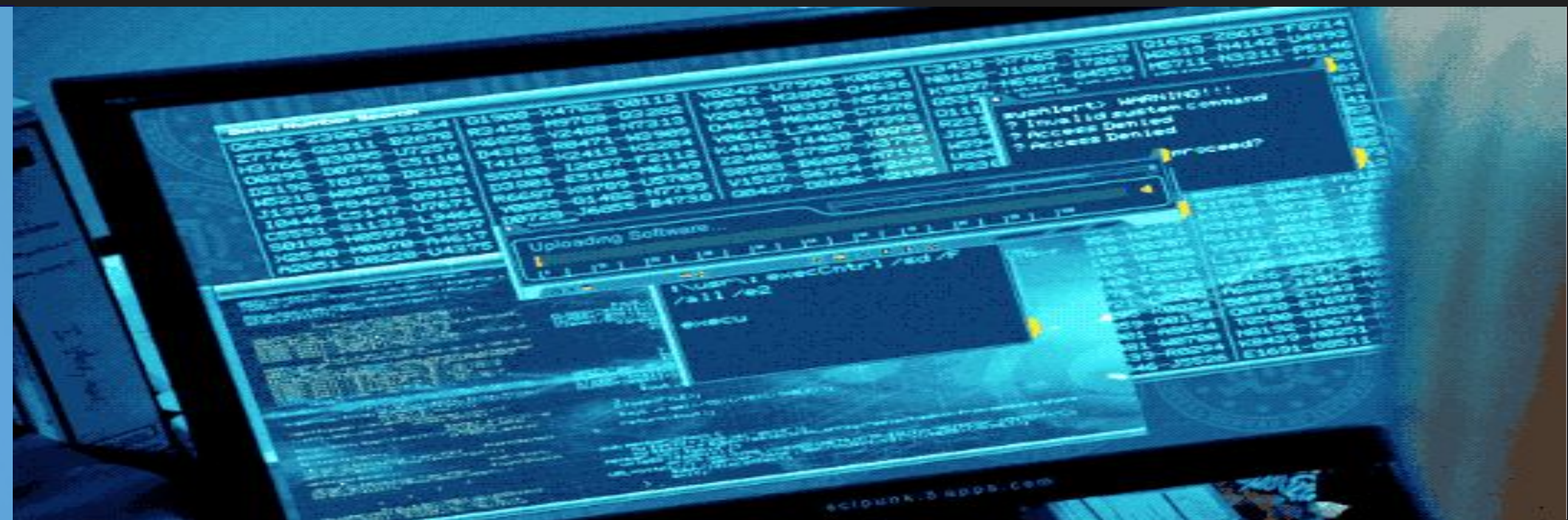


# Системное и прикладное программное обеспечение

## Программное обеспечение

Совокупность программ, предназначенная для решения задач на ПК, называется программным обеспечением. Состав программного обеспечения ПК называют программной конфигурацией.



Подготовил:

Студент группы 23-КС

Соловей Денис

# Содержание

## План

1	Введение
2	Обзор операционных систем
3	Системные программы
4	Прикладные программы
5	Операционная система как расширенная машина
6	История операционных систем
7	Процессоры
8	Процессы
9	Безопасность
10	Литература



# Введение

## Операционные системы

Современная компьютерная система состоит из:

- одного или нескольких процессов,
  - оперативной памяти,
  - дисков,
  - клавиатуры,
  - монитора,
  - принтеров,
  - сетевого интерфейса
- и других устройств, то есть является сложной комплексной системой.

Написание программ, которые следят за всеми компонентами, корректно используют их и при этом работают оптимально, представляет собой крайне трудную задачу.

По этой причине компьютеры оснащаются специальным уровнем программного обеспечения, называемым операционной системой.

Операционная система отвечает за управление всеми перечисленными устройствами и обеспечивает пользователя имеющимися простой, доступный интерфейс программами для работы с аппаратурой.

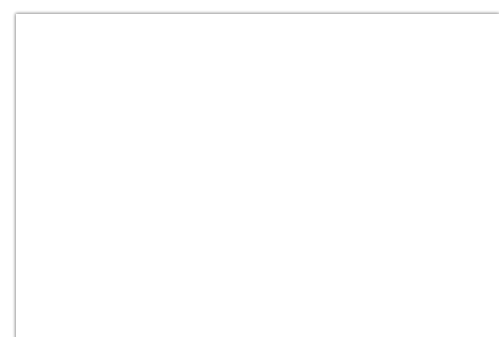


# Операционная Система

**Операционная система** предназначена для того, чтобы скрыть от пользователя сложности, связанные с управлением устройствами.

