****

**ПРЕДМЕТ:**

**“Человеко-компьютерное взаимодействие”**

**ПРАКТИЧЕСКАЯ РАБОТА**

**Выполнил: студент Абдурахмонов Самандар**

**группы 655-20**

**«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_\_ г.**

**Принял(-а): преподаватель Порубай О.В.**

**Оценка: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_\_ г.**

**Фергана 2022**

****

**Практическая работа № 1.**

Согласно своему варианту выполнить задание практической работы:

**Задание 1.** Определить тематику для разработки пользовательского интерфейса.

|  |  |
| --- | --- |
|  | Аэропорт (Ж/д станция). Пассажирское расписание и перевозки |

**Задание 2.** Подготовить презентацию (см. вариант)

|  |  |
| --- | --- |
| *Вариант* ***3****.* | Определить основные поколения интерфейсов. Распечатать отчет и защитить теоретическую часть. |

**Контрольные вопросы:**

1. Классификация интерфейсов.

В вычислительной технике понятие *интерфейс (interface)*определяется как граница раздела двух систем, устройств или программ; а так же совокупность средств сопряжения и связи, обеспечивающая эффективное взаимодействие систем или их частей. Интерфейс систем предусматривает вопросы сопряжения на физическом (число проводов, элементы связи, типы соединений, разъемы, номера контактов и т. п.) и логическом (понятные сигналы, их длительности, полярности, частоты и амплитуда, протоколы взаимодействия) уровнях. Интерфейсы ЭВМ классифицируются по различным признакам, один из возможных способов классификации приведен

1. По способу передачи информации интерфейсы подразделяются на параллельные и последовательные.

В *параллельном*интерфейсе все биты передаваемого слова (обычно байта) выставляются и передаются по соответствующим параллельно идущим проводам одновременно. В ПК традиционно используется параллельный интерфейс *Centronics,*реализуемый LPT-портами.

В *последовательном*интерфейсе биты передаются друг за другом, обычно по одной линии. СОМ-порты ПК обеспечивают последовательный интерфейс в соответствии со стандартом *RS-232C.*

2. По виду или по месту расположения связываемых устройств интерфейсы подразделяются на внутримашинные и внешние.



*Внутримашинный интерфейс —*система связи и сопряжения узлов и блоков компьютера между собой. Представляет собой совокупность электрических линий связи (проводов), схем сопряжения с компонентами компьютера, протоколов (алгоритмов) передачи и преобразования сигналов.

Существует два варианта организации внутримашинного интерфейса:

– *многосвязный интерфейс:*каждый блок ПК связан с прочими блоками своими локальными проводами; многосвязный интерфейс иногда применяется в качестве периферийного интерфейса (для связи с внешними устройствами ПК), дополняющего системный, а в качестве системного — лишь в некоторых простых компьютерах;

– *односвязный интерфейс:*все блоки ПК связаны друг с другом через общую или системную шину).

В подавляющем большинстве современных ПК в качестве системного интерфейса используется *системная*шина. *Шина*(bus) — совокупность линий связи, по которым информация передается одновременно. Под основной, или *системной,*шиной обычно понимается шина между процессором и подсистемой памяти. В качестве системной шины в разных ПК использовались и могут использоваться:

– *шины расширений —*шины общего назначения, позволяющие подключать большое число самых разнообразных устройств;

– *локальные шины,*часто специализирующиеся на обслуживании небольшого количества устройств определенного класса, преимущественно видеосистем.

3. По режиму передачи данных интерфейсы подразделя­ются на *синхронные*и *асинхронные*.

При *синхронной*передаче, работа приемо – передающих устройств синхронизируется специальными синхроимпульсами предаваемыми ведущим устройством; время между последовательно предаваемыми блоками данных строго определено.

При *асинхронной*передаче работа устройств не синхронизируется. Время между последовательно предаваемыми блоками данных определяется ведущим устройством и может меняться в любых пределах.

4. Для интерфейса, соединяющего (физически или логически) два устройства, различают три возможных режима обмена — *дуплексный, полудуплексный и симплексный.*

*Дуплексный*режим позволяет по одному каналу связи одновременно передавать информацию в обоих направлениях. Он может быть асимметричным, если пропускная способность в направлениях «туда» и «обратно» имеет существенно различающиеся значения, или симметричным.

*Полудуплексный*режим позволяет передавать информацию «туда» и «обратно» поочередно, при этом интерфейс имеет средства переключения направления канала.

*Симплексный*(односторонний) режим предусматривает только одно направление передачи информации (во встречном направлении передаются только вспомогательные сигналы интерфейса).

5. С появлением шин USB и FireWire в качестве классификационной характеристики интерфейса стала фигурировать и *топология соединения.*Для интерфейсов характерны три вида топологии соединений:

*двухточечная топология.*Для интерфейсов *RS-232C*и *Centronics*практически всегда применялась *двухточечная топология*ПК — устройство (или ПК — ПК), которые подключаются к СОМ- или LPT-портам. Аналогично обстоит дело и с адаптерами локальных сетей (например, Paraport) и внешних дисковых накопителей (Iomega Zip), подключаемых к LPT-портам. Хотя стандарты для параллельного порта (IEEE 1284.3) и предусматривают соединение устройств в цепочку *(Daisy Chain)*или через мультиплексоры, широкого распространения такие способы подключения пока не получили.

К другому классу подключений относится построение моноканала на эффективной *шинной технологии Ethernet,*гдевсе устройства подключены к одному каналу связи.

Интерфейсные шины USB иFire Wire реализуют *древовидную топологию,*в которой внешние устройства могут быть как оконечными, так и промежуточными (разветвителями). Эта топология позволяет подключать множество устройств к одному порту USB илиFireWire.

**2. Описать технологии командного интерфейса.**

# Командный интерфейс

В командном интерфейсе человек отдает компьютеру команды в явном виде, компьютер их выполняет и каким-либо образом выводит результат. Командный интерфейс реализован на двух уровнях: пакетная технология и технология командной строки.

*Пакетная технология*. На вход компьютера подается заранее подготовленный пакет данных – последовательность символов, в которых по определенным правилам указывается последовательность запущенных на выполнение команд или программ. После выполнения очередной программы запускается следующая и т.д. В качестве источника пакета данных может выступать, например, перфолента, стопка перфокарт, последовательность нажатия клавиш электрической пишущей машинки. Машина также выдает свои сообщения с результатом на перфоратор, алфавитно-цифровое печатающее устройство (АЦПУ), ленту пишущей машинки.

*Технология командной строки*. В качестве единственного способа ввода информации от человека к компьютеру служит клавиатура, а компьютер выводит информацию с помощью алфавитно-цифрового дисплея (монитора). Эта комбинацию (монитор + клавиатура) получила название «терминал», «консоль».

Технология командной строки работала уже на монохромных алфавитно-цифровых дисплеях. Поскольку вводить позволяется только буквы, цифры и знаки препинания, то технические характеристики дисплея несущественны. В качестве монитора можно использовать телевизионный приемник и даже трубку осциллографа.

Таким образом, обе эти технологии реализуются в виде командного интерфейса – машине подаются на вход команды, а их выполняет. Преобладающим видом файлов при работе с командным интерфейсом являются текстовые файлы, поскольку их достаточно легко создать с помощью клавиатуры. На время наиболее широкого использования интерфейса командной строки приходится появление операционной системы UNIX и первых восьмиразрядных персональных компьютеров.

**3. Каковы этапы развития графического интерфейса? Опишите их.**

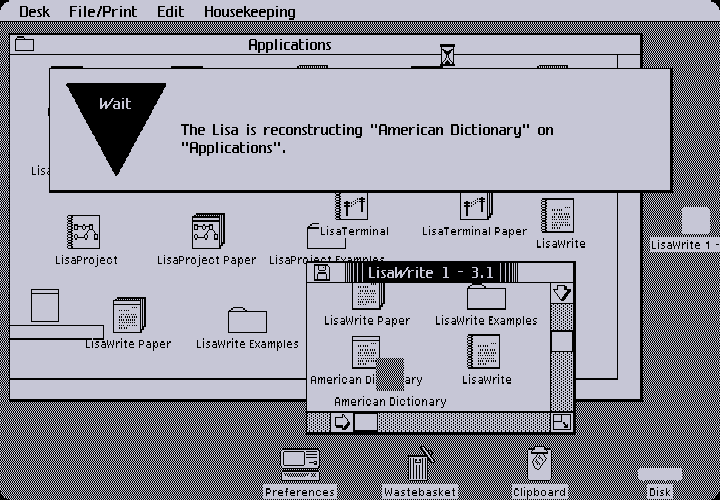
Любое устройство, механическое или электронное, в задачи которого входит прямое взаимодействие с пользователем, помимо своей внутренней начинки должно обладать чем-то, благодаря чему это самое взаимодействие смогло бы осуществляться. Имя этого посредника сегодня известно каждому. Это — интерфейс. Он может быть аналоговым, но обычно под интерфейсом понимают графическую оболочку или иначе ***GUI*** операционных систем и программного обеспечения.

Большинству интерфейсов популярных ныне операционных систем свойственно интуитивно-понятное графическое оформление с использованием визуальных эффектов, однако так было не всегда. С точки зрения современного пользователя первые ***GUI*** были довольно примитивны, хотя, нужно отдать им должное, это не всегда означало отсутствие качественного по тем временам *юзабилити*.

