**Типовая структура операционной системы.**

Структура ОС  
Обычно в составе ОС выделяют два уровня: ядро системы и вспомогательные системные программные средства, иногда называемые системными утилитами.

Ядро выполняет все функции по управлению ресурсами системы – как физическими, так и логическими – и разделяет доступ пользователей (программ пользователей) к этим ресурсам.

При помощи системного программного обеспечения пользователь управляет средствами, предоставляемыми ядром.

В ядро типичной операционной системы входят следующие компоненты: система управления сеансами пользователей, файловая система, система управления задачами(процессами), система ввода/вывода

Интерфейс ядра ОС с прикладными программами осуществляется при помощи программного интерфейса системных вызовов, интерфейс с аппаратным обеспечением – при помощи драйвера.

Система управления сеансами пользователей осуществляет регистрацию сеанса пользователя при начале его работы с ОС, хранит оперативную информацию, входящую в информационное окружение сеанса, при помощи системы ввода/вывода поддерживает соответствие пользовательского терминала реальными или виртуальными устройствам, корректно завершает сеанс, при окончании работы пользователя с системой.  
  
**Структура ядра ОС**

Прикладное ПО

Аппаратное обеспечение

Драйвера

Система ввода/вывода

Система управления процессами

Файловая система

Система управления сеансами пользователей

Интерфейс системных вызовов

**Модули ядра Linux**

Пояснение к схеме:

* Memory Manager
* Virtual File System
* Process Scheduler
* Inter-Process Communication
* Network Interface

**Файловая система ОС**

Файловая система выполняет преобразование данных, хранимых на внешних запоминающих устройствах (например, на дисковых накопителях или на Flash-накопителях), в логические объекты – файлы и каталоги.

Также файловая система выполняет функции разграничения доступа к файлам и каталогам при обращении к ним со стороны системы управления сеансами или при использовании файловой системы через интерфейс системных вызовов

**Система управления процессами**

Система управления процессами распределяет ресурсы между выполняемыми задачами (процессами), обеспечивает защиту памяти процессов от модификации её другими процессами, реализует механизм меж процессного взаимодействия

**Система ввода–вывода**

Система ввода/вывода обрабатывает запросы всех рассмотренных выше компонент ядра и преобразует их в вызовы логических устройств, поддерживаемых ОС.  
Каждое такое устройство представляет собой логический объект, обращение к которому происходит стандартным для ОС средствами (например как к адресу в оперативной памяти либо как к специальному файлу)

**Драйверы**

Драйверы устройства – это системные программы, которые преобразуют запросы системы ввода/вывода в последовательности управляющей команд для аппаратных устройств.  
Драйвера каждого устройства скрывает особенности его аппаратной реализации и предоставляет системе ввода/вывода стандартизированный интерфейс доступа к аппаратному обеспечению системы.

**Структура ОС**

С точки зрения прикладного программиста доступ к компонентам ядра ОС осуществляется при помощи интерфейса системных вызовов – набора библиотек, включающих в себя стандартизированные наборы функций  
Каждый такой набор предназначен для решения того или иного класса прикладных задач: доступа к сетевым ресурсам, графическому режиму, реализации меж процессного взаимодействия и т.п.

**Интерфейс в ОС**

Понятие интерфейса можно представить весьма широко.

Под интерфейсом можно понимать:

* Некоторые языки, обеспечивающий взаимодействие с которой системой например Shell или Bash, предназначенный для ввода команд с помощью командной строки.
* Способ представления информации (внешний вид программы), например в оконном виде с соответствующей поддержкой ввода команда посредством управления клавиатурой, мышью, джойстиком и др.
* Механизм обмена данными посредством некоторого соединения, например USB, LPT, COM и т.п. К примеру говоря о USB–интерфейсе как способе обмена данными между устройствами.
* Некоторый протокол обмена информацией на основе языка программирования. Например, графический интерфейс языка Java – набор классов языка, обеспечивающий графическое взаимодействие между некоторой системой и пользователем.

**Графический интерфейс пользователя**

Графический интерфейс пользователя (GUI) - разновидность пользовательского интерфейса, в котором элементы управления (меню, кнопки, значки, списки и т.п.) отображаются на экране в виде графических изображений.

Чаще всего элементы интерфейса в GUI реализованы на основе метафор (набор визуальных образов пользовательского интерфейса, к примеру, файл и папка) и при этом отображают их значение и свойства, что облегчает понимание и освоение программ неподготовленным пользователям.

Но графический интерфейс далеко не всегда является лучшим решением