

Министерство образования Республики Беларусь  
Учреждение образования «Белорусский государственный университет  
информатики и радиоэлектроники»

Факультет компьютерных систем и сетей  
Кафедра информатики  
Дисциплина «Операционные системы и среды»

«К ЗАЩИТЕ ДОПУСТИТЬ»  
Руководитель курсового проекта  
магистр техн.наук, ассистент  
\_\_\_\_\_ А.В. Давыдчик  
\_\_\_\_\_.\_\_\_\_\_.2023

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**  
к курсовому проекту  
на тему:  
**«АНАЛИЗ КОНФИГУРАЦИИ И ПОДКЛЮЧЕННЫХ УСТРОЙСТВ,  
ПРОИЗВОДИТЕЛЬНОСТИ ПОДСИСТЕМ КОМПЬЮТЕРА,  
ОГРАНИЧЕННОЕ УПРАВЛЕНИЕ УСТРОЙСТВАМИ»**

БГУИР КП 1-40 04 01 025 ПЗ

Выполнила студент группы 053504  
ЦАРЁВА Ангелина Александровна

\_\_\_\_\_  
(подпись студента)  
Курсовой проект представлен на  
проверку \_\_\_\_\_.\_\_\_\_\_.2023  
\_\_\_\_\_  
(подпись студента)

Минск 2023

## СОДЕРЖАНИЕ

<b>ВВЕДЕНИЕ.....</b>	<b>4</b>
<b>1 ОПЕРАЦИОННАЯ СИСТЕМА И ЕЕ ПОДСИСТЕМЫ.....</b>	<b>6</b>
<b>1.1 Понятие операционной системы .....</b>	<b>6</b>
<b>1.2 Состав операционной системы и назначение компонентов .....</b>	<b>7</b>
<b>1.3 Подсистемы операционной системы .....</b>	<b>8</b>
<b>2 АНАЛИЗ КОНФИГУРАЦИИ И ПРОИЗВОДИТЕЛЬНОСТИ КОМПЬЮТЕРА .....</b>	<b>14</b>
<b>3 УПРАВЛЕНИЕ УСТРОЙСТВАМИ КОМПЬЮТЕРА .....</b>	<b>16</b>
<b>4 ТЕОРЕТИЧЕСКОЕ ОБОСНОВАНИЕ РАЗРАБОТКИ ПРОГРАММНОГО ПРОДУКТА ..</b>	<b>18</b>
<b>ЗАКЛЮЧЕНИЕ.....</b>	<b>19</b>
<b>СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....</b>	<b>20</b>
<b>ПРИЛОЖЕНИЕ А .....</b>	<b>21</b>

## ВВЕДЕНИЕ

Современные компьютеры являются сложными техническими устройствами, состоящими из множества подсистем и устройств, которые могут быть подключены к ним. Каждый компонент компьютера играет свою роль в общей работе системы, и настройка каждого из них может влиять на производительность компьютера в целом.

Один из важных аспектов работы с компьютером - это управление устройствами, подключенными к нему. Некоторые из этих устройств могут иметь ограниченное управление, что может ограничить их функциональность и влиять на работу всей системы.

Для того чтобы добиться оптимальной производительности компьютера, необходимо проанализировать его конфигурацию и подключенные устройства. Это может включать в себя анализ железных компонентов, таких как процессор, оперативная память, жесткий диск, видеокарта и т.д. Также важно убедиться в правильности настроек операционной системы и программного обеспечения.

Один из способов анализа конфигурации компьютера - это использование специальных программ для тестирования производительности. Такие программы могут помочь выявить узкие места в работе компьютера и оптимизировать его работу. Помимо этого, важно проанализировать все устройства, подключенные к компьютеру. Это может включать в себя внешние жесткие диски, принтеры, сканеры, мониторы и другие устройства. Необходимо убедиться, что они правильно подключены, настроены и не мешают работе компьютера.

Некоторые устройства могут иметь ограниченное управление, что может ограничить их функциональность и влиять на работу всей системы. Например, если на компьютер подключен принтер, который имеет ограниченные возможности настройки, то это может ограничить возможности пользователя. Для решения этой проблемы может быть использована специальная программа управления устройствами. Она позволяет настраивать и управлять всеми устройствами, подключенными к компьютеру, включая те, которые имеют ограниченное управление.

