

Операционная система

Запрос «OS» перенаправляется сюда; см. также [другие значения](#).

[[править](#) | [править код](#)]



В статье есть список [источников](#), но в этом разделе **не хватает сносок**.

Без сносок сложно [определить, из какого источника](#) взято каждое отдельное утверждение. Вы можете улучшить статью, проставив [сноски на источники](#), подтверждающие информацию. Сведения без сносок [могут быть удалены](#).

Операцио́нная систе́ма, сокр. **ОС** ([англ.](#) *operating system*, OS) — комплекс взаимосвязанных программ, предназначенных для управления ресурсами [компьютера](#) и организации взаимодействия с пользователем.

В логической структуре типичной [вычислительной системы](#) операционная система занимает положение между [устройствами](#) с их микроархитектурой, [машинным языком](#) и, возможно, [собственными \(встроенными\) микропрограммами \(драйверами\)](#) — с одной стороны — и [прикладными программами](#) с другой.

Разработчикам [программного обеспечения](#) операционная система позволяет абстрагироваться от деталей реализации и функционирования устройств, предоставляя минимально необходимый набор функций (см.: [интерфейс программирования приложений](#)).

В большинстве вычислительных систем операционная система является основной, наиболее важной (а иногда и единственной) частью [системного программного обеспечения](#). С 1990-х годов наиболее распространёнными операционными системами являются системы семейства [Windows](#), [Unix](#) и [UNIX-подобные системы](#).

Содержание [\[скрыть\]](#)

- История
 - Пакетный режим
 - Разделение времени и многозадачность
 - Разделение полномочий
 - Масштаб реального времени
 - Файловые системы и структуры
- Функции
- Понятие
- Ядро
- Существующие операционные системы
- UNIX, стандартизация операционных систем и POSIX
- Пост-UNIX-архитектуры
- См. также
- Примечания
- Литература
- Ссылки

История [\[править | править код \]](#)

Основная статья: [История операционных систем](#)

Предшественником операционных систем следует считать служебные программы (загрузчики и мониторы), а также библиотеки часто используемых [подпрограмм](#), начавшие разрабатываться с появлением универсальных [компьютеров 1-го поколения](#) (конец [1940-х годов](#)). Служебные программы минимизировали физические манипуляции оператора с оборудованием, а библиотеки позволяли избежать многократного программирования одних и тех же действий (осуществления операций [ввода-вывода](#), вычисления математических функций и т. п.).

В [1950—1960-х годах](#) сформировались и были реализованы основные идеи, определяющие функциональность ОС: [пакетный режим](#), разделение времени и многозадачность, разделение полномочий, реальный масштаб времени, файловые структуры и [файловые системы](#).

Пакетный режим [\[править | править код \]](#)

Необходимость оптимального использования дорогостоящих вычислительных ресурсов привела к появлению концепции «пакетного режима» исполнения программ. Пакетный режим предполагает наличие очереди программ на исполнение, причём система может обеспечивать загрузку программы с внешних носителей данных в оперативную память, не дожидаясь завершения исполнения предыдущей программы, что позволяет избежать простоя процессора.

Разделение времени и многозадачность [\[править | править код \]](#)

Основная статья: [Разделение времени](#)

Уже пакетный режим в своём развитом варианте требует разделения процессорного времени между выполнением нескольких программ.

Необходимость в разделении времени (многозадачности, мультипрограммировании) проявилась ещё сильнее при распространении в качестве устройств ввода-вывода [телетайпов](#) (а позднее, терминалов с электронно-лучевыми дисплеями) ([1960-е годы](#)). Поскольку скорость клавиатурного ввода (и даже чтения с экрана) данных оператором намного ниже, чем скорость обработки этих данных компьютером, использование компьютера в «монопольном» режиме (с одним оператором) могло привести к простоя дорогостоящих вычислительных ресурсов.

Разделение времени позволило создать «многопользовательские» системы, в которых один (как правило) [центральный процессор](#) и блок оперативной памяти соединялся с многочисленными терминалами. При этом часть задач (таких как ввод или редактирование данных оператором) могла исполняться в режиме диалога, а другие задачи (такие как массивные вычисления) — в пакетном режиме.

Разделение полномочий [\[править | править код \]](#)

Распространение многопользовательских систем потребовало решения задачи разделения полномочий, позволяющей избежать возможности изменения исполняемой программы или данных одной программы в памяти компьютера другой программой (намеренно или по ошибке), а также изменения самой системы [прикладной программой](#).