### **Xerox – первый настоящий GUI**

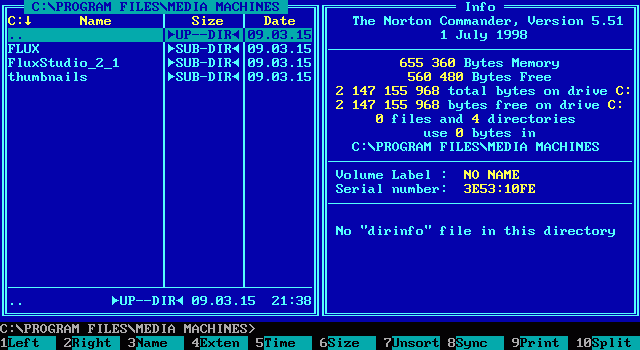
Родоначальником всех ныне существующих графических интерфейсов правильнее считать ***GUI***, разработанный в рамках проекта **Xerox Alto** — первого персонального компьютера, созданного в 1973 году. Оболочка Xerox Alto была очень проста, но уже тогда в ней присутствовали меню, кнопки и примитивные окна. Был в ней и курсор мыши с присущими ему функциями выделения, копирования и вставки.

 Дальше — больше. В 1981 году появляется новая система под названием **Xerox Star**, основанная на той же Xerox Alto, но с более совершенным функционалом и графическим интерфейсом. Возможно, вы будете удивлены, но рабочий стол Xerox Star мало чем отличался от нынешних десктопов, если, конечно, не брать в расчёт визуальные эффекты.

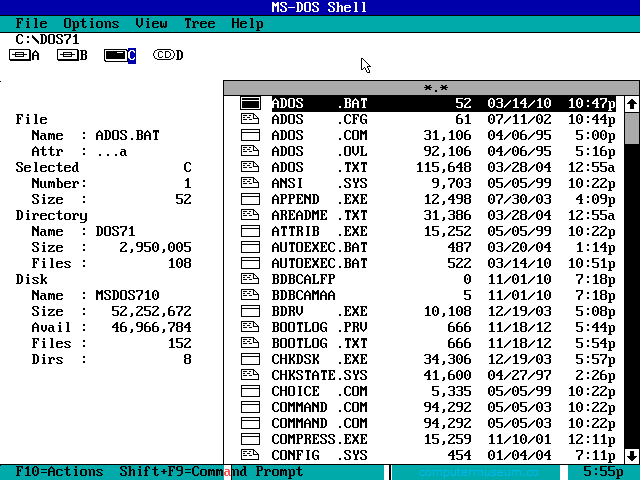
Дальше — больше. В 1981 году появляется новая система под названием **Xerox Star**, основанная на той же Xerox Alto, но с более совершенным функционалом и графическим интерфейсом. Возможно, вы будете удивлены, но рабочий стол Xerox Star мало чем отличался от нынешних десктопов, если, конечно, не брать в расчёт визуальные эффекты. 

### **DOS. Norton Commander как попытка создания оригинального GUI**

Заимствования идей Xerox Lab, однако, вовсе не означают, что никаких попыток создания оригинальных интерфейсов для операционных систем не предпринималось. В 1986 году программистом Джоном Соча был создан **Norton Commander** — файловый менеджер для **MS-DOS**, до этого не имевшей практически никакого графического оформления. Роль окон в нем играли панели, делящие экран по вертикали и содержащие списки папок и файлов. В верхней и нижней части менеджера располагались текстовые меню, позволяющие выполнять те или иные операции.



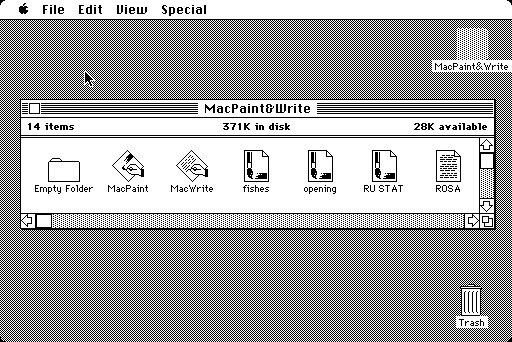
Впрочем, ***GUI*** в полном смысле этого слова *Norton Commander* не являлся. Как и вышедшей в *1988* году его аналог **DOS Shell**, он относится к псевдографическим интерфейсам, имитирующим графику, оставаясь при этом текстовыми.

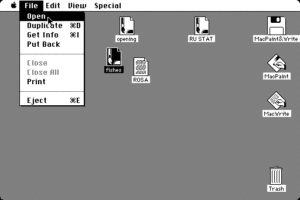
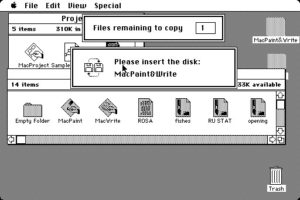
[](https://www.white-windows.ru/wp-content/uploads/2017/07/ms_dos_shell.png)

Тем не менее, оба эти приложения существенно облегчили работу с данными, избавив пользователей от необходимости вводить ***DOS***-команды, чем долгое время и обуславливалась популярность этих программ.

### **MacOS 1.1**

Выйдя из команды разработчиков **Apple Lisa**, в 1982 году Стив Джобс возглавил собственный проект **Macintosh**. Разработанная для маков система получила название **Mac OS**. Внешне она была похожа на Apple Lisa, но в ней имелись также и только ей одной присущие особенности, причём касались они как внешнего вида элементов интерфейса, так и самого взаимодействия пользователя с оболочкой. Как и Apple Lisa, MacOS 1.1 была основана на оконном принципе, в ней использовались меню, иконки и диалоги.

[](https://www.white-windows.ru/wp-content/uploads/2017/07/moremacmacpaintwrite.jpg)

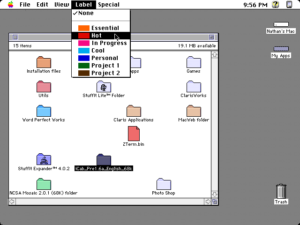
[](https://www.white-windows.ru/wp-content/uploads/2017/07/moremacopenfile.jpg)     [](https://www.white-windows.ru/wp-content/uploads/2017/07/moremacfcopy.jpg)

Оболочка **MacOS 1.1** позволяла быстро переименовывать файлы и папки, выделять их, копировать перетаскиванием в место назначения, одновременно закрывать все окна, хотя закрытие окон не всегда предполагало завершения работы приложения, закрывать программы нужно было правильно — через главное меню системы. При закрытии отредактированных, но не сохранённых файлов появлялось диалоговое окно с запросом на подтверждение сохранения изменений или их отмены.

### **Mac OS 7.5.5**

За семь лет своего существования Mac OS прошла через множество изменений, но почти все они были незначительными и только в седьмой версии появились нововведения, о которых стоило бы упомянуть. Пожалуй, самое главное из них это поддержка цветов, так как до этого интерфейс системы был практически монохромным. Теперь пользователь мог менять цвет иконок папок и некоторых других элементов, делая их синими, желтыми или красными.

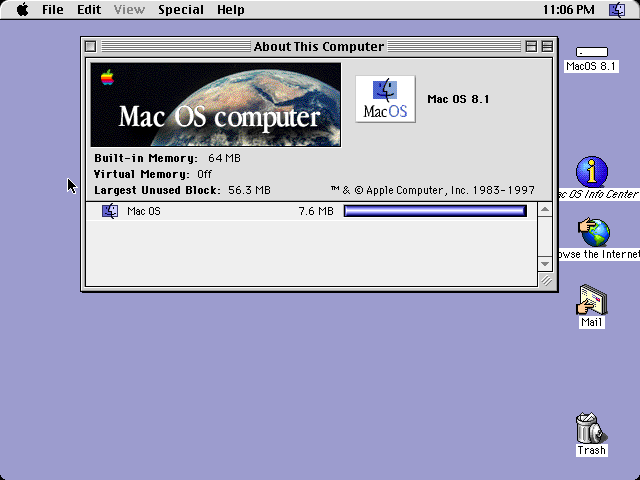
[](https://www.white-windows.ru/wp-content/uploads/2017/07/mac755boot.png)

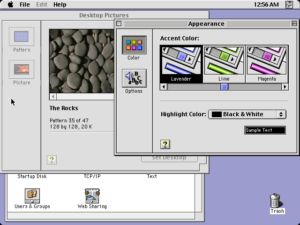
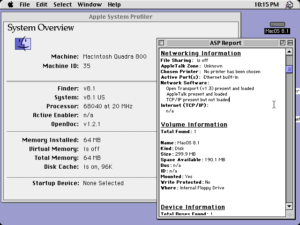
[](https://www.white-windows.ru/wp-content/uploads/2017/07/mac755alias.png)     [](https://www.white-windows.ru/wp-content/uploads/2017/07/mac755label.png)

Присутствовали в цветовой гамме Mac OS 7.5.5 и другие оттенки. В это же время становится цветным «яблочный» логотип в левой части главного меню. Из прочих изменений можно отметить показ иконок модулей во время загрузки системы, расширение функционала меню, добавление всплывающих подсказок при наведении на доступные в меню опции, а также реализация доступа к приложениям из единой панели управления.

### **Mac OS 8.1**

Работа над использованием цвета в графическом интерфейсе была активно продолжена в восьмой версии системы. Системные иконки в **Mac OS 8.1** были цветными по умолчанию, а в самой ОС появилось новое приложение **Appearance Manager**, позволяющее управлять цветовыми схемами. MacOS 8.1 обзавелась набором фоновых изображений, кроме того, в качестве фонов пользователь мог устанавливать произвольные картинки.

