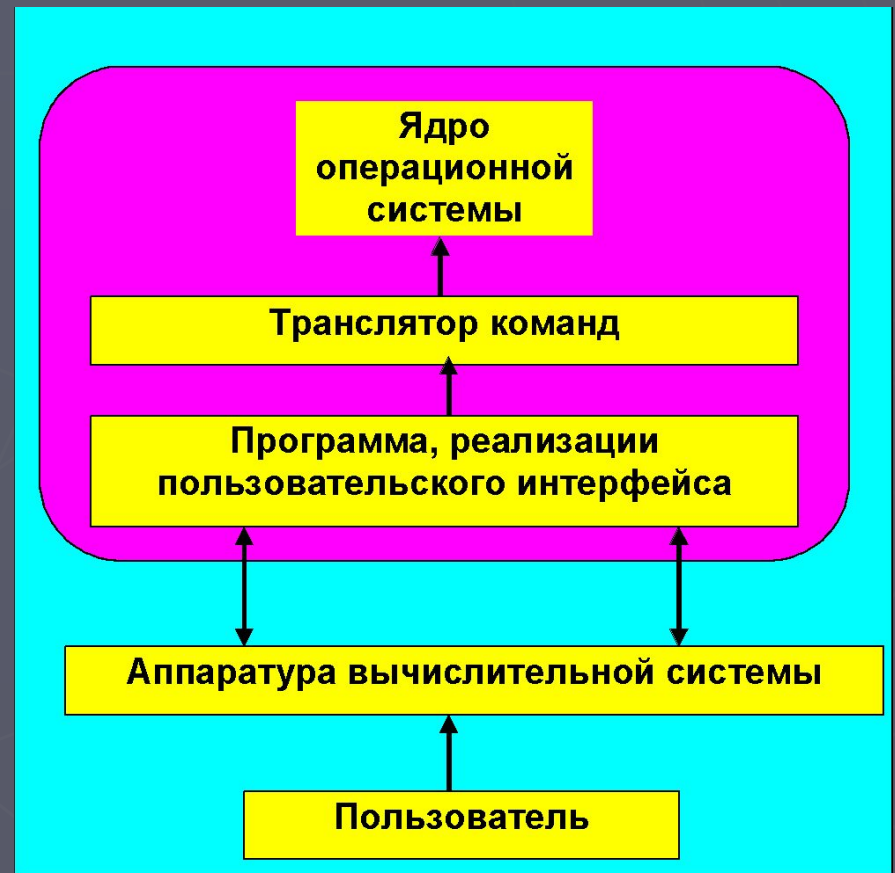
- запрос на управление виртуальными устройствами (напомним, что управление вводом-выводом является привилегированной функцией самой операционной системы, и никакая из пользовательских задач не должна иметь возможности непосредственно управлять устройствами);
- файловые операции (запросы к системе управления файлами на создание, изменение и удаление данных, организованных в файлы).

Виды интерфейсов



Пользовательские интерфейсы

Интерфейс пользователя с операционной системой реализуется с помощью специальных программных модулей, которые принимают его команды на соответствующем языке (возможно, с использованием графического интерфейса) и транслируют их в обычные вызовы в соответствии с основным интерфейсом системы.