Она состоит из уровня программного обеспечения, который частично избавляет от необходимости общения с аппаратурой напрямую, вместо этого предоставляя программисту более удобную систему команд.

Например: чтения блока из файла.





# Системные программы

Существуют системы, в которых понятие ядра размыто: встроенные системы;

интерпретируемые системы (Java-машина).

В традиционных компьютерах операционная система представляет собой набор программ, запускающихся в режиме ядра.

Во многих системах есть программы, которые работают в пользовательском режиме, но выполняют специализированные функции.

Например:  
смена паролей.



# Прикладные Программы

Над системными программами расположены прикладные программы.

Обычно они покупаются или пишутся пользователем для решения собственных проблем:

- обработки текста;
- электронных таблиц;
- технических расчетов;
- сохранения информации в базе данных.



## Приложения разделяют на два класса:

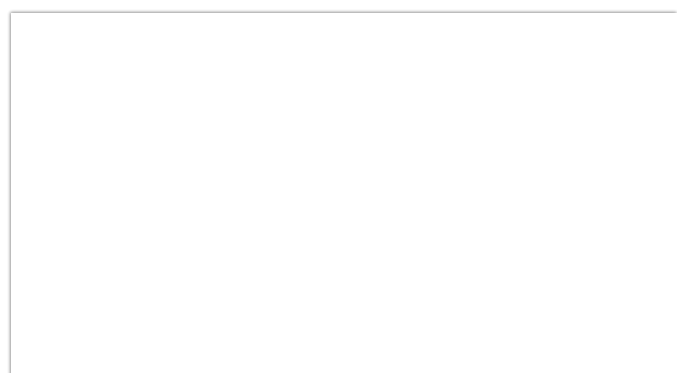
Это системные утилиты и прикладное программное обеспечение. В основном используются для обработки поступающей информации с какого-нибудь оборудования.

# Операционная система как расширенная машина

Архитектура (система команд, организация памяти, ввод-вывод данных и структура шин) большинства компьютеров на уровне машинного языка примитивна и неудобна для работы с программами, особенно в процессе ввода-вывода данных.

Например:

Ввод-вывод данных с гибкого диска через совместимые микросхемы контроллера NEC PD765. Контроллер PD765 имеет 16 команд: чтение и запись данных, перемещение головки диска и форматирование дорожек, а также для инициализация, распознавание, установка в исходное положение и калибровка контроллера и приводов.



# Операционная система

## как расширенная машина

Программа, скрывающая истину об аппаратном обеспечении, является операционной системой. Операционная система не только устраняет необходимость работы непосредственно с дисками и предоставляет простой интерфейс, но и скрывает множество неприятной работы с прерываниями, счетчиками времени, организацией памяти и другими элементами низкого уровня.

В любом случае абстракция, предлагаемая операционной системой, намного проще и удобнее в обращении, чем то, что может предложить нам непосредственно основное оборудование.

С точки зрения пользователя операционная система выполняет функцию расширенной машины или виртуальной машины, в которой проще программировать и легче работать, чем непосредственно с аппаратным обеспечением, составляющим реальный компьютер.

Операционная система предоставляет нам ряд возможностей, которые могут использовать программы с помощью специальных команд, называемых системными вызовами.



# История Операционных систем

**История развития операционных систем** насчитывает уже много лет. Так как операционные системы появились и развивались в процессе конструирования компьютеров, то эти события исторически тесно связаны. Поэтому чтобы представить, как выглядели операционные системы, мы обсудим следующие друг за другом поколения компьютеров.

Первый настоящий цифровой компьютер был изобретен английским математиком *Чарльзом Бэббиджем (Charles Babbage, 1792-1871)*.