[](https://www.white-windows.ru/wp-content/uploads/2017/07/macos81about.png)

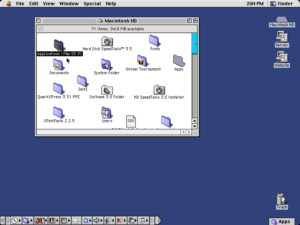
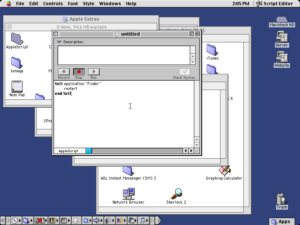
[](https://www.white-windows.ru/wp-content/uploads/2017/07/macos81appear.png)     [](https://www.white-windows.ru/wp-content/uploads/2017/07/macos81asp.png)

В этой же редакции впервые появляется знаменитая платиново-серая тема, ставшая впоследствии визитной карточкой всех последующих версий Mac OS. Другим интересным изменением стало применение к иконкам изометрии, благодаря чему они стали походить на трехмерные объекты, не являясь таковыми на самом деле. Были улучшены настройки отображения содержимого файловой системы — файлы стало можно просматривать в виде списков и значков, размер которых также можно было изменять.

### **Mac OS 9.2.2**

Версией **9.2.2** завершается история Mac OS на основе оригинальной операционной системы Macintosh и казалось, что в ней должно быть больше нововведений, чем в прошлых версиях. В девятой версии действительно много изменений, но коснулись они по большей части функционала, интерфейс же изменился незначительно.

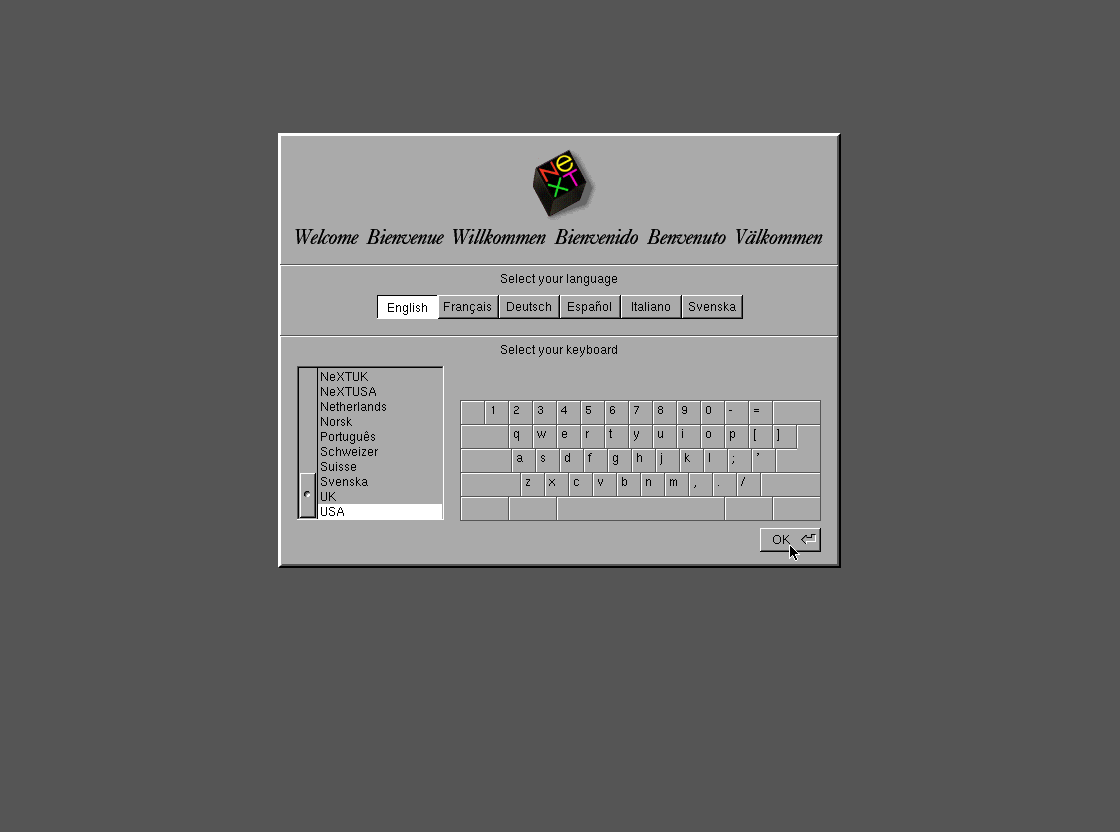
[](https://www.white-windows.ru/wp-content/uploads/2017/07/macos9about.png)

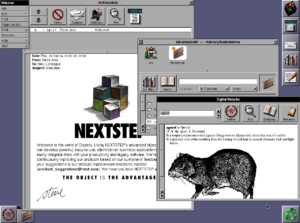
[](https://www.white-windows.ru/wp-content/uploads/2017/07/macos9folders.png)     [](https://www.white-windows.ru/wp-content/uploads/2017/07/macos9script.png)

Из наиболее значимых модификаций, затронувших графическую оболочку, стала интегрированная поддержка нескольких учётных записей. При старте системы на экране появлялось окно выбора профиля, а каждый пользователь мог устанавливать свои темы оформления, причём сторонние графические пакеты тоже поддерживались. В этой же версии также была улучшена панель управления звуком и добавлена возможность установки голосового пароля.

### **NeXTSTEP 3.3**

Покинув Apple, в 1985 году Стив Джобс основал собственную компанию **NeXT Computers**, разработчиками которой была создана ОС ***NeXTSTEP***. Основой новой системы стало ядро **Mach**, а идеи графического оформления позаимствованы у Mac OS и более ранней Apple Lisa.

[](https://www.white-windows.ru/wp-content/uploads/2017/07/ns33welcome.png)

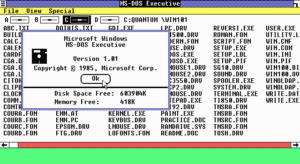
[](https://www.white-windows.ru/wp-content/uploads/2017/07/ns33about.png)     [](https://www.white-windows.ru/wp-content/uploads/2017/07/ns33mail.png)

Подобно Mac OS, первоначально NeXTSTEP имела упрощенный монохромный интерфейс, основанный на оконном принципе с использованием меню, иконок и диалоговых окон. Полная поддержка цветов появилась только в версии ***3.3***, ставшей последней. От поздних версий Mac OS система NeXTSTEP 3.3 отличалась более чистым и лаконичным оформлением.

### **Microsoft Windows 1.0**

Не сидели сложа руки и в Microsoft. В 1985 году компания представляет свою графическую оболочку для MS-DOS с говорящим названием **Windows**. Оболочка частично поддерживала цветную графику, в ней имелись 32x32-пиксельные иконки, простые меню и диалоги. Фиксированной области, в которой бы отображались значки запущенных приложений пока не было, располагаться они могли в любом месте экрана, перекрываясь при этом открытыми окнами.

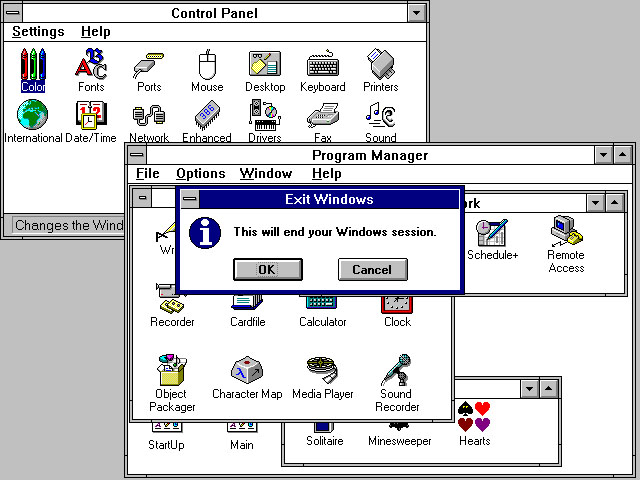
[](https://www.white-windows.ru/wp-content/uploads/2017/07/win101logo.png)

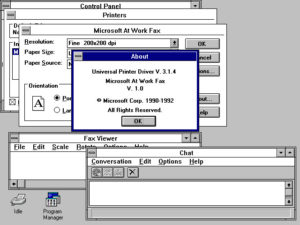
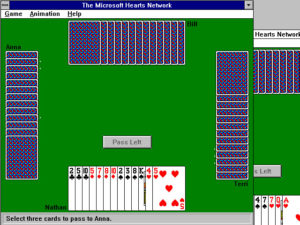
[](https://www.white-windows.ru/wp-content/uploads/2017/07/win101executive.png)     [](https://www.white-windows.ru/wp-content/uploads/2017/07/win101write.png)

Сами окна в первой версии были довольно примитивными. Их можно было перетаскивать мышкой, изменять их размер, но при этом сами они не могли перекрывать друг друга. Сворачивать их также было нельзя. Интерфейс Windows 1.0 облегчал работу с системой и файлами, избавив пользователя от необходимости вводить команды в консоли, но в то же время ему недоставало удобства. Так что в плане юзабилити первая версия Windows значительно уступала системам от Apple.