Реализация разделения полномочий в операционных системах была поддержана разработчиками процессоров, предложивших архитектуры с двумя режимами работы процессора — «реальным» (в котором исполняемой программе доступно всё [адресное пространство](#) компьютера) и «защищённым» (в котором доступность адресного пространства ограничена диапазоном, выделенным при запуске программы на исполнение).

Масштаб реального времени [\[править | править код \]](#)

Применение универсальных компьютеров для управления производственными процессами потребовало реализации «масштаба реального времени» («реального времени») — синхронизации исполнения программ с внешними физическими процессами.

Включение функции масштаба реального времени позволило создавать решения, одновременно обслуживающие производственные процессы и решающие другие задачи (в пакетном режиме и/или в режиме разделения времени).

Файловые системы и структуры [\[править | править код \]](#)

Постепенная замена носителей с последовательным доступом ([перфолент](#), [перфокарт](#) и [магнитных лент](#)) накопителями произвольного доступа (на [магнитных дисках](#)).

Файловая система — способ хранения данных на внешних запоминающих устройствах.

Функции [\[править | править код \]](#)

Основные функции:

- Исполнение запросов программ (ввод и вывод данных, запуск и остановка других программ, выделение и освобождение дополнительной памяти и др.).
- [Загрузка программ](#) в оперативную память и их выполнение.
- Стандартизированный доступ к периферийным устройствам ([устройства ввода-вывода](#)).
- Управление оперативной памятью (распределение между процессами, организация [виртуальной памяти](#)).
- Управление доступом к данным на энергонезависимых носителях (таких как [жёсткий диск](#), [оптические диски](#) и др.), организованным в той или иной [файловой системе](#).
- Обеспечение [пользовательского интерфейса](#).
- Сохранение информации об ошибках системы.

Дополнительные функции:

- Параллельное или псевдопараллельное выполнение задач ([многозадачность](#)).
- Эффективное распределение ресурсов вычислительной системы между [процессами](#).
- Разграничение доступа различных процессов к ресурсам.
- Организация надёжных вычислений (невозможности одного вычислительного процесса намеренно или по ошибке повлиять на вычисления в другом процессе), основана на разграничении доступа к ресурсам.
- [Взаимодействие между процессами](#): обмен данными, взаимная синхронизация.
- Защита самой системы, а также пользовательских данных и программ от действий пользователей (злонамеренных или по незнанию) или приложений.
- Многопользовательский режим работы и разграничение прав доступа (см.: [аутентификация](#), [авторизация](#)).



[OS/360](#) использовалась на большинстве компьютеров [IBM](#) начиная с 1966, включая те компьютеры, которые помогали [NASA](#) отправить человека на Луну.

Понятие [[править](#) | [править код](#)]

Существуют две группы определений операционной системы: «набор программ, управляющих оборудованием» и «набор программ, управляющих другими программами». Обе они имеют свой точный технический смысл, который связан с вопросом, в каких случаях требуется операционная система.

Есть приложения вычислительной техники, для которых операционные системы излишни. Например, встроенные [микрокомпьютеры](#), содержащиеся во многих бытовых приборах, автомобилях (иногда по десятку в каждом), простейших сотовых телефонах, постоянно исполняют лишь одну программу, запускающуюся по включении. Многие простые игровые приставки — также представляющие собой специализированные микрокомпьютеры — могут обходиться без операционной системы, запуская при включении программу, записанную на вставленном в устройство «картридже» или [компакт-диске](#).

Операционные системы нужны:

- если нужен универсальный механизм сохранения данных;
- для предоставления программам [системных библиотек](#) с часто используемыми подпрограммами;
- для распределения полномочий;
- необходима возможность имитации «одновременного» исполнения нескольких программ на одном компьютере;
- для управления процессами выполнения отдельных программ.

Таким образом, современные универсальные операционные системы можно охарактеризовать, прежде всего, как:

- использующие файловые системы (с универсальным механизмом доступа к данным),
- многопользовательские (с разделением полномочий),
- многозадачные (с разделением времени).

Многозадачность и распределение полномочий требуют определённой иерархии привилегий компонентов в самой операционной системе. В составе операционной системы различают три группы компонентов:

- [ядро](#), содержащее планировщик; драйверы устройств, непосредственно управляющие оборудованием; сетевая подсистема, файловая система;
- [системные библиотеки](#);
- [оболочка](#) с утилитами.