Также важно отметить, что при анализе конфигурации компьютера и производительности его подсистем необходимо учитывать целый ряд факторов. Один из них - это возможность апгрейда компонентов. Если конфигурация компьютера не соответствует требованиям пользовательских задач, то возможно потребуется замена или улучшение компонентов, таких

как оперативная память, жесткий диск или видеокарта.

Кроме того, необходимо учитывать особенности работы с определенными приложениями. Некоторые программы могут требовать определенных характеристик компьютера для работы в полную силу, например, приложения для обработки видео или игры с высокими требованиями к производительности. В таких случаях важно настроить конфигурацию компьютера с учетом требований приложения.

Также следует учитывать влияние внешних факторов на производительность компьютера. Например, влияние температурных условий на работу процессора или влияние запущенных фоновых процессов на производительность. Для устранения таких проблем может потребоваться проведение дополнительных настроек и оптимизация системы.

Важным аспектом управления устройствами является безопасность. Некоторые устройства, такие как сетевые адаптеры или внешние жесткие диски, могут содержать конфиденциальную информацию, которую необходимо защитить от несанкционированного доступа. В таких случаях необходимо использовать специальные программы для шифрования данных или установки паролей доступа.

В заключение, анализ конфигурации и подключенных устройств, производительности подсистем компьютера и управление устройствами являются важными аспектами при работе с компьютером. Необходимо учитывать множество факторов, включая требования пользовательских задач, возможности апгрейда компонентов, влияние внешних факторов и безопасность. Это позволит обеспечить оптимальную работу компьютера и улучшить производительность системы в целом.

Целью данной курсовой работы является проведение анализа конфигурации и подключенных устройств компьютера, а также оценка производительности его подсистем. Также будут рассмотрены вопросы ограниченного управления устройствами, для предотвращения возможных проблем и повышения безопасности и производительности компьютера.

# 1 ОПЕРАЦИОННАЯ СИСТЕМА И ЕЕ ПОДСИСТЕМЫ

## 1.1 Понятие операционной системы

Операционная система (ОС) – это комплект программного обеспечения, предназначенный для управления компьютером и обеспечивающий поддержку хранения, исполнения и разработки прикладных программ.

Операционная система выполняет роль своеобразного интерфейса (Интерфейс – совокупность аппаратуры и программных средств, необходимых для подключения периферийных устройств к персональной электронной вычислительной машине (ПЭВМ)) между пользователем и вычислительной системой (ВС). Вычислительная система – взаимосвязанная совокупность аппаратных средств вычислительной техники и программного обеспечения, предназначенная для обработки информации. Это означает, что ОС в значительной степени формирует у пользователя представление о возможностях ВС, удобстве работы с ней, ее пропускной способности.

Различные операционные системы на одних и тех же технических средствах могут предоставить пользователю различные возможности для организации вычислительного процесса или автоматизированной обработки данных. В программном обеспечении ВС операционная система занимает основное положение, поскольку осуществляет планирование и контроль всего вычислительного процесса. Любая из компонент программного обеспечения обязательно работает под управлением операционной системы.

Современные операционные системы решают следующие задачи:

- хранение, загрузка и исполнение прикладных программ;
- организация файловой системы на устройствах долговременной памяти;
- планирование и динамическое перераспределение ресурсов компьютера;
- взаимодействие параллельных программ и синхронизация их работы;
- учет и разграничение полномочий пользователей системы;
- предоставление пользовательского интерфейса для работы с компьютером;
- защита данных от несанкционированного доступа, разрушения, других случайных или намеренных вредоносных действий, как во время выполнения программ, так и при хранении данных в файлах.

Современные операционные системы являются многозадачными, то есть пользователь может запускать одновременно несколько приложений,

наблюдая результат выполнения каждой из них. Это возможно благодаря конструкции ОС и функциональности современных процессоров – не зря операционные системы пишутся для процессора, а не наоборот. Современный процессор представляет собой не одноядерное, а двухъядерное и даже четырехъядерное решение, что увеличивает его производительность во много раз. Этим пользуется операционная система, оптимально распределяя ресурсы процессора между всеми запущенными процессами.