### **Microsoft Windows 3.11**

Версии Windows 1.0, 2.0 и 3.0 не были операционными системами в том смысле слова, в котором его принято понимать сегодня. Это были скорее графические оболочки MS-DOS, первые признаки, выделяющие Windows в отдельную ОС, появились только с выходом версий 3.1 и 3.11, но относятся они не столько к **GUI**, сколько к функционалу. В плане графического оформления существенных изменений было не так уже и много.

[](https://www.white-windows.ru/wp-content/uploads/2017/07/win311exit.jpg)

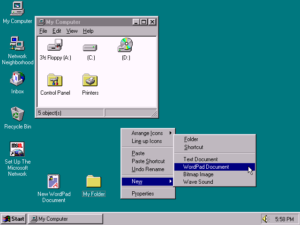
[](https://www.white-windows.ru/wp-content/uploads/2017/07/win311fax.jpg)     [](https://www.white-windows.ru/wp-content/uploads/2017/07/win311hearts.jpg)

В **Windows 3.11** уже имеется полная поддержка цветов, окна могут перекрывать друг друга, их можно сворачивать и разворачивать. Незначительно улучшается графика отдельных элементов (объемные кнопки и полосы прокрутки), используются пропорциональные шрифты, внешний вид программ **File Manager** и **Program Manager** реализуется в стиле самой оболочки. Цвета элементов интерфейса пользователь может менять по своему усмотрению.

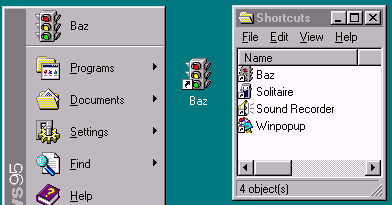
### **Microsoft Windows 95**

Настоящая революция в оформлении Windows свершилась в 1995 году, именно тогда в системе появляются хорошо ныне всем знакомые кнопка ***Пуск***, ***Проводник***, ***Панель задач*** и рабочий стол со значками, который в тоже время являлся отдельной папкой. В этой же версии был реализован показ дисков в папке **«Мой компьютер»** и способ управления файлами из меню, вызываемого правой кнопкой мыши. Немаловажным нововведением стал переход на 32-битную архитектуру.

[](https://www.white-windows.ru/wp-content/uploads/2017/07/win95boot.png)

[](https://www.white-windows.ru/wp-content/uploads/2017/07/win95desktop.png)     [](https://www.white-windows.ru/wp-content/uploads/2017/07/win95desktop2.png)

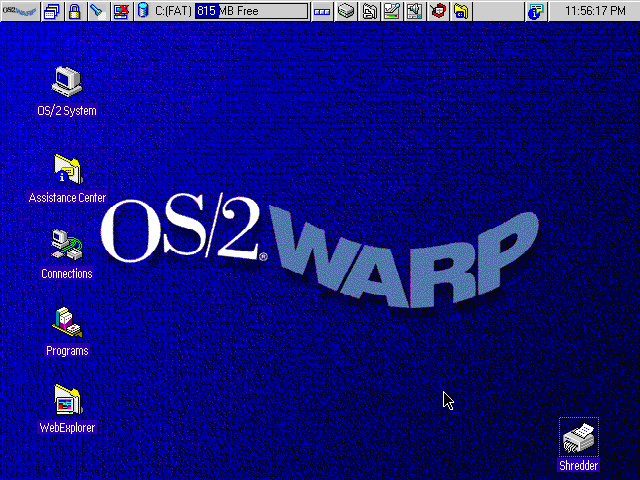
Интересна также история **Windows 95** и та роль, которую она сыграла в крахе проекта **OS/2** — операционной системы совместного детища Microsoft и IBM. На момент выхода Windows 3.0 между компаниями возникли разногласия. Microsoft стремилась продвигать Windows, а IBM ставило приоритетом разработку OS/2. В итоге между компаниями был заключен договор, согласно которому IBM должна была заниматься OS/2 2.0 и Windows 3.0, а Microsoft — OS/2 3.0.

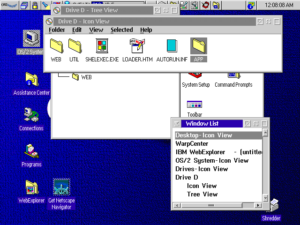
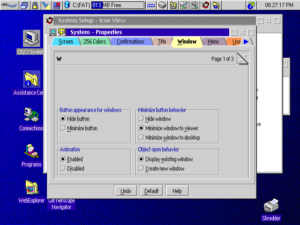
[](https://www.white-windows.ru/wp-content/uploads/2017/07/win95start3.png)

Однако глава Microsoft решил поступить по-своему, объявив OS/2 2.0 устаревшей, а более новую OS/2 3.0 переименовав в Windows NT. В это же время Microsoft выпускает Windows 3.1, а затем и обновление для версии 3.1 под кодовым названием **Chicago**, положенное в основу будущей Windows 95. После этого пути IBM и Microsoft разошлись окончательно. Некоторое время IBM ещё занималась разработкой OS/2, но выход более конкурентоспособной Windows 95 окончательно добил её, и IBM вынуждена была свернуть проект.

### **IBM OS/2 Warp 4**

Поскольку мы затронули тему конфликта между IBM и Microsoft, было бы несправедливо обойти вниманием **OS/2**, плод некогда плодотворного сотрудничества обеих компаний. После ссоры и окончательного разрыва с Microsoft, разработчики IBM продолжили работу над OS/2. В вышедшей в 1996 году версии **OS/2 Warp 4** произведены значительные улучшения внешнего оформления рабочего стола и графической оболочки в целом. На рабочем столе имелись иконки, обеспечивающие доступ к разным системным настройкам, но они же могли играть роль каталогов, в которые пользователь мог помещать файлы и папки, присутствовала Windows-подобная панель задач и единое меню доступа ко всем программам.

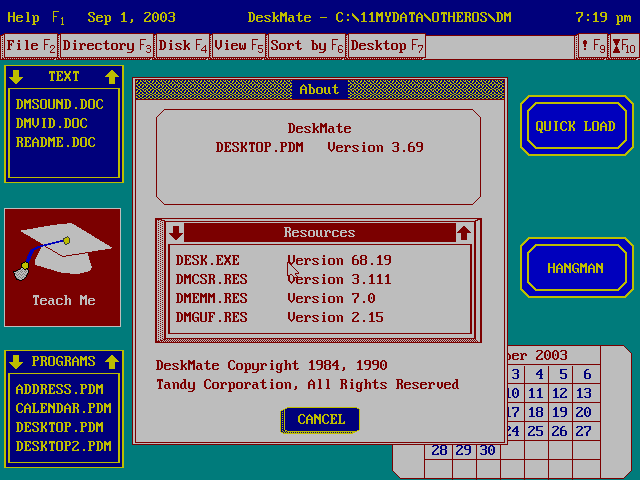
[](https://www.white-windows.ru/wp-content/uploads/2017/07/os24_1stscreen.png)

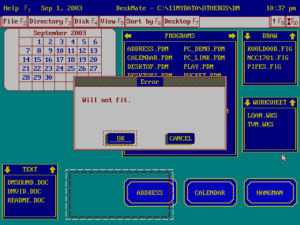
[](https://www.white-windows.ru/wp-content/uploads/2017/07/os24_winlist.png)     [](https://www.white-windows.ru/wp-content/uploads/2017/07/os24_minimize1.png)

**OS/2 Warp 4** имеет много общего с Windows, но есть в ней и весьма существенные отличия. Перетаскивание по умолчанию в OS/2 Warp 4 осуществлялось не левой, а правой кнопкой мыши, **Корзина** служила не для временного хранения удаленных файлов, а для немедленного их уничтожения. Диалоговые окна в этой системе не имели кнопок **«Сохранить»** или **«ОК»**, данные сохранялись автоматически при закрытии окна, если только пользователь не отменял действие. Другой интересной особенностью системы была возможность просмотра древовидной структуры диска в окнах.

### **DeskMate 3.05 (десктопная версия 3.69)**

Выход Windows в 1985 году не ослабил интерес к **MS DOS**, которая по-прежнему пользовалась популярностью, равно как и созданные под неё графические оболочки, среди которых выпущенная фирмой ***Tandy*** среда **DeskMate** заслуживает особого внимания. DeskMate — это не просто графическая надстройка, это полноценная операционная среда, включающая в себя собственный набор программного обеспечения.

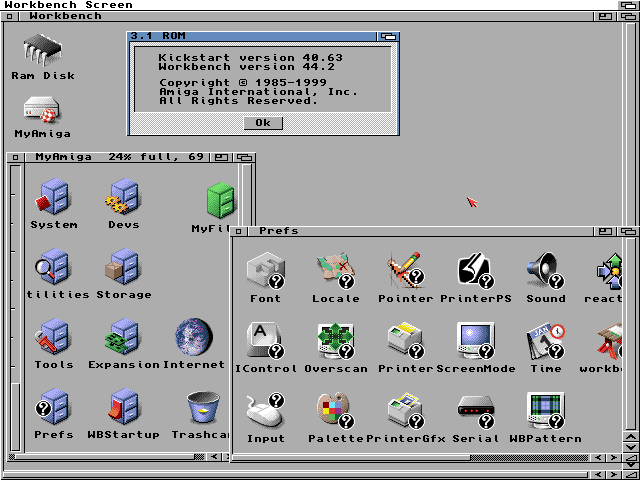
[](https://www.white-windows.ru/wp-content/uploads/2017/07/deskmateabout.png)

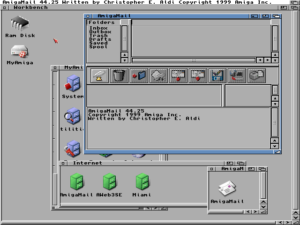
[](https://www.white-windows.ru/wp-content/uploads/2017/07/deskmatefiler2.png)     [](https://www.white-windows.ru/wp-content/uploads/2017/07/deskmatepuzzle.png)

В отличие от Norton Commander, в DeskMate 3.05 имелись полноценные меню, кнопки и некое подобие окон, которые можно размещать на примитивном рабочем столе. В среду был интегрирован файловый менеджер с ограниченной поддержкой древовидной структуры, встроенный учебник, календарь, органайзер, СУБД, текстовый и векторный графический редакторы. Другой отличительной чертой DeskMate 3.05 являлась поддержка основных цветов.