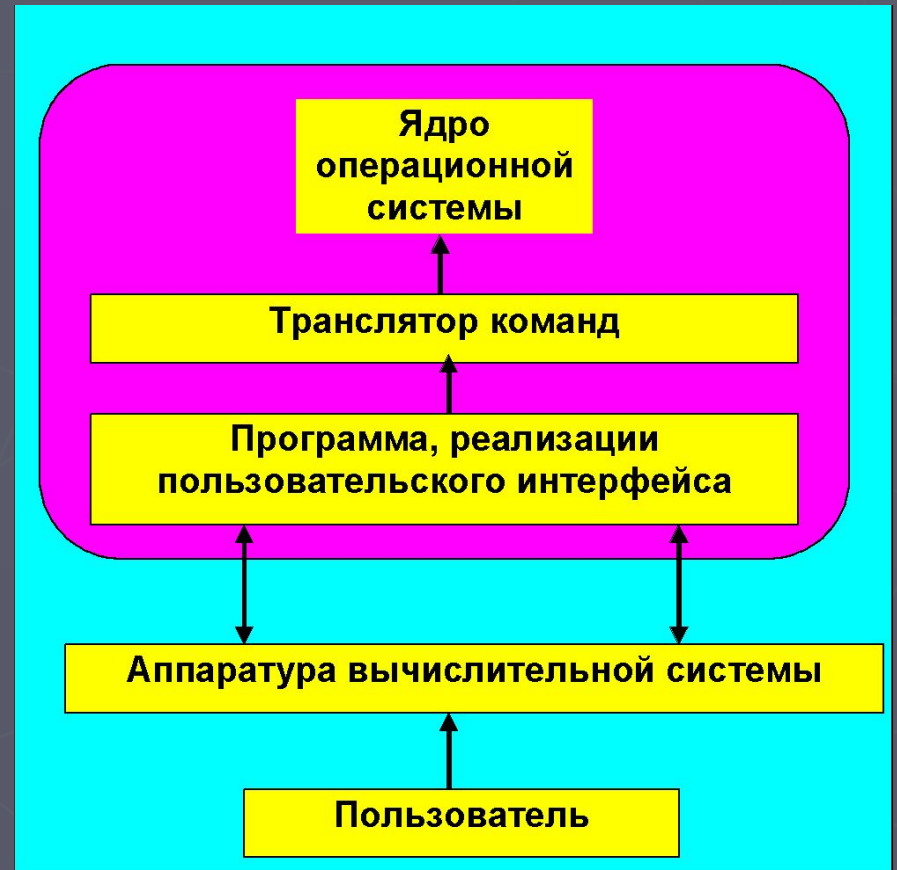
Примеры команд

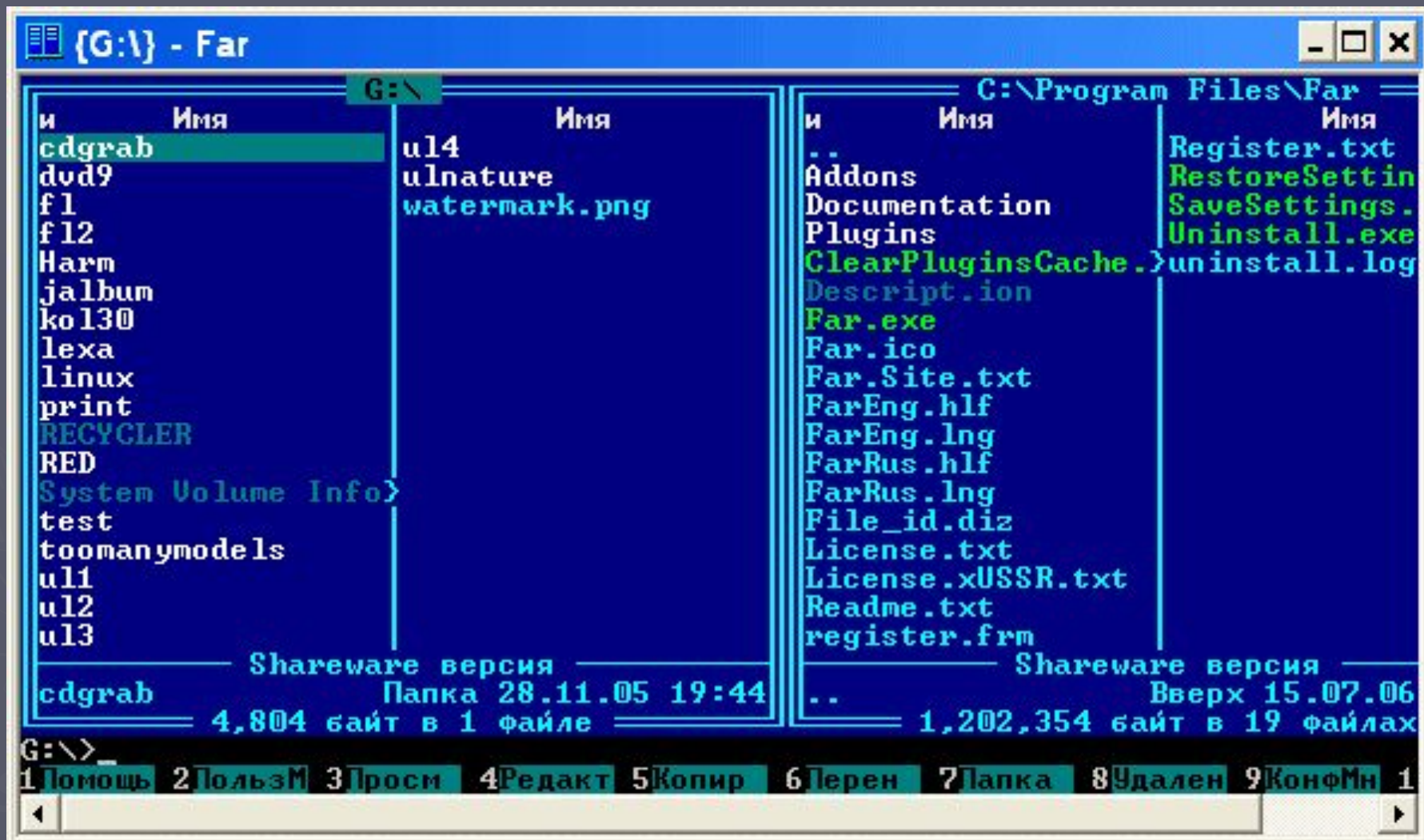
C:\Work\program.exe

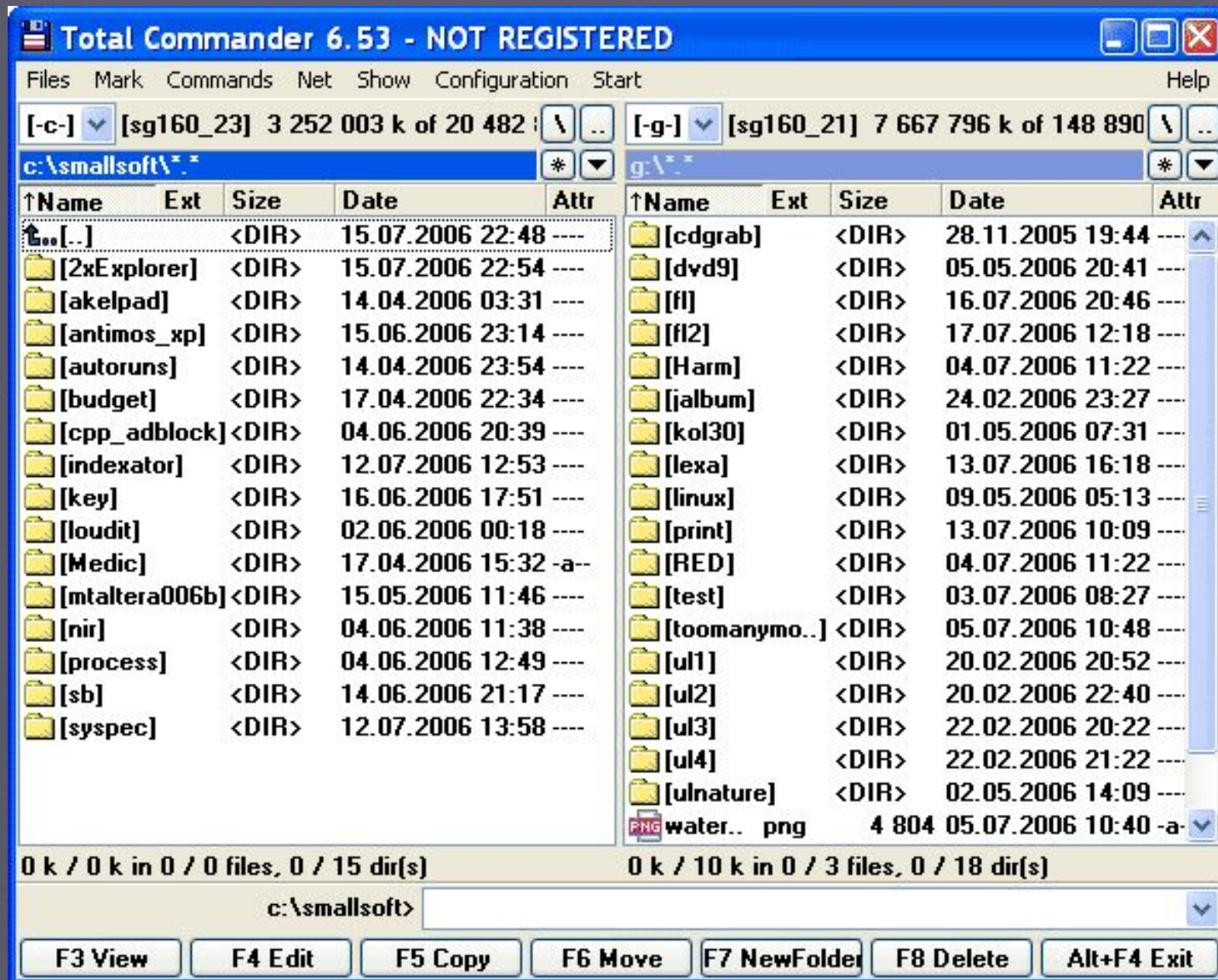
Copy C:st.doc D:\K1\c1.*

Del C:\Work\lb2.xls

Rename st.doc stat.*







Application Program Interface - интерфейс прикладного программирования

Необходимо однозначно разделить общий термин API на следующие направления:

- API как интерфейс высокого уровня, принадлежащий к библиотекам RTL;
- API прикладных и системных программ, входящих в поставку операционной системы;
- прочие интерфейсы API.

Интерфейс прикладного программирования, как это и следует из названия, предназначен для использования прикладными программами системных ресурсов компьютера и реализуемых операционной системой разнообразных системных функций.

API описывает совокупность функций и процедур, принадлежащих ядру или надстройкам операционной системы.

Функции API позволяют разработчику строить результирующую прикладную программу так, чтобы использовать средства целевой вычислительной системы для выполнения типовых операций. При этом разработчик программы избавлен от необходимости создавать исходный код для выполнения этих операций.

Варианты реализации API

- реализация на уровне модулей операционной системы;
- реализация на уровне системы программирования;
- реализация на уровне внешней библиотеки процедур и функций.

Система программирования в каждом из этих вариантов предоставляет разработчику средства для подключения функций API к исходному коду программы и организации их вызовов.

Возможности API можно оценивать со следующих позиций:

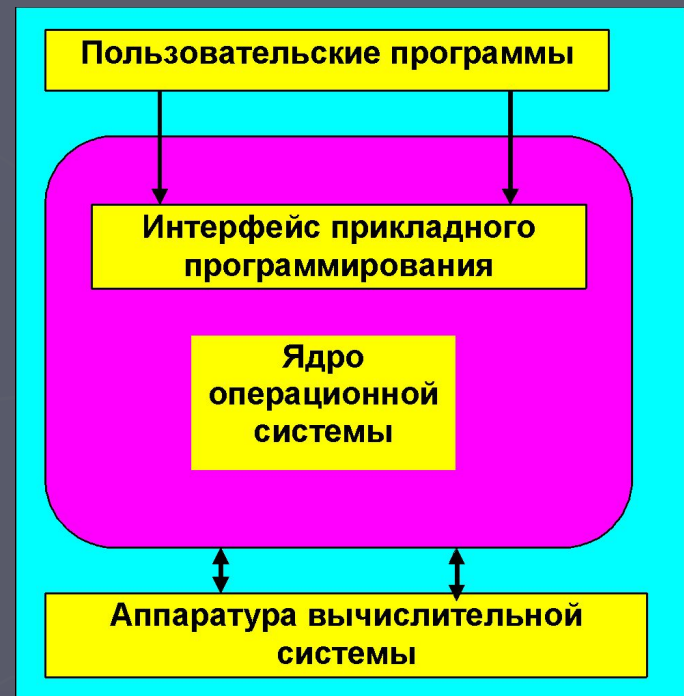
- эффективности выполнения функций API (эффективность включает в себя скорость выполнения функций и объем вычислительных ресурсов, необходимых для их выполнения);
- широты предоставляемых возможностей;
- зависимости прикладной программы от архитектуры целевой вычислительной системы.

Реализация функций API на уровне модулей операционной системы

При реализации функций API на уровне модулей операционной системы операционная система ответственна за выполнение функций API.

Объектный код, выполняющий функции, либо непосредственно входит в состав операционной системы (или даже ядра операционной системы), либо находится в составе динамически загружаемых библиотек, поставляемых вместе с системой.

Система программирования ответственна только за то, чтобы организовать интерфейс для вызова этого кода.



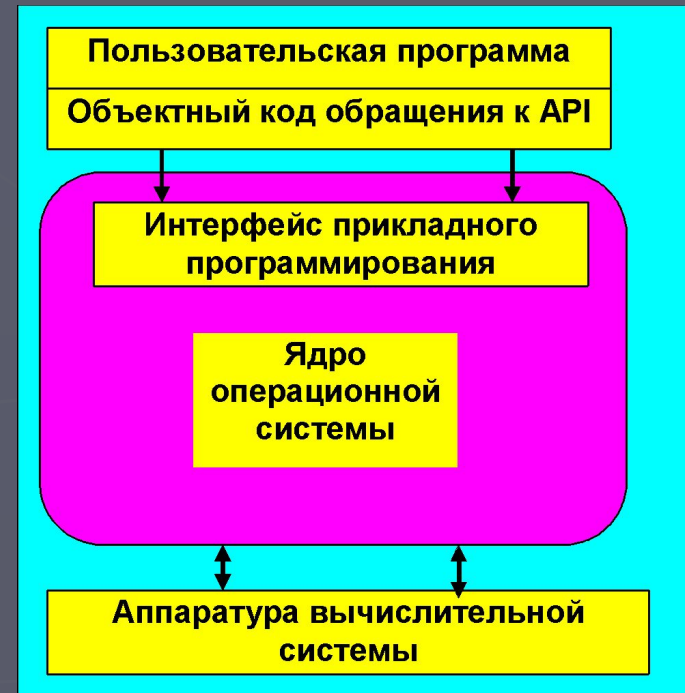
Недостатком организации API по такой схеме является практически полное отсутствие переносимости не только кода результирующей программы, но и кода исходной программы. Программа, созданная для одной архитектуры вычислительной системы, не сможет исполняться на вычислительной системе другой архитектуры даже после того, как ее объектный код полностью перестроен. Переносимости можно было бы добиться, если унифицировать функции API в различных операционных системах.

Реализация функций API на уровне системы программирования

При реализации функций API на уровне системы программирования эти функции предоставляются пользователю в виде библиотеки функций соответствующего языка программирования.

Система программирования предоставляет пользователю библиотеку функций и обеспечивает подключение к результирующей программе объектного кода, ответственного за выполнение этих функций.