Главными характеристиками операционной системы являются стабильность ее работы и устойчивость к различным угрозам – внешним (вирусам) и внутренним (аппаратным сбоям и конфликтам). Сегодня выпускается два типа операционных систем – серверные и однопользовательские. Первые представляют собой серьезную программную реализацию, поскольку содержат механизмы, с помощью которых контролируются отношения и поддерживается локальная сеть. Вторые – облегченные, могут работать как автономно, так и в составе сети, подчиняясь ее правилам.

Важным свойством операционных систем является унификация программного обеспечения. Раньше программы были машинно-зависимыми. То есть программа, написанная для одного компьютера, не могла работать на другом, пусть даже таком же компьютере, без корректировки. С появлением операционных систем, программистам больше не надо переписывать приложения для каждого нового компьютера, так как все машинно-зависимые части программы перенесены в код операционных систем.

Операционная система организована таким образом, чтобы допускались эффективная разработка, тестирование и внедрение новых приложений и системных функций, причем это не препятствует нормальному функционированию вычислительной системы.

В определении состава операционной системы значение имеет критерий операциональной 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).

## **1.2 Состав операционной системы и назначение компонентов**

Операционная система обеспечивает совместное функционирование всех устройств компьютера и предоставляет пользователю доступ к его

ресурсам. Процесс работы компьютера в определенном смысле сводится к обмену файлами между устройствами. В операционной системе имеются программные модули, управляющие файловой системой.

В состав операционной системы входит командный процессор, который запрашивает у пользователя команды и выполняет их. Пользователь может дать, например, команду выполнения какой-либо операции над файлами (копирование, удаление, переименование), команду вывода документа на печать и т. д. Операционная система должна эти команды выполнить.

К магистрали компьютера подключаются различные устройства (дисководы, монитор, клавиатура, мышь, принтер и др.). В состав операционной системы входят драйверы устройств — специальные программы, которые обеспечивают управление работой устройств и согласование информационного обмена с другими устройствами. Любому устройству соответствует свой драйвер.

Для упрощения работы пользователя в состав современных операционных систем, и в частности в состав *Windows*, входят программные модули, создающие графический пользовательский интерфейс. В операционных системах с графическим интерфейсом пользователь может вводить команды посредством мыши, тогда как в режиме командной строки необходимо вводить команды с помощью клавиатуры.

Операционная система содержит также сервисные программы, или утилиты. Такие программы позволяют обслуживать диски (проверять, сжимать, дефрагментировать и т. д.), выполнять операции с файлами (архивировать и т. д.), работать в компьютерных сетях и т. д.

Для удобства пользователя в операционной системе обычно имеется и справочная система предназначенная для оперативного получения необходимой информации о функционировании как операционной системы в целом, так и о работе ее отдельных модулей.

### **1.3 Подсистемы операционной системы**

В любой современной операционной системе, независимо от архитектуры и методов реализации, можно выделить несколько относительно независимых базовых подсистем, совместно работающих для решения перечисленных выше задач. Состав операционной системы и связь ее базовых подсистем с решаемыми задачами, схематично показаны на рисунке 1.