### **AmigaOS 3.5**

**AmigaOS** — операционная система, специально созданная для компьютеров семейства ***Amiga*** в 1985 году. Интерфейс первых версий представлял собой нечто среднее между оболочкой Apple Lisa и псевдографическими оболочками DOS, но уже в AmigaOS 3.5 появились перекрывающие друг друга окна, цветные иконки для быстрого доступа к файлам, приложениям и дисковым накопителям.

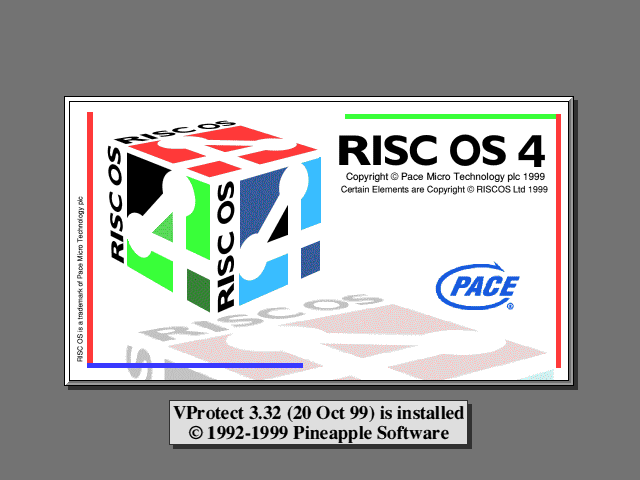
[](https://www.white-windows.ru/wp-content/uploads/2017/07/amiga35about.png)

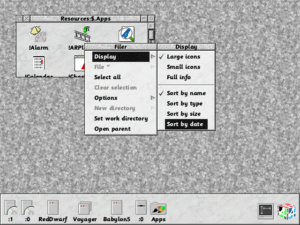
[](https://www.white-windows.ru/wp-content/uploads/2017/07/amiga35mail.png)     [](https://www.white-windows.ru/wp-content/uploads/2017/07/amiga35aweb.png)

Если брать в целом, AmigaOS 3.5 имеет много общего с MacOS. Меню в верхней части экрана показывает опции в зависимости от того, какие приложение являются в данным момент активными, окна оснащены простейшими элементами управления, есть полосы прокрутки. Отличительной чертой ***AmigaOS*** является функция, позволяющая работать с несколькими экранами, причём каждый экран мог иметь свое разрешение и глубину цвета.

### **RISC OS 4**

Маленькая, быстрая и несколько необычная ***Unix***-подобная операционная система, разработанная для платформы ***Raspberry Pi***. Внешне отличалась минималистичным пользовательским интерфейсом, поддержкой всех доступных на тот момент разрешений для компьютеров **Acorn**, перетаскивания, в ней имелись окна, своеобразная панель задач в нижней части рабочего стола, цветные иконки и псевдотрёхмерные декоративные элементы управления.

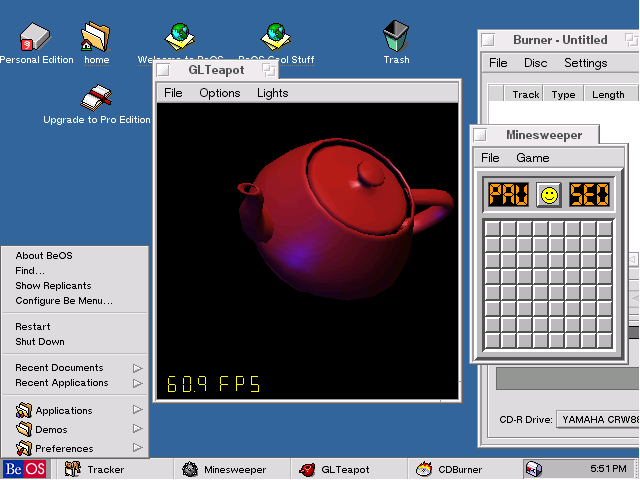
[](https://www.white-windows.ru/wp-content/uploads/2017/07/riscos4start.png)

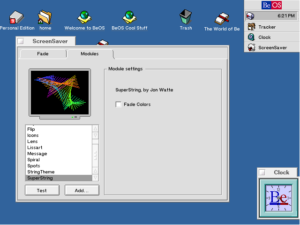
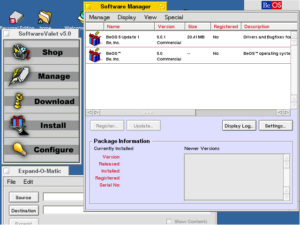
[](https://www.white-windows.ru/wp-content/uploads/2017/07/riscos4openingamenu.png)     [](https://www.white-windows.ru/wp-content/uploads/2017/07/riscos4config.png)

Собственными фишками **RISC OS 4** являлись однопользовательская многозадачность, модульность и двоичный интерфейс приложений, все файлы в ней представляли собой тома, приложения также были реализованы в виде каталогов с восклицательным знаком. Несмотря на минималистичность, с точки зрения пользователей, привыкших к окружению Windows или LXDE, рабочий стол RISC OS 4 мог показаться неудобным, так как многие способы управления в нём имели существенные отличия.

### **BeOS 5.0 PE**

Достаточно мощная операционная система, созданная компанией ***Be Inc***. и ориентированная на работу с мультимедиа. Изначально разрабатывалась для компьютеров **BeBox**, но затем перешла на **Macintosh**, а потом уже и на **PC**. История ***BeOS*** тесно связана с Apple, так как основателем Be Inc. являлся никто иной, как Жан-Луи Гассье - бывший исполнительный директор Apple. Помимо многопоточности и поддержки многопроцессорных архитектур, главной примечательностью BeOS 5.0 PE был её интерфейс, обладающий чертами Windows и систем от Apple.

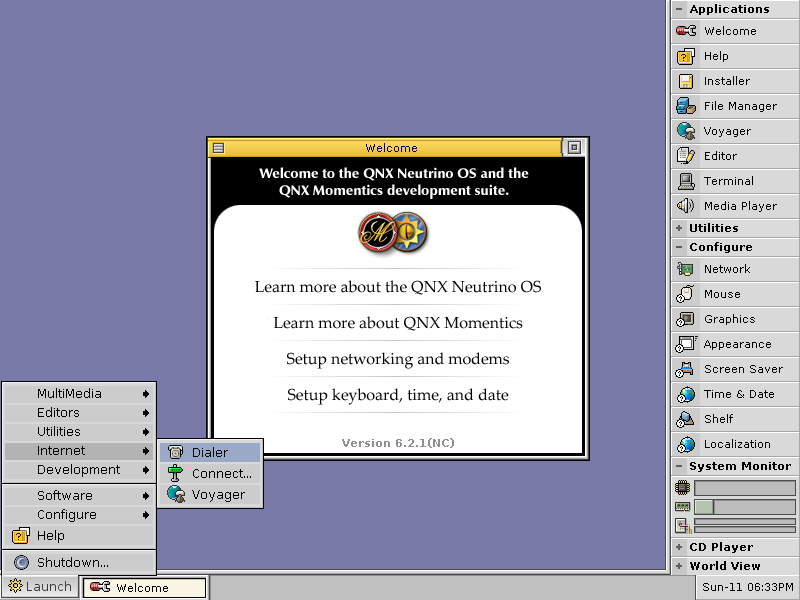
[](https://www.white-windows.ru/wp-content/uploads/2017/07/b5pedemos.png)

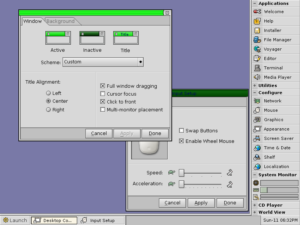
[](https://www.white-windows.ru/wp-content/uploads/2017/07/b5pessavers.png)     [](https://www.white-windows.ru/wp-content/uploads/2017/07/b5pevalet.png)

Оболочка системы выгодно отличалась хорошо продуманным юзабилити и соединённой с изяществом простотой. Как и положено, в ней имелись способные перекрывать друг друга окна, меню, «сборная» панель задач, имеющая сходство с аналогичным элементом оболочки Windows, полноцветные иконки и папки, которые можно было размещать на выполненном в минималистском стиле рабочем столе. Отличительной чертой ***BeOS 5.0 PE*** являлось строение окон — вместо традиционного заголовка в них использовались расположенные поверх окна вкладки.