Большинство программ, как системных (входящих в операционную систему), так и прикладных, исполняются в непривилегированном («пользовательском») режиме работы [процессора](#) и получают доступ к оборудованию (и, при необходимости, к другим ресурсам ядра, а также ресурсам иных программ) только посредством [системных вызовов](#). Ядро исполняется в привилегированном режиме: именно в этом смысле говорят, что система (точнее, её ядро) управляет оборудованием.

В определении состава операционной системы значение имеет критерий операционной целостности (замкнутости): система должна позволять полноценно использовать (включая модификацию) свои компоненты. Поэтому в полный состав операционной системы включают и набор инструментальных средств (от текстовых редакторов до компиляторов, отладчиков и компоновщиков).

Ядро [[править](#) | [править код](#)]

Основная статья: [Ядро операционной системы](#)

Ядро — центральная часть операционной системы, управляющая выполнением [процессов](#), ресурсами [вычислительной системы](#) и предоставляющая процессам координированный доступ к этим ресурсам. Основными ресурсами являются [процессорное время](#), [память](#) и [устройства ввода-вывода](#). Доступ к [файловой системе](#) и сетевое взаимодействие также могут быть реализованы на уровне ядра.

Как основополагающий элемент операционной системы, ядро представляет собой наиболее низкий уровень абстракции для доступа приложений к ресурсам вычислительной системы, необходимым для их работы. Как правило, ядро предоставляет такой доступ исполняемым процессам соответствующих приложений за счёт использования механизмов [межпроцессного взаимодействия](#) и обращения приложений к системным вызовам ОС.

Описанная задача может различаться в зависимости от типа архитектуры ядра и способа её реализации.

Объекты ядра ОС:

- [процессы](#),
- [файлы](#),
- события,
- [потoki](#),
- [семафоры](#),
- [мьютексы](#),
- [каналы](#),
- [файлы, проецируемые в память](#).

Существующие операционные системы [\[править | править код \]](#)

Основная статья: [Список операционных систем](#)

UNIX, стандартизация операционных систем и POSIX [\[править | править код \]](#)

К концу [1960-х годов](#) отраслью и научно-образовательным сообществом был создан целый ряд операционных систем, реализующих все или часть очерченных выше функций. К ним относятся [Atlas](#) ([Манчестерский университет](#)), [CTTS](#) ^{рус.}_(англ.) и [ITS](#) ^{рус.}_(англ.) ([Массачусетский технологический институт](#), MIT), [THE](#) ([Эйндховенский технологический университет](#)), RS4000 ([Университет Орхуса](#)) и др. (всего эксплуатировалось более сотни различных ОС).

Наиболее развитые операционные системы, такие как [OS/360](#) ([IBM](#)), [SCOPE](#) ([CDC](#)) и завершённый уже в [1970-х годах Multics](#) (MIT и [Bell Labs](#)), предусматривали возможность исполнения на многопроцессорных компьютерах.

Эклектичный характер разработки операционных систем привёл к нарастанию кризисных явлений, прежде всего, связанных с чрезмерными сложностью и размерами создаваемых систем. Системы были плохо масштабируемыми (более простые не могли использовать все возможности крупных вычислительных систем; более развитые неоптимально исполнялись на малых или не могли исполняться на них вовсе) и полностью несовместимыми между собой, их разработка и совершенствование затягивались.

Задуманная и реализованная в [1969 году Кеном Томпсоном](#) при участии нескольких коллег (включая [Денниса Ритчи](#) и [Брайана Кернигана](#)), операционная система [UNIX](#) (первоначально UNICS, что обыгрывало название Multics) вобрала в себя многие черты более ранних систем, но обладала целым рядом свойств, отличающих её от большинства предшественниц:

- простая метафорика (два ключевых понятия: [вычислительный процесс](#) и файл);
- компонентная архитектура: принцип «одна программа — одна функция» плюс мощные средства связывания различных программ для решения возникающих задач («оболочка»);
- минимизация ядра (кода, выполняющегося в «реальном» (привилегированном) режиме процессора) и количества системных вызовов;
- независимость от аппаратной архитектуры и реализация на машиннонезависимом [языке программирования](#) (язык программирования [Си](#) стал побочным продуктом разработки UNIX);
- унификация файлов.