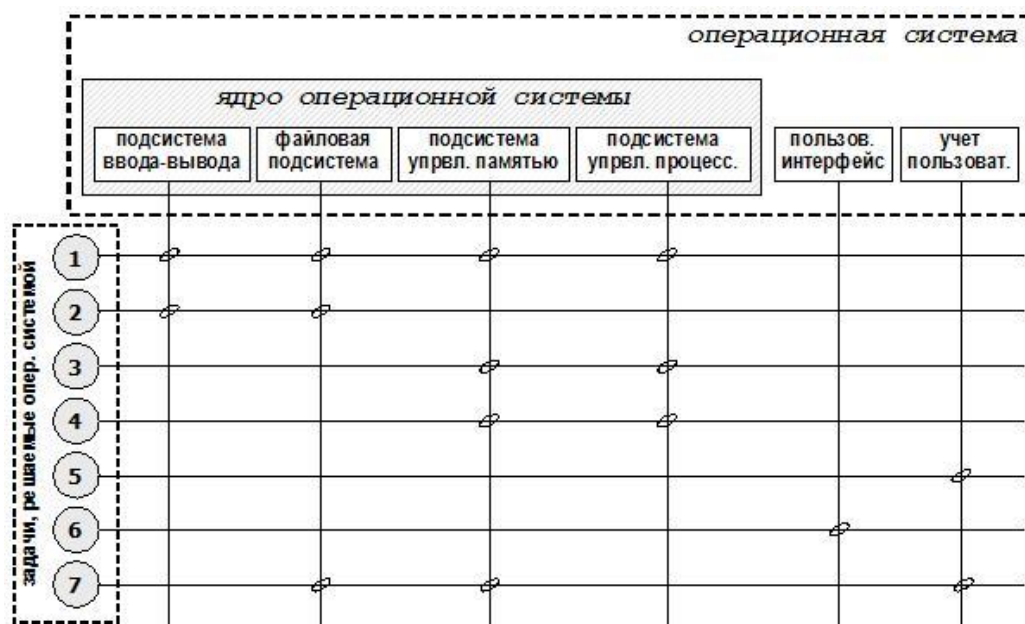


Рисунок 1 – Состав операционной системы

Ряд подсистем операционной системы на рисунке 1 объединены в общий блок, названный ядро операционной системы.

Понятие ядра является одним из наиболее важных понятий, как и роль ядра в операционной системе. Ядро объединяет в себе базовые компоненты операционной системы, во многом определяющие специфику и основную область применения данной операционной системы. Важнейшим свойством ядра, отличающим его от любых других системных или пользовательских программ, является то, что код ядра выполняется с максимально высоким уровнем привилегий – ему доступны все команды процессора и разрешен прямой доступ к любой аппаратуре компьютера. В конечном итоге, ядро изолирует все остальные программы от аппаратуры компьютера. Все остальные программы получают доступ к аппаратным ресурсам только через посредничество ядра операционной системы.

Функциональность ядра во многом определяет функциональность всей операционной системы. Чем больше возможности ядра системы, тем более разносторонний сервис операционная система может в конечном итоге предоставить пользователю. Однако потребности различных категорий пользователей в универсальных операционных системах могут существенно различаться. Возможности, необходимые одним пользователям, могут совершенно не интересовать других, и наоборот. Ядро же должно удовлетворить всех, в результате код ядра становится очень громоздким, оно начинает расходовать неоправданно много ресурсов, прежде всего памяти.



Так как код ядра в силу своей специфики является аппаратно зависимым, громоздкое ядро универсальной операционной системы оказывается наиболее узким местом при переносе операционной системы на другую аппаратную платформу, например, на вновь разработанный или даже просто модернизированный компьютер. Также от устойчивости кода ядра зависит устойчивость всей операционной системы, а отлаживать громоздкое ядро чрезвычайно сложно.

Стремление найти кардинальное решение описанных проблем привело к появлению новой концепции построения операционной системы – концепции микроядра. При этом многие подсистемы, традиционно относимые к ядру, например, файловая система, выносятся из ядра. Код ядра предельно минимизируется и самостоятельно уже не обеспечивает даже базовой функциональности операционной системы. Зато его легко отладить и относительно легко переписать при переносе операционной системы на новую аппаратную платформу.

Подсистема ввода-вывода принимает запросы на передачу или прием данных от всех выполняющихся программ, как прикладных, так и системных, упорядочивает их и переадресует внешним устройствам с учетом их аппаратных особенностей.

Можно выделить две основные функции подсистемы ввода-вывода:

- изоляция всех других компонентов операционной системы и прикладных программ от особенностей аппаратной реализации внешних устройств компьютера;
- организация бесконфликтного и эффективного использования внешних устройств всеми исполняющимися программами.

Фактически, задачу изоляции от аппаратных особенностей внешних устройств решают драйверы – специальные подпрограммы в составе подсистемы ввода-вывода, которые непосредственно взаимодействуют с внешними устройствами, при этом для каждого внешнего устройства необходим собственный драйвер. Подсистема ввода-вывода любой современной операционной системы организована так, чтобы набор драйверов можно было легко заменять, обеспечивая работу прикладных программ и самой операционной системы при различных аппаратных конфигурациях компьютера.