### **QNX 6.2.1**

**QNX 6.2.1** — малоизвестная канадская ***Unix***-подобная система, относящаяся к типу так называемых операционных систем реального времени. Будучи универсальной, она отличалась высокой скоростью работы и нетребовательностью к аппаратным ресурсам. Графическая оболочка системы называется **Photon**, но её рабочий стол имеет много общего с десктопом Windows XP. В нижней части экрана имеется горизонтальная панель управления с подобием кнопки Пуск и вертикальная панель управления справа, поддерживается смена фоновых изображений.

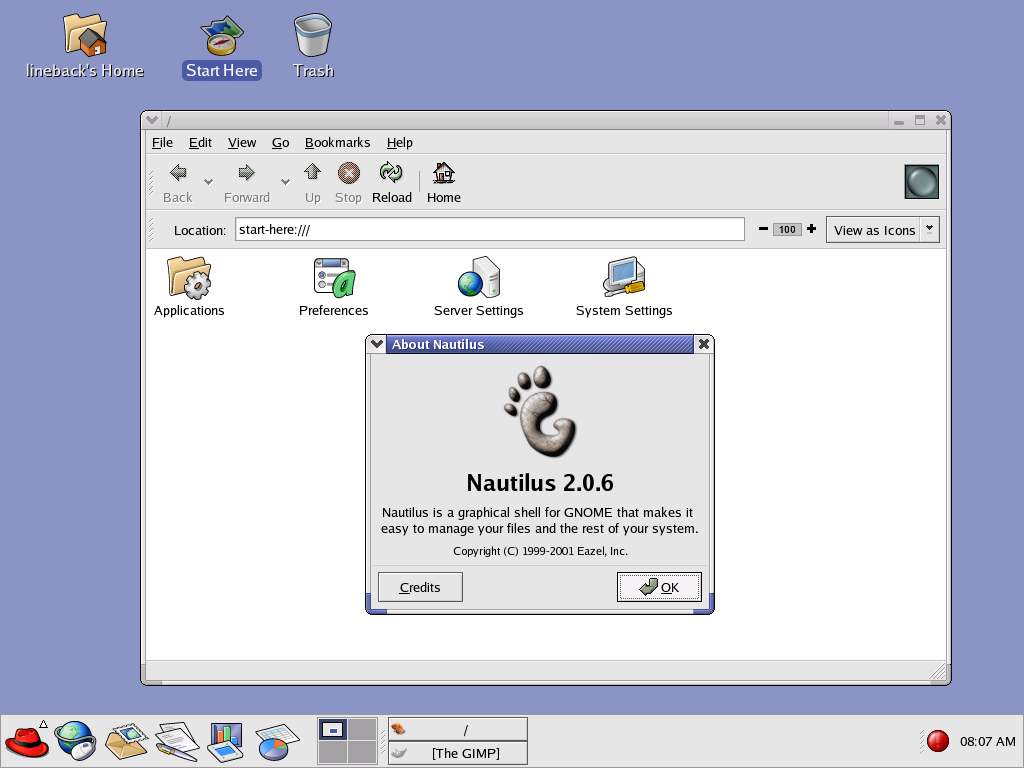
[](https://www.white-windows.ru/wp-content/uploads/2017/07/qnx621about2.png)

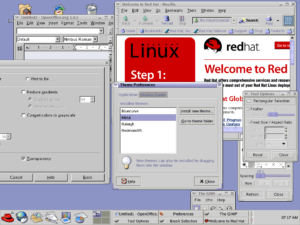
[](https://www.white-windows.ru/wp-content/uploads/2017/07/qnx621calc.png)     [](https://www.white-windows.ru/wp-content/uploads/2017/07/qnx621disp2.png)

Окна QNX 6.2.1, в которых запускаются приложения, имеют аналогичное строение с окнами Windows. Их можно сворачивать и разворачивать, масштабировать, располагать каскадом и прочее. Проводника, как его принято понимать в Windows в QNX 6.2.1 нет, его заменяет файловый менеджер **QNX Photon**, сходный по функциональности с Проводником Windows. Благодаря своей лёгкости, удобству и простоте QNX 6.2.1 в своё время конкурировала и в чём-то даже опережала Windows, однако приложений под QNX писалось мало, что и определило её дальнейшую судьбу. Сегодня QNX используется в основном на специализированных устройствах.

### **Red Hat 8.0 с интерфейсом GNOME 2.06**

Системы **Linux**, особенно ранние, обычно воспринимались как ориентированные на компьютерных гиков, однако нельзя сказать, чтобы разработчики ***Unix***-подобных систем ничего не предпринимали для того, чтобы сделать свои продукты столь же удобными, как и Windows. Примером тому может послужить **Red Hat 8.0** — основанная на Linux Fedora OC с оболочкой **GNOME 2.06**, кстати, изначально разрабатываемой именно для Red Hat и только потом распространившейся и на другие дистрибутивы Linux.

[](https://www.white-windows.ru/wp-content/uploads/2017/07/rh8defaultdesk.png)

[](https://www.white-windows.ru/wp-content/uploads/2017/07/rh8start.png)     [](https://www.white-windows.ru/wp-content/uploads/2017/07/rh8themes.png)

Red Hat 8.0 имеет простой и чистый рабочий стол с возможностью размещения на нём иконок и прочих объектов, Windows-подобную панель задач, окна, элементы управления которых располагались не в левом, как это свойственно многим Linux-системам, а в правом верхнем углу, есть в Red Hat 8.0 даже своя кнопка и меню Пуск. Сами меню организованы по каскадному принципу, что ещё больше в плане оформления роднило эту систему с Windows.

**4. Какова суть SILK - интерфейса.**

*SILK-интерфейс* расшифровывается - Spich (речь) Image (образ) Language (язык) Knowledge (знание). При использовании SILK-интерфейса на экране по речевой команде происходит перемещение от одних поисковых образов к другим по смысловым семантическим связям.

Общественный интерфейс будет включать в себя лучшие решения WIMP- и SILK-интерфейсов. Предполагается, что при использовании общественного интерфейса не нужно будет разбираться в меню. Экранные образы однозначно укажут дальнейший путь. Перемещение от одних поисковых образов к другим будет проходить по смысловым семантическим связям.

**5. В каких программных продуктах и приложениях наиболее целесообразно использование биометрической технологии, почему?**

Многие исследователи единогласно заявляют о том, что стандартные текстовые [пароли](https://www.tadviser.ru/index.php/%D0%A1%D1%82%D0%B0%D1%82%D1%8C%D1%8F:%D0%9F%D0%B0%D1%80%D0%BE%D0%BB%D0%B8" \o "Пароли) — весьма ненадежный способ авторизации, особенно сейчас, когда методы их кражи становятся все более изощренными. В связи с этим энтузиасты предлагают другие способы, и, кажется, фантазия их безгранична.

**Что такое биометрия?**

Биометрия является наиболее подходящим средством идентификации и аутентификации людей надежным и быстрым способом с использованием уникальных биологических признаков.

Биометрия позволяет идентифицировать и аутентифицировать человека на основе набора узнаваемых и проверяемых данных, которые являются уникальными и специфичными для них.

Биометрическая аутентификация — это процесс сравнения данных о характеристиках человека с его биометрическим «шаблоном» для определения сходства.

Эталонная модель сначала сохраняется в базе данных или защищенном переносном элементе, таком как смарт-карта.

Сохраненные данные затем сравниваются с биометрическими данными человека, которые должны быть аутентифицированы. Здесь проверяется личность человека.

В этом случае задается вопрос: «Вы действительно мистер или миссис Х?»

Биометрическая идентификация состоит из определения личности человека.

Цель состоит в том, чтобы получить элемент биометрических данных от этого человека. Это может быть фотография их лица, запись их голоса или изображение их отпечатка пальца.

Затем эти данные сравниваются с биометрическими данными нескольких других лиц, которые хранятся в базе данных.

В этом режиме вопрос прост: «Кто ты?»

**Биометрия: тенденции в 2020 году**

Биометрические технологии в сфере безопасности и государственного управления, включая сферу миграции, а также сегмент путешествий, определяют общие тенденции рынка. Третьим крупным рынком для биометрических технологий является финансовый сегмент, доля которого составляет порядка 15%. Доля сегмента здравоохранения составляет 9%

Ожидается, что мировой биометрический рынок к 2024 году превысит 50 миллиардов долларов США, согласно данным Global Markets Insights.

Разработка биометрии предлагает широкий спектр методов и может использоваться в самых разных областях. Эти технологии в основном используются в секторах судебной идентификации, банковских учреждениях, управления идентификацией, а также для биометрического контроля доступа в учреждениях и организациях.

Хорошо известные методы биометрии включают отпечатки пальцев, распознавание лиц, рисунок вен ладони, радужную оболочку и распознавание на основе ДНК. Ведутся исследования для новых типов биометрических систем, таких как форма ушей или лицевая термография. Вот несколько тенденций в биометрии, которые стоит ожидать в 2020 году и далее:

**Мультимодальная биометрия**

Мультимодальная биометрия предполагает объединение нескольких биометрических источников для повышения безопасности и точности. Мультимодальные биометрические системы требуют двух биометрических учетных данных - таких как лицо и отпечатки пальцев для положительной идентификации. Они способны преодолевать ограничения, которые обычно встречаются в унимодальных системах.