UNIX, благодаря своему удобству прежде всего в качестве инструментальной среды (среды разработки), обрела популярность сначала в университетах, а затем и в отрасли, получившей прототип единой операционной системы, которая могла использоваться на самых разных вычислительных системах и, более того, могла быть быстро и с минимальными усилиями перенесена на любую вновь разработанную аппаратную архитектуру.

В конце 1970-х годов сотрудники [Калифорнийского университета в Беркли](#) внесли ряд усовершенствований в исходные коды UNIX, включая работу с протоколами [TCP/IP](#). Их разработка стала известна под именем [BSD](#) (Berkeley Software Distribution).

Задачу разработать независимую (от авторских прав Bell Labs) реализацию той же архитектуры поставил и [Ричард Столлман](#), основатель [проекта GNU](#).

Благодаря конкурентности реализаций архитектура UNIX стала вначале фактическим отраслевым стандартом, а затем обрела статус и стандарта юридического — [ISO/IEC 9945^{\[1\]}](#) (POSIX).

Только системы, отвечающие спецификации [Single UNIX Specification](#), имеют право носить имя UNIX. К таким системам относятся [AIX](#), [HP-UX](#), [IRIX](#), [Mac OS X](#), [SCO OpenServer](#), [Solaris](#), [Tru64](#) и [z/OS](#).

Операционные системы, следующие стандарту [POSIX](#) или опирающиеся на него, называют «POSIX-совместимыми» (чаще встречается словупотребление «[UNIX-подобные](#)» или «семейство UNIX», но оно противоречит статусу торгового знака «UNIX», принадлежащего консорциуму [The Open Group](#) и зарезервированному для обозначения только операционных систем, строго следующих стандарту). Сертификация на совместимость со стандартом платная, из-за чего некоторые системы не проходили этот процесс, однако считаются POSIX-совместимыми по существу.

К UNIX-подобным относятся операционные системы, основанные на последней версии UNIX, выпущенной Bell Labs ([System V](#)), на разработках университета Беркли ([FreeBSD](#), [OpenBSD](#), [NetBSD](#)), на основе Solaris ([OpenSolaris](#), [BeleniX](#), [Nexenta OS](#)), а также [Linux](#), разработанная в части утилит и библиотек [проектом GNU](#) и в части ядра — сообществом, возглавляемым [Линусом Торвальдсом](#).

Стандартизация операционных систем преследует цель упрощения замены самой системы или оборудования при развитии вычислительной системы или сети и упрощении переноса прикладного программного обеспечения (строгое следование стандарту предполагает полную совместимость программ на уровне исходного текста; из-за профилирования стандарта и его развития некоторые изменения бывают всё же необходимы, но перенос программы между POSIX-совместимыми системами обходится на порядки дешевле, чем между альтернативными), а также преемственность опыта пользователей.

Самым заметным эффектом существования этого стандарта стало эффективное разворачивание [Интернета](#) в [1990-х годах](#).

Пост-UNIX-архитектуры [\[править | править код \]](#)



Возможно, этот раздел содержит **оригинальное исследование**.

Добавьте [ссылки на источники](#), в противном случае этот раздел может быть удалён. Соответствующую дискуссию можно найти на [странице обсуждения](#). (14 мая 2020)

Коллектив, создавший UNIX, развил концепцию унификации объектов операционной системы, включив в исходную концепцию UNIX «устройство — это тоже файл» также и процессы, и любые другие системные, сетевые и прикладные сервисы, создав новую концепцию: «что угодно — это файл». Эта концепция стала одним из основных принципов системы [Plan 9](#) (название было позаимствовано из фантастического триллера «[План 9 из открытого космоса](#)» [Эдварда Вуда-младшего](#)), призванной преодолеть принципиальные недостатки дизайна UNIX и сменившей «рабочую лошадку» UNIX System V на компьютерах сети Bell Labs в [1992 году](#).