Как показывают многочисленные замеры и эксперименты, программы, кроме чисто вычислительных, большую часть своей жизни проводят в состоянии ожидания завершения операций ввода-вывода. Поэтому от эффективности работы подсистемы ввода-вывода существенно зависит

эффективность работы всех остальных программ. Для повышения скорости выполнения запросов ввода-вывода, подсистема ввода-вывода широко использует такие приемы, как переупорядочивание запросов ввода-вывода или кэширование данных. В совокупности, подобные меры могут поднять производительность подсистемы ввода-вывода более, чем в 100 раз. В дальнейшем, при детальном изучении подсистемы ввода-вывода, основные приемы повышения производительности будут рассмотрены довольно подробно.

Файловая подсистема обеспечивает пользователя удобным, аппаратно независимым интерфейсом для упорядоченного хранения пользовательских программ и данных на внешних устройствах долговременной памяти.

В современных ЭВМ в качестве основного устройства долговременной памяти обычно используется накопитель на жестких магнитных дисках. Физически, данные на диске адресуются подобно ячейкам массива – по номерам блоков данных. Современный диск содержит миллиарды таких блоков, поэтому человеку практически невозможно работать с сырыми данными на диске. Фактически, файловая система позволяет всего лишь обращаться по осмысленному символическому имени к группе дисковых блоков, представляющих логически связанные данные, но это делает работу с данными на диске не только возможной, но и весьма удобной для человека.

Кроме того, благодаря файловой подсистеме пользователи совершенно единообразно работают с различными типами внешних устройств долговременной памяти. Причем пользователи централизованной файловой системы, представляющей все каталоги от единого корня, не связанного с физическим носителем, часто даже не знают, на каком именно физическом устройстве содержатся те или иные данные.

Для выполнения своей функции, файловая система должна, как минимум, постоянно вести учет свободных и занятых блоков на диске, отслеживать принадлежность дисковых блоков к конкретным файлам и порядок следования блоков в файле.

Кроме того, современная файловая система должна быть устойчива к случайным сбоям, таким как внезапное выключение компьютера, или повреждение части данных на диске, другими словами, потеря нескольких блоков данных не должна приводить к разрушению всей файловой системы и, как следствие, потере всех данных.

Также во многих современных операционных системах интерфейс файловой подсистемы используется не только для доступа к данным на дисках, но и для всех операций ввода-вывода. Тогда взаимодействие

прикладных программ с клавиатурой, монитором, принтером или другим периферийным оборудованием выглядит как запись или чтение некоторых файлов.

Подсистема управления памятью в современной операционной системе решает следующие основные задачи:

- бесконфликтное распределение памяти между исполняющимися программами;
- защита памяти от несанкционированного доступа.

В любой современной операционной системе эти задачи полностью решаются средствами виртуальной памяти, механизмы работы которой будут весьма подробно рассмотрены в данном курсе через некоторое время.

Выполнение любой программы в рамках любой современной операционной системы начинается с создания процесса – специального объекта операционной системы, обеспечивающего выполнение программы, таким образом, подсистема управления процессами в конечном итоге отвечает за выполнение программ.

Подсистема управления процессами в современной операционной системе решает довольно обширный набор задач, которые удобно представить в виде диаграммы:



Рисунок 2 – Задачи подсистемы управления процессами

Таким образом, подсистема управления процессами решает два класса задач: создание и удаление процессов, и управление существующими процессами. Задача создания процесса сводится к размещению в памяти и инициализации структур данных, описывающих процесс. Задача удаления процесса зачастую сводится к освобождению памяти и ресурсов, удерживаемых процессом.

Управление процессами требует решения двух классов задач: задач планирования процессов и задач межпроцессного взаимодействия.

Подсистема планирования процессов решает две независимые задачи:

1. Определяет, какой из готовых к выполнению процессов, когда, и на какой интервал времени следует передать процессору для выполнения, т.е. выполняет собственно планирование процессов;
2. Выполняет переключение процессов, т.е. снимает с исполнения текущий процесс и ставит на исполнение запланированный процесс. Такое переключение процессов называют диспетчеризацией.