**Биометрия с многофакторной аутентификацией**

Биометрия улучшит многофакторную аутентификацию (MFA) в ближайшие годы. Геолокация, IP-адреса используемого устройства и схемы клавиш (биометрический элемент) могут создать мощную комбинацию для наиболее безопасной аутентификации пользователей.

**Биометрическая карта для оплаты**

Интеграция сканера отпечатков пальцев в смарт-карты - это новая тенденция, которую мы увидим в 2020 году. Биометрическая платежная карта поставляется с датчиком в области большого пальца и предлагает безопасный и удобный способ аутентификации людей.

Эти биометрические сенсорные карты открывают новый способ идентификации с помощью простого в использовании, портативного и безопасного устройства.

Запущенные в 2018 году Банком Кипра и Gemalto впервые для карт EMV (бесконтактная и контактная оплата), эти карты используют распознавание отпечатков пальцев вместо PIN-кода для аутентификации владельца карты.

**6. Что представляет собой семантический интерфейс?**

Этот вид интерфейса возник в конце 70-х годов XX века, с развитием искусственного интеллекта. Его трудно назвать самостоятельным видом интерфейса - он включает в себя и интерфейс командной строки, и графический, и речевой, и мимический интерфейс. Основная его отличительная черта - это отсутствие команд при общении с компьютером. Запрос формируется на естественном языке, в виде связанного текста и образов. По своей сути это трудно называть интерфейсом - это уже моделирование "общения" человека с компьютером. С середины 90-х годов XX века публикации, относящихся к семантическому интерфейсу, уже не встречались. Похоже, что в связи с важным военным значением этих разработок (например, для автономного ведения современного боя машинами - роботами, для "семантической" криптографии) эти направления были засекречены. Информация, что эти исследования продолжаются, иногда появляется в периодической печати (обычно в разделах компьютерных новостей).

7. Приведите примеры программ, относящихся к различным видам интерфейса.

Текстовый интерфейс пользователя:

• Командная строка(терминал)

Графического пользовательского интерфейса:

• простой: типовые экранные формы и стандартные элементы интерфейса, обеспечиваемые самой подсистемой GUI;

• истинно-графический, двумерный: нестандартные элементы интерфейса и оригинальные метафоры, реализованные собственными средствами приложения или сторонней библиотекой;

• трёхмерный.

Тактильный интерфейс (сенсорный интерфейс)

• Apple iWatch

• Корейцы недавно представили серию многофункциональных принтеров Smart MultiXpress, оснащённых «планшетным» интерфейсом с разнообразной тактильной связью.

• браслет Tactilu, который передаёт «прикосновение» от одного пользователя другому.

**Практическая работа №2**

**Контрольные вопросы**

1. Что такое прототипирование интерфейсов?

это процесс в работе дизайнера, когда он создаёт «иллюзию» рабочего интерфейса на базе нарисованных макетов или вайрфреймов в короткие временные сроки. Это и будет являться прототипом интерфейса.

1. Какие вы знаете версии прототипов интерфейсов

Бумажное прототипирование относится к статичным моделям, которые отличаются содержанием статичных изображений. Помимо концептов, нарисованных от руки, к статичным прототипам относятся эскизы, созданные в графических редакторах, нарисованные на планшете или маркерной доске. Они отображают исключительно проект дизайна сайта.

В отличие от статических, интерактивные прототипы представляют собой проектирование взаимодействия всех составляющих ресурса. Это упрощенные макеты всех страниц вэбсайта с высокой детализацией. При этом все элементы кликабельны. Вы можете перейти со страницы на страницу, развернуть меню и т.п. Интерактивный прототип помогает лучше понять механизм работы проекта, осуществить поставленные проектные решения, выявить целесообразность их реализации в готовом макете.

1. Опишите бумажную версию прототипа интерфейса

Сессию бумажного прототипирования стоит провести вместе с клиентом — так вы сможете быстро показать ему, какие функции планируете добавить в приложение. Плюс, вы сразу проверите, насколько все понятно и интуитивно для пользователей.

При помощи бумажного прототипа, можно вовлечь пользователей в тестирование на очень ранних стадиях — а это помогает здорово ускорить процесс дизайна. Кроме того, бумажные прототипы позволяют снять креативные блоки в процессе работы. Любую идею можно быстро визуализировать, чтобы посмотреть на нее под другим углом.

## 10 причин использовать бумажные прототипы

1. **Быстро и просто.**Нужна только ручка и листок бумаги.
2. **Быстрые итерации.**Можно собирать разные идеи и быстро их тестировать, чтобы выбрать стоящие. А еще, если вы что-то пропустили в первой версии прототипа, можно взять и дорисовать — и протестировать на той же сессии.
3. **Проверка идей.**Новую идею можно тут же визуализировать и быстро проверить на жизнеспособность.
4. **Обратная связь и комментарии.**Наблюдая за пользователями, вы можете сделать интересные открытия, и тут же внедрить идеи в прототип.
5. **Доступность.**Не нужны никакие специальные навыки: этим методом могут воспользоваться и разработчики, и другие специалисты.
6. **Взаимодействие с пользователем и более глубокие интервью.**Почему пользователь решил нажать на эту кнопку? Как, по его мнению, работает приложение? Что он пытается сделать в нашем интерфейсе?
7. **Творчество и генерация идей.** Отличный способ прокачать креативность и обменяться идеями с командой. Плюс, много идей обычно возникает прямо во время сессии.
8. **Вовлечение клиента.**Бумажное прототипирование помогает вовлечь клиента в процесс дизайна. Благодаря этому финальный продукт будет более целостным. Плюс, клиент будет понимать причины тех или иных изменений.
9. **Выявление проблем, не связанных с UI.**В ходе сессии могут выплыть другие проблемы: например, с концепцией продукта в целом.
10. **Смена окружения и перспективы.**Иногда бывает здорово отлипнуть от компьютера и поговорить с кем-то лично. Это также здорово заряжает на творчество.
11. Опишите презентационную версию прототипа интерфейса

После исчерпания возможностей бумажной версии прототипа стоит создать новую (следующую) версию. Для этого точно так же рисуется интерфейс, но уже не на бумаге, а в какой-либо презентационной программе. С этой версией прототипа можно тестировать значительно более сложное взаимодействие человека с системой, нежели с бумажной. С другой стороны, исправление найденных ошибок значительно более трудоёмко. Такая версия прототипа может создаваться уже на этапе высокоуровневого проектирования, а на этапе низкоуровневого проектирования ПИ она обязательна.

Достаточно распространенным инструментом для создания прототипов этого типа является на сегодняшний день *MS Visio'1.* При работе с *Visio* можно выбрать одну из двух возможностей: можно либо [[1]](https://ozlib.com/849294/informatika/prezentatsionnaya_versiya_prototipa" \l "gads_btm)

рисовать все экраны на одном листе, соединяя взаимосвязанные элементы управления и экраны линиями, либо рисовать каждый экран на отдельном листе, связывая экраны ссылками. Первый вариант удобен, поскольку позволяет оценить интерфейс в целом, второй предпочтительнее для заказчика и субъектов тестирования, поскольку его легче понять. Как правило, превратить второй вариант в первый оказывается гораздо легче.

Большим достоинством *Visio* является возможность записывать результат в *HTML-*файл, что позволяет без проблем тестировать интерфейс на территории субъектов тестирования.

В настоящее время практически все программы машинного проектирования имеют возможность создания прототипов. Создано большое количество отдельных пакетов упрощающих жизнь проектировщиков компьютерных интерфейсов и сайтов. Их число постоянно увеличивается, что не позволяет нам выбрать лучшие решения. В качестве примера приведём краткие описания ряда из них.

*Mockingbird* — очень популярный инструмент для создания прототипов. Он позволяет легко создавать, просматривать и редактировать макет, а так же обмениваться наработками друг с другом. Сервис обладает красивым и удобным интерфейсом, включает привязку к сетке, перетаскивания, большой набор разнообразных элементов и ряд других полезных функций и возможностей.

*Pencil Project* — скорее большой плагин для FireFox, нежели онлайн сервис или веб-приложение. После его установки создавать схемы, макеты и пользовательские интерфейсы можно прямо в браузере. Имеется большое количество инструментов и функций (многостраничные документы, поддержка текста, экспорт *html, png*, файлов *OpenOffice* и многое другое).

1. Опишите псевдореальную версию прототипа интерфейса

# **Псевдореальная версия прототипа**

В тех случаях, когда в интерфейсе появляются нестандартные элементы или необходимо проверить реальную скорость взаимодействия пользователя с системой, создаётся ещё одна версия прототипа — реально выглядящая, но лишенная каких-либо алгоритмов и, соответственно, не показывающая реальных данных. Делать этот вариант можно как в средах разработки, благо в них есть визуальные инструменты создания интерфейсов, так и в редакторах изображений, что обычно быстрее. Создаются снимки экрана, на которых и производят тестирование. Понятно, что как-то модифицировать эти экраны затруднительно, так что лучше не увлекаться такой работой, не получив каких-либо гарантий её правильности. Эта версия прототипа больше соответствует этапу низкоуровневой разработки ПИ, но может применяться и на этапе высокоуровневой разработки.

1. Опишите реальную версию прототипа интерфейса

# **Реальная версия прототипа**

Иногда необходимо тестировать взаимодействие пользователя не только с интерфейсом системы, но и с обрабатываемыми системой данными. Например, работая с графической программой, пользователь не только нажимает на экранные кнопки, но также создает и модифицирует изображения мышью. Область же редактирования данных зачастую вообще не содержит каких-либо визуальных интерфейсных элементов, из чего вовсе не следует, что интерфейса в ней нет, его наоборот много. Просто счёт в нем идет не на кнопки и переключатели, но на пиксели и миллисекунды.