Кроме реализации всех объектов системы в виде файлов и размещения их на едином и персональном для каждого терминала вычислительной сети пространстве (namespace), были пересмотрены другие архитектурные решения UNIX. Например, в Plan 9 отсутствует понятие «суперпользователь», и, соответственно, исключаются любые нарушения режима безопасности, связанные с нелегальным получением прав суперпользователя в системе. Для представления (хранения, обмена) информации [Роб Пайк](#) и Кен Томпсон разработали универсальную кодировку [UTF-8](#), на сегодняшний день ставшую стандартом де-факто. Для доступа к файлам используется единый универсальный протокол 9P, по сети работающий поверх сетевого протокола (TCP или UDP). Таким образом, для прикладного ПО сети не существует — доступ к локальным и к удалённым файлам единообразен. 9P — байт-ориентированный протокол, в отличие от других подобных протоколов, являющихся блок-ориентированными. Это также результат работы концепции: доступ побайтно — к унифицированным файлам, а не поблочно — к разнообразным и сильно изменяющимся с развитием технологий устройствам. Для контроля доступа к объектам не требуется иных решений, кроме уже существующего в операционной системе контроля доступа к файлам. Новая концепция системы хранения избавила администратора системы от изнурительного труда по сопровождению архивов и предвосхитила современные системы управления версиями файлов.

Операционные системы, созданные на базе или идеях UNIX, такие как всё семейство BSD и системы Linux, постепенно перенимают новые идеи из Bell Labs. Возможно, эти новые идеи ждёт большое будущее и признание ИТ-разработчиков.

Новые концепции были использованы Робом Пайком в [Inferno](#).

На основе Plan 9 в [Испании](#) разрабатываются системы Off++ и [Plan B](#), носящие экспериментальный характер.

К попыткам создать пост-UNIX-архитектуру можно также отнести разработку [языка программирования и операционной среды](#) Оберон в [Швейцарской высшей технической школе](#) (ETH Zurich) под руководством профессора [Никлауса Вирта](#).

См. также [\[править | править код \]](#)

- [Аппаратная платформа компьютера](#)
- [Сетевая операционная система](#)

Примечания [\[править | править код \]](#)

- ↑ Свежая версия ISO/IEC 9945 принята [Международной организацией по стандартизации](#) (ISO) в [2003 году](#).

Литература [[править](#) | [править код](#)]

- Гордеев А. В. Операционные системы: Учебник для вузов. — 2-е изд. — СПб.: Питер, 2007. — 416 с. — ISBN 978-5-94723-632-3.
- Деннинг П. Дж., Браун Р. Л. Операционные системы // Современный компьютер. — М., 1986.
- Иртегов Д. В. Введение в операционные системы. — 2-е изд. — СПб.: BHV-СПб, 2007. — ISBN 978-5-94157-695-1.
- Керниган Б. У., Пайк Р. У. UNIX — универсальная среда программирования = The UNIX Programming Environment. — М., 1992.
- Олифер В. Г., Олифер Н. А. Сетевые операционные системы. — СПб.: Питер, 2002. — 544 с. — ISBN 5-272-00120-6.
- Столлингс У. Операционные системы = Operating Systems: Internals and Design Principles. — М.: Вильямс, 2004. — 848 с. — ISBN 0-1303-1999-6.
- Таненбаум Э. С. Многоуровневая организация ЭВМ = Structured Computer Organization. — М.: Мир, 1979. — 547 с.
- Таненбаум Э. С. Современные операционные системы = Modern Operating Systems. — 2-е изд. — СПб.: Питер, 2005. — 1038 с. — ISBN 5-318-00299-4.
- Таненбаум Э. С., Вудхалл А. С. Операционные системы. Разработка и реализация = Operating Systems: Design and Implementation. — 3-е изд. — СПб.: Питер, 2007. — 704 с. — ISBN 978-5-469-01403-4.
- Шоу А. Логическое проектирование операционных систем = The Logical Design of Operating Systems. — М.: Мир, 1981. — 360 с.
- Рэймонд Э. С. Искусство программирования для UNIX = The Art of UNIX Programming. — М.: Вильямс, 2005. — 544 с. — ISBN 5-8459-0791-8.
- Mark G. Sobell. UNIX System V. A Practical Guide. — 3rd ed. — 1995.

Ссылки [[править](#) | [править код](#)]

- [Выбор операционной системы для сервера](#)
- [Операционная система](#) в каталоге ссылок Curlie (dmoz)
- [Обзор различных операционных систем](#).
- [Отставнов М. Е. Свободное программное обеспечение в школе](#) (недоступная ссылка). *Свободное ПО для школы* (2003). Дата обращения: 16 апреля 2010. [Архивировано](#) 6 июля 2008 года.
- [Операционная система \(Operating system\) по ГОСТ 15971-90](#)

Категории: [Операционная система](#) | [Изобретения США](#)