Заметим, что задача планирования является аппаратно независимой. Планирование процессов осуществляется исходя из степени важности процессов для системы или пользователя и временной статистики их исполнения, т.е. без учета каких-либо аппаратных особенностей компьютера. Задача диспетчеризации, наоборот, является аппаратно зависимой, т.к. переключение процессов связано с сохранением и загрузкой регистров конкретного процессора.

В рамках межпроцессного взаимодействия необходимо решать две задачи: задачу синхронизации процессов и задачу межпроцессной коммуникации.

Синхронизация процессов является одним из важнейших аспектов функционирования многозадачной операционной системы. Действительно, в многозадачной операционной системе часто возникает ситуация, когда некоторый процесс не может выполняться дальше, пока не будут завершены определенные действия другого процесса. Например, один процесс, назовем его процесс-писатель, помещает данные в некоторый буфер, а другой процесс – процесс-читатель, считывает данные из этого буфера. Очевидно, что процесс-писатель не должен помещать в буфер новые данные, пока процесс-читатель не считал ранее записанные данные. Как будет показано в нашем курсе далее, осуществить необходимую синхронизацию на уровне пользовательских процессов, без специальных средств операционной системы, не представляется возможным.

Что касается межпроцессной коммуникации, то каждый процесс выполняется в рамках собственного виртуального адресного пространства, закрытого для других процессов, поэтому межпроцессные коммуникации могут быть реализованы только при наличии специальной поддержки со стороны операционной системы.

## 2 АНАЛИЗ КОНФИГУРАЦИИ И ПРОИЗВОДИТЕЛЬНОСТИ КОМПЬЮТЕРА

Анализ конфигурации и производительности компьютера является важным этапом при управлении компьютерными системами. Этот анализ позволяет выявить проблемы, которые могут снижать производительность компьютера, и предоставляет информацию для принятия мер по улучшению производительности. Далее будут рассмотрены основные аспекты анализа конфигурации и производительности компьютера.

Конфигурация компьютера - это набор компонентов, которые определяют характеристики компьютера. Некоторые из основных компонентов, которые могут влиять на производительность компьютера, описаны далее.

Процессор является одним из самых важных компонентов компьютера, который определяет его производительность. Производительность процессора зависит от нескольких факторов, таких как частота процессора, количество ядер и кэш-память. Чем выше частота процессора, тем выше производительность компьютера. Также важно учитывать количество ядер и кэш-память, так как они могут повысить производительность при многозадачной работе.

Оперативная память (ОЗУ) является вторым важным компонентом компьютера, который влияет на его производительность. ОЗУ используется для временного хранения данных, к которым компьютер часто обращается во время работы. Чем больше ОЗУ, тем больше данных компьютер может хранить и обрабатывать одновременно, что повышает производительность.

Жесткий диск является основным устройством для хранения данных на компьютере. Его производительность зависит от нескольких факторов, таких как скорость вращения диска и скорость передачи данных. Чем быстрее вращается диск и чем выше скорость передачи данных, тем выше производительность.

Графический процессор (*GPU*) является компонентом, который отвечает за вывод графического контента на экран. Он может использоваться для обработки графических операций, таких как игры, видео и *3D*-моделирование. Чем выше производительность *GPU*, тем выше качество графики и скорость обработки.

Системная плата является основой компьютерной системы и обеспечивает связь между всеми компонентами компьютера. Она определяет тип и количество разъемов для подключения компонентов, таких как

процессор, ОЗУ, жесткий диск и другие устройства. Важно учитывать совместимость компонентов, так как некоторые компоненты могут быть несовместимы с определенными моделями системной платы.

Анализ производительности компьютера включает в себя оценку скорости работы различных компонентов компьютера и выявление проблем, которые могут влиять на производительность. Для анализа производительности компьютера можно использовать различные программы, такие как:

- программы для бенчмаркинга;
- утилиты для мониторинга;
- утилиты для очистки.

Программы для бенчмаркинга позволяют оценить производительность компьютера в целом или отдельных компонентов, таких как процессор, ОЗУ, жесткий диск и *GPU*. Они тестируют компоненты на различных нагрузках и показывают результаты в виде числовых значений, которые можно сравнить с другими компьютерами или ориентировочными показателями.