Понятно, что прототип в таких условиях практически не будет отличаться от готового ПИ. Поэтому лучше всего написать нужные участки программы до написания всего остального и проводить юзабилити-тестирование на реальной версии прототипа ПИ. Понятно, что прототип такой версии, если он все-таки разрабатывается, возможен только на этапе низкоуровневого проектирования.

1. Какие прототипы называются статичные прототипы?

**Статический прототип** – это детально прорисованные страницы, в которых по внешнему виду понятно поведение элементов, но весь сайт в данном случае будет представлен как набор изображений. В целом статический прототип от сетки будет отличаться более детальной прорисовкой элементов и иллюстрацией поведения интерактивных элементов.

1. Какие прототипы называются динамические прототипы?

**Динамический прототип** – это рабочая модель системы, в которой можно прочувствовать, как работает система.

**Практическая работа № 3**

Контрольные вопросы:

1. Что такое интерфейс пользователя (UI) – это

Пользовательский интерфейс (user interface или сокращенно UI) – это интерфейс, с помощью которого человек может управлять программным обеспечением или аппаратным оснащением. UI должны быть удобными в использовании, чтобы взаимодействие с ними происходило на максимально интуитивном уровне. Интерфейсы программного обеспечения также называют графическими пользовательскими интерфейсами (graphical user interface или GUI).

1. Что такое Эргономика?

Эргономика – это наука, основанная на физиологии, технике и психологии того, как люди взаимодействуют со своей рабочей средой. Цель данной науки, это предоставление рекомендаций по повышению эффективности и комфорта при обустройстве рабочей среды.

1. На какие этапы ЖЦ разбивается процесс разработки пользовательского интерфейса?

**Манипулирование простой графикой - не более 2 сек**

**Время ответа системы**

**Массовому пользователю необходимо подробно объяснить характер ошибки и пути ее исправления.**

**Опытным пользователям лишь указать на наличие ошибки;**

**Следует предусмотреть вариантность в обеспечении пользователя нужной информацией для исправления ошибки;**

**Следует избегать повторного ввода после исправления ошибки верно введенных ранее данных;**

**Сообщения об ошибках должны быть заранее специфицированы и четко определены;**

**Диагностика ошибок должна быть максимально ясной;**

**Избегать зашифрованных сообщений об ошибках;**

**Неадекватными инструкциями для поведения в непредвиденных ситуациях.**

**Недостатком мотивации (отсутствие интереса);**

**Пользовательской скукой;**

**при разработке процедур обнаружения и диагностики ошибок следует:**

**При разработке пользовательского интерфейса большое значение имеет обоснованный выбор времени ответа системы на различные запросы пользователей. При этом:**

**- надо стремиться к постоянству времени реакции системы на однотипные запросы пользователей**

**- учитывать, что время реакции человека в среднем составляет 2 сек.**

**Некоторые характерные времена реакции системы:**

**- ввод с клавиатуры – не более 0.1 – 0.2 сек**

**- инициализация системы - не более 3 сек**

**- вставка символов - не более 2 сек**

**- выполнение простых запросов - не более 2 сек**

**- выполнение сложного запроса - не более 5 сек**

**- листание страницы - не более 1 сек**

**- выбор функции - не более 2 сек**

Процесс разработки пользовательского интерфейса (ПИ) разбивается на этапы жизненного цикла:

Анализ трудовой деятельности пользователя, объединение бизнес-функций в роли.

1. 2. Построение пользовательской модели данных, привязка объектов к ролям и формирование рабочих мест.

3. Формулировка требований к работе пользователя и выбор показателей оценки пользовательского интерфейса.

4. Разработка обобщенного сценария взаимодействия пользователя с программным модулем (функциональной модели) и его предварительная оценка пользователями и Заказчиком.

5. Корректировка и детализация сценария взаимодействия, выбор и дополнение стандарта (руководства) для построения прототипа.

6. Разработка макетов и прототипов ПИ и их оценка в деловой игре, выбор окончательного варианта.

7. Имплементация ПИ в коде, создание тестовой версии.

8. Разработка средств поддержки пользователя (пользовательские ловари, подсказки, сообщения, помощь и пр.) и их встраивание в программный код.

9. Usability - тестирование тестовой версии ПИ по набору ранее определенных показателей.

10. Подготовка пользовательской документации и разработка программы обучения.

1. Что такое юзабилити?

Юзабилити — это показатель того, насколько легко и удобно пользователю взаимодействовать с интерфейсом сайта. С английского usability так и переводится — удобство использования.

Когда человек попадает на сайт, он должен легко сориентироваться, быстро найти нужную информацию и сделать то, чего ждёт владелец страницы: купить товар, оставить заявку на расчёт, подписаться на рассылку или прочитать статью.

1. Для того, чтобы разобраться в технологии решения задач пользователя, разработчику

Для того, чтобы разобраться в технологии решения задач пользователя, разработчику необходимо выяснить следующие моменты (исследуя деятельность пользователя):

1. *Какая информация необходима пользователю для решения задачи?*
2. *Какую информацию пользователь может игнорировать (не учитывать)?*
3. *Совместно с пользователем разделить всю информацию на сигнальную, отображаемую, редактируемую, поисковую и результирующую.*
4. *Какие решения пользователю необходимо принимать в процессе работы с программой?*
5. *Может ли пользователь совершать несколько различных действий (решать несколько задач) одновременно?*
6. *Какие типовые операции использует пользователь при решении задачи?*
7. *Что произойдет, если пользователь будет действовать не по предписанному Вами алгоритму, пропуская те или иные шаги или обходя их?*

*Производительность работы*отражает объем затраченных ресурсов при выполнении задачи, как вычислительных, так и психофизиологических.

Дизайн ПИ должен обеспечивать минимизацию усилий пользователя при выполнении работы и приводить к:

* сокращению длительности операций чтения, редактирования и поиска информации,
* уменьшению времени навигации и выбора команды,
* повышению общей продуктивности пользователя, заключающейся в объеме обработанных данных за определенный период времени.
* увеличению длительности устойчивой работы пользователя и др.

Какую информацию пользователь может игнорировать (не учитывать)?

Стилевая гибкость

– возможность использовать различные интерфейсы с одним и тем же приложением, на практике реализуется в виде набора “skins”, для web-интерфейсов – с помощью таблицы стилей, в том числе возможность в выборе пользователем собственных установок ПИ (цвет, иконы, подсказки и пр.).

Совместное наращивание функциональности

– возможность развивать приложение без разрушения (т.е. оставаясь в рамках) существующего интерфейса.

Масштабируемость

– возможность легко настраивать и расширять как интерфейс, так и само приложение при увеличении числа пользователей, рабочих мест, объема и характеристик данных.

Адаптивность к действиям пользователя

– приложение должно допускать возможность ввода данных и команд множеством разных способов (клавиатура, мышь, другие устройства) и много вариативность доступа к прикладным функциям (иконы, «горячие клавиши», меню …), кроме того программа должна учитывать возможность перехода и возврат от окна к окну, от режима к режиму, и правильно обрабатывать такие ситуации.

Независимость в ресурсах

– для создания пользовательского интерфейса должны предоставляться отдельные ресурсы, направленные на хранение и обработку данных, необходимых для поддержки пользователя (пользовательские словари, контекстно-зависимые списки, наборы данных по умолчанию или по последнему запросу, истории запросов и пр.)

Переносимость

– при переходе на другую аппаратную (программную) платформу, должен осуществляется автоматически перенос и пользовательского интерфейса, и конечного приложения.

1. Совместно с пользователем разделить всю информацию на сигнальную, отображаемую, редактируемую, поисковую и результирующую.

Совместно с пользователем разделить всю информацию на сигнальную, отображаемую, редактируемую, поисковую и результирующую.

Какие решения пользователю необходимо принимать в процессе работы с программой?

Может ли пользователь совершать несколько различных действий (решать несколько задач) одновременно?

Какие типовые операции использует пользователь при решении задачи?

Что произойдет, если пользователь будет действовать не по предписанному Вами алгоритму, пропуская те или иные шаги или обходя их?

**Практическая работа №5**

**Контрольные вопросы**

1. В каком году был озвучен закон Фитса?
2. Дайте понятие закона Фитса.
3. Что утверждает закон Фитса?
4. Привести пример использования закона Фитса.

Ответы:

1. Закон Фиттса озвучен – 1954 году
2. Закон Фиттса — общий закон, касающийся сенсорно-моторных процессов, связывающий время движения с точностью движения и с расстоянием перемещения: чем дальше или точнее выполняется движение, тем больше коррекции необходимо для его выполнения, и соответственно, больше времени требуется для внесения этой коррекции.
3. Закон Фиттса гласит, что «*время достижения цели зависит от расстояния до цели и ее размера*». По сути, Быстрые движения + маленькие цели приводят к большему количеству ошибок.
4. Например: чем дальше кнопка (от последнего клика мыши) и чем меньше её размер, тем больше времени необходимо пользователю для клика по ней. Также закон Фиттса можно использовать для намеренного осложнения пути пользователя, например сделав меньше кнопку «Удалить» пользователи реже будут кликать по ней случайно.