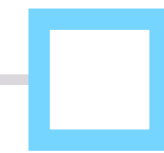
**1945-1955**

*Первое поколение  
электронные лампы и  
коммутационные панели.*



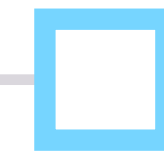
**1955-65**

*Второе поколение  
транзисторы и системы  
пакетной обработки*



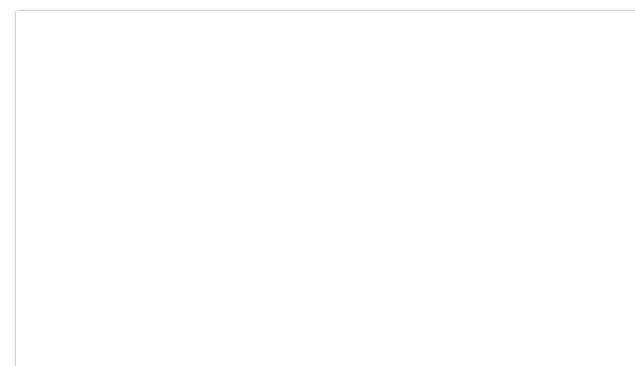
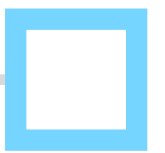
**1965-1980**

*Третье поколение  
интегральные схемы и  
многозадачность*



**2020**

*Четвертое поколение (с  
1980 года по наши дни):  
персональные  
компьютеры  
Онтогенез повторяет  
филогенез*



# Процессоры PC - BRAIN

«Мозгом» компьютера является центральный процессор (CPU - Central Processing Unit).

Для каждого центрального процессора существует набор команд, который он в состоянии выполнить. Например: процессор Pentium не может обработать программы, написанные для SPARC, и наоборот.

Цикл работы центрального процессора выглядит так:

**#1**

*Он читает первую команду из памяти, декодирует ее для определения ее типа и операндов,*

---

**#2**

*выполняет команду*

---

**#3**

*затем считывает, декодирует и выполняет последующие команды.*



# Процессы

*Рассмотрим системы, работающие в режиме разделения времени.*

Предположим, периодически операционная система решает остановить работу одного процесса и запустить другой, потому что первый израсходовал отведенную для него часть рабочего времени центрального процессора в прошедшую секунду.

Если процесс был приостановлен подобным образом, позже он должен быть запущен заново из того же состояния, в каком его остановили. Следовательно, всю информацию о процессе нужно где-либо явно сохранить на время его приостановки. Вся информация о каждом процессе, хранится в таблице процессов.

Пример:

Процесс, называемый интерпретатором команд, читает команды с терминала. Пользователь напечатал команду, содержащую запрос на компиляцию программы. Теперь оболочка должна создать новый процесс, который запустит компилятор. Когда процесс закончит компиляцию, он выполнит системный вызов, завершающий его собственную работу.

## Ключевое понятие операционной системы - процесс

Ключевое понятие операционной системы - процесс.

Процессом, по существу, называют программу в момент выполнения.

С каждым процессом связывается его адресное пространство - список адресов в памяти, которые процесс может прочесть и в которые он может писать.



# Безопасность

Компьютеры содержат большое количество информации, конфиденциальность которой пользователи зачастую хотят сохранить: электронную почту, бизнес-планы и многое другое. В задачу операционной системы входит управление системой защиты подобных файлов.

В качестве простейшего примера, дающего представление о том, как работает система безопасности, рассмотрим систему UNIX. В UNIX для защиты файлов им присваивается 9-битовый двоичный код.

Кроме защиты файлов, существует еще множество других вопросов безопасности: защита системы от нежелательных гостей, людей, и не только (вирусов).



# Литература

**1** Таненбаум Э.С. Современные операционные системы. 3-е изд. СПб.:Питер, 2010. - 1120с.

---

**2** Олифер В.Г., Олифер Н.А. Сетевые операционные системы. 2-е изд. СПб.:Питер, 2009. - 672с.

---

**3** Иванова Г.С. Технология программирования: Учебник для ВУЗов. изд.:КноРус, 2011 – 336с.

---

**4** Соммервилл И. Инженерия программного обеспечения, 6-е издание, 2002.

---

**5** Таненбаум Э.С., Вудхалл А.С. Операционные системы: Разработка и реализации

---

**6** Калянов Г.Н. CASE структурный и системный анализ /М.: Изд-во «ЛОРИ», 1996.

---

**7** Роббинс Дж. Отладка приложений: Пер. с англ. – СПб.; БХВ-Петербург, 2001.

---