Утилиты для мониторинга позволяют отслеживать производительность компьютера в реальном времени. Они могут отображать информацию о загрузке процессора, ОЗУ, жестких дисков и других компонентов. Эта информация может быть полезна для выявления проблем, связанных с перегрузкой компонентов и их загрузкой.

Утилиты для очистки позволяют удалить ненужные файлы и программы, которые могут замедлять работу компьютера. Они также могут очищать реестр и удалять ошибки, которые могут влиять на производительность компьютера.

### 3 УПРАВЛЕНИЕ УСТРОЙСТВАМИ КОМПЬЮТЕРА

Внешние устройства подключаются к системному интерфейсу через специальные устройства - контроллеры (адаптеры). Каждый контроллер имеет в своем составе ряд программно-доступных регистров (как минимум имеет хотя бы регистр данных, регистр состояния и регистр управления).

Каждый контроллер имеет свой набор команд. Получив команду от процессора, выполняющего программу ввода-вывода, контроллер отрабатывает команду автономно, управляя внешним устройством через "малый" интерфейс между устройством и контроллером. Контроллер, отрабатывая принятую от процессора команду, пересылает во внешнее устройство свои команды, данные и читает из устройства состояния, данные (команды и состояния обычно представляют собой высокий или низкий уровень напряжения в соответствующей линии "малого" интерфейса, импульс или последовательность импульсов в соответствующей линии "малого" интерфейса). Кроме того, контроллер может выполнять ряд вспомогательных аппаратных функций инициируемых аппаратными сигналами или записью управляющей информации в его программно-доступный регистр (например, сброс по сигналу *RESET* или включение шпиндельного двигателя гибкого диска путем записи в регистр *3F2* контроллера гибкого диска управляющей информации).

Существуют простые контроллеры и более сложные (интеллектуальные) контроллеры, выполняющие более сложные аппаратные функции и команды. Процессор управляет внешним устройством, выполняя соответствующую программу ввода/вывода, где он с помощью команд *IN*, *OUT* (чтение порта, запись в порт) имеет доступ к программно-доступным регистрам контроллера. В регистр управления процессор записывает команду, из регистра состояния читает информацию о состоянии внешнего устройства и контроллера, в регистр данных записывает выводимые на устройство данные, или читает из регистра данных считываемую с устройства информацию.

Возможны два способа организации программного обмена с внешними устройствами: обмен с опросом готовности устройства, обмен по запросам на обслуживание устройства (запросам на маскируемое прерывание).

При первом способе обмена, выполняя программу ввода-вывода, процессор запустит в контроллере операцию и циклически читает регистр состояния контроллера. Он ожидает появления (в соответствующем разряде регистра состояния) признака готовности к обмену данными с процессором.

При появлении этого признака в регистре состояния процессор с помощью команды *OUT* осуществляет запись в регистр данных (при выводе на устройство), или выполняет чтение информации из регистра данных контроллера с помощью команды *IN* (при вводе с устройства). После чего запускает в контроллере следующую операцию и т. д.

При выполнении обмена по второму варианту, процессор, выполняя программу, и запустив на контроллере операцию, не ожидает готовности контроллера к обмену данными. Он продолжает выполнять полезную программу. Контроллер внешнего устройства в случае полной готовности к обмену данными с процессором выдает на соответствующую линию *IRQ* сигнал запроса на обслуживание. Процессор через выполнение процедуры прерывания по сигналу *IRQ* (в ее выполнении участвуют контроллер прерываний и контроллер шины) переходит на выполнение программы ввода/вывода для данного устройства. Эта программа осуществляет обмен данными через регистр данных контроллера, запускает в контроллере следующую операцию и т. д. Процессор возвращается к выполнению прерванной программы через выполнение команды *IRET* (возврат из прерывания), которой заканчивается программа ввода/вывода, и т. д.

Прямой доступ - это способ обмена между внешним устройством и динамической памятью, при котором управление операциями обмена по интерфейсу между регистром данных устройства и последовательно расположенными ячейками памяти осуществляет контроллер прямого доступа к памяти, а микропроцессор в это время свободен. Прямой доступ может осуществляться двумя способами: обычным *DMA* и *busmastering DMA*. В первом случае используется стандартный контроллер прямого доступа, который интегрирован в *ICH*. Во втором случае, функции аналогичные функциям контроллера выполняет аппаратура, расположенная в составе адаптера устройства (например, контроллер жестких дисков поддерживает режим *Bus Master*).