Эффективность вызова функций API в таком варианте будет несколько ниже, чем при непосредственном обращении к функциям операционной системы.



Переносимость исходного кода программы в таком варианте оказывается самой высокой, поскольку синтаксис и семантика всех функций строго регламентированы в стандарте соответствующего языка программирования. Единообразное выполнение функций языка обеспечивается системой программирования. При ориентации на различные архитектуры целевой вычислительной системы в системе программирования могут потребоваться различные комбинации вызовов функций операционной системы для выполнения одних и тех же функций исходного языка.

Проблема главным образом заключается в том, что большинство языков программирования предоставляют пользователю не очень широкий набор стандартизованных функций. Поэтому разработчик исходного кода существенно ограничен в выборе доступных функций API.

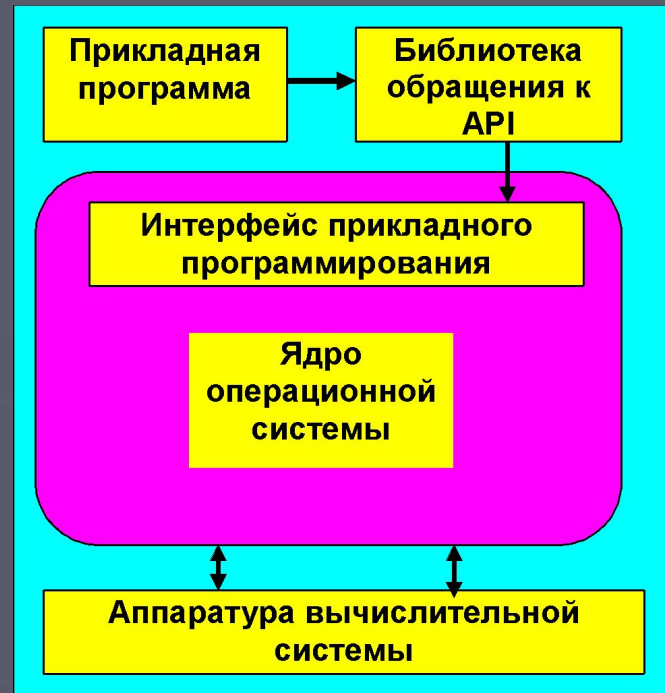
Как правило, набора стандартных функций оказывается недостаточно для создания полноценной прикладной программы.

Реализация функций API с помощью внешних библиотек

При реализации функций API на уровне системы программирования эти функции предоставляются пользователю в виде библиотеки функций соответствующего языка программирования.

Система программирования предоставляет пользователю библиотеку функций и обеспечивает подключение к результирующей программе объектного кода, ответственного за выполнение этих функций.

Эффективность вызова функций API в таком варианте будет несколько ниже, чем при непосредственном обращении к функциям операционной системы.



С точки зрения эффективности выполнения этот метод реализации API имеет самые низкие результаты, поскольку внешняя библиотека обращается как к функциям операционной системы, так и к функциям языка программирования.

Только при очень высоком качестве внешней библиотеки ее эффективность сравнима с эффективностью предыдущего подхода.

Если говорить о переносимости исходного кода, то здесь требование только одно - используемая внешняя библиотека должна быть доступна в любой из архитектур вычислительных систем, на которые ориентирована прикладная программа. Тогда удастся достигнуть переносимости.

Заключение

В целом развитие функций API идет в направлении попытки создать библиотеки API, обеспечивающие широкую переносимость исходного кода.

С учетом корпоративных интересов различных производителей и сложившейся ситуации на рынке в ближайшее время вряд ли удастся достичь значительных успехов в этом направлении. Разработка широко применимого стандарта API пока еще остается делом будущего.

Поэтому разработчики системных программ вынуждены оставаться в довольно узких рамках ограничений стандартных библиотек языков программирования.