#### **4 ТЕОРЕТИЧЕСКОЕ ОБОСНОВАНИЕ РАЗРАБОТКИ ПРОГРАММНОГО ПРОДУКТА**

Разработка приложения, которое позволяет анализировать конфигурацию компьютера, производительность подсистем и управлять подключенными устройствами, является целесообразной, поскольку это позволяет пользователям получать дополнительные возможности для мониторинга и управления своим компьютером.

Одним из главных преимуществ такого приложения является возможность получать информацию о конфигурации компьютера, включая информацию о установленных компонентах, их производительности и связанных с ними проблемах. Это может быть особенно полезно для пользователей, которые хотят узнать больше о своих компьютерах или диагностировать проблемы с их работой.

Кроме того, такое приложение позволяет управлять подключенными устройствами, такими как принтеры, сканеры, флешки и другие устройства. Данная функция удобна для пользователей, которые работают с различными устройствами и хотят упростить их управление. Например, приложение может позволить пользователю быстро подключать и отключать устройства или управлять их параметрами, такими как яркость, контрастность и другие.

Данное приложение может обеспечить мониторинг производительности подсистем компьютера, что позволит пользователю быстро определить, какие компоненты работают наиболее эффективно, а какие нуждаются в замене или дополнительной настройке.

Приложение с ограниченным управлением устройствами может быть особенно полезно для организаций, которые хотят ограничить доступ к устройствам или определенным функциям компьютера для безопасности или производительности.

Таким образом, разработка приложения, которое позволяет анализировать конфигурацию и производительность компьютера, управлять подключенными устройствами и ограничивать доступ к ним, является необходимой для того, чтобы предоставить пользователям дополнительные возможности для мониторинга и управления своим компьютером.

## ЗАКЛЮЧЕНИЕ

Анализ конфигурации и подключенных устройств, а также производительности подсистем компьютера является важным этапом в обеспечении стабильной и безопасной работы компьютера. Подключение некоторых устройств может привести к ограниченному управлению компьютером, поэтому необходимо тщательно проанализировать все подключенные устройства и их влияние на работу системы.

В рамках данного курсового проекта для анализа конфигурации и подключенных устройств было разработано приложение, реализующее данные функции. Также проанализирована производительность подсистем компьютера, таких как процессор, оперативная память, жесткий диск и другие. Это поможет определить возможные проблемы и узкие места в работе системы, а также выбрать наиболее подходящие компоненты для обновления.

Ограниченное управление устройствами может быть связано с различными факторами, такими как наличие ограничений в настройках операционной системы или настройках *BIOS*. Для решения проблем с ограниченным управлением были проанализированы настройки.

Таким образом, анализ конфигурации и подключенных устройств, производительности подсистем компьютера и решение проблем с ограниченным управлением устройств являются важными этапами в обеспечении стабильной и безопасной работы компьютера.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- [1] Курсы операционных систем / А. С. Ткачев, А. А. Астапенко, М. И. Кузнецов. – СПб.: БХВ-Петербург, 2018. – 656 с.
- [2] Томпсон Э. Л., Стил Д. М. Операционная система UNIX. – М.: Техносфера, 2018. – 432 с.
- [3] Таненбаум Э. С. Современные операционные системы. – СПб.: Питер, 2018. – 1120 с.
- [4] Стивенс Р., Феннер Б., Рудофф Э. UNIX. Разработка сетевых приложений. – СПб.: Питер, 2018. – 1008 с.
- [5] Procomputer [Электронный ресурс]. – Режим доступа : <http://procomputer.ru/operatsionnye-sistemy>. – Дата доступа: 24.02.2023.
- [6] Megabook [Электронный ресурс]. – Режим доступа : <https://megabook.ru/article/Конфигурация%20компьютера>. – Дата доступа: 24.02.2023.

**ПРИЛОЖЕНИЕ А**  
**(обязательное)**  
**Листинг программного кода**