

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования

**«Южно-Уральский государственный университет
(национальный исследовательский университет)»**
Высшая школа электроники и компьютерных наук
Кафедра системного программирования

ДОПУСТИТЬ К ЗАЩИТЕ

Заведующий кафедрой, д.ф.-м.н.,
профессор

_____ Л.Б. Соколинский

« ____ » _____ 2024 г.

**Разработка утилиты по сбору информации
об участниках домена**

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
ЮУрГУ – 09.03.04.2024.308-368.ВКР

Научный руководитель,
доцент кафедры СП, к.т.н.
_____ М.В. Сухов

Автор работы,
студент группы КЭ-433
_____ Г.А. Трубецкой

Ученый секретарь
(нормоконтролер)
_____ И.Д. Володченко
« ____ » _____ 2024 г.

Челябинск, 2024 г.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования

**«Южно-Уральский государственный университет
(национальный исследовательский университет)»**
Высшая школа электроники и компьютерных наук
Кафедра системного программирования

УТВЕРЖДАЮ

Зав. кафедрой СП

_____ Л.Б. Соколинский

29.01.2024 г.

ЗАДАНИЕ

на выполнение выпускной квалификационной работы бакалавра

студенту группы КЭ-433

Трубецкому Георгию Александровичу,

обучающемуся по направлению

09.03.04 «Программная инженерия»

- 1. Тема работы** (утверждена приказом ректора от 22.04.2024 г. № 764-13/12)
Разработка утилиты по сбору информации об участниках домена.
- 2. Срок сдачи студентом законченной работы:** 03.06.2024 г.
- 3. Исходные данные к работе**
 - 3.1. Desmond B., Richards J., Allen R. Designing. Deploying and Running Active Directory. // O'Reilly Media; 5 edition, 2013. – 735 p.
 - 3.2. Francis D. Mastering Active Directory. Design, deploy and protect Active Directory Domain Services for Windows Server 2022. // Packt; 3 edition, 2021. – 780 p.
 - 3.3. Holmes L. Powershell Cookbook. // O'Reilly Media; 4 edition, 2021. – 1034 p.
- 4. Перечень подлежащих разработке вопросов**
 - 4.1. Провести анализ предметной области.
 - 4.2. Спроектировать и реализовать утилиту по сбору информации об участниках домена.
 - 4.3. Организовать тестовую среду.
 - 4.4. Провести тестирование разработанного продукта.
- 5. Дата выдачи задания:** 29.01.2024 г.

Научный руководитель,

доцент кафедры СП, к.т.н.

М.В. Сухов

Задание принял к исполнению

Г.А. Трубецкой

ГЛОССАРИЙ

1. *AD DS (Active directory domain services)* – это коммерческая служба каталогов компании Microsoft [15].
2. *Azure* – это облачная платформа компании Microsoft [16].
3. *CIDR (Classless inter-domain routing)* – метод IP-адресации, позволяющий гибко управлять пространством IP-адресов, не используя рамки классовой адресации [17].
4. *CLI (Command line interface)* – это командный интерфейс пользователя [18].
5. *CLR (Common Language Runtime)* – это виртуальная машина, которая исполняет программы, написанные на языках, совместимых с платформой .NET [19].
6. *CN (Common name)* – это поле в цифровых сертификатах (SSL/TLS), которое используется для указания доменного имени или имени хоста [20].
7. *CPU (Central processing unit)* – это электронный блок или интегральная схема, исполняющая машинные инструкции [21].
8. *DC (Domain controller)* – это сервер, управляющий доступом к сетевым ресурсам в рамках одного домена [22].
9. *DHCP (Dynamic host configuration protocol)* – это протокол прикладного уровня модели TCP/IP, который служит для назначения IP-адреса клиенту [23].
10. *DIT (Direction information tree)* – это каталог информационной сети, структура которого представлена в виде дерева [24].
11. *DN (Distinguished name)* – это имя, которое уникально идентифицирует каждую запись каталога [25].
12. *DNS (Domain name system)* – это компьютерная распределенная система для получения информации о доменах [26].
13. *Ethernet* – это протокол, который управляет процессами передачи данных в локальной сети [27].

14. *Exchange Server* – это коммерческий почтовый сервер компании Microsoft [28].

15. *FQDN (Fully qualified domain name)* – это доменное имя, которое не имеет неоднозначностей в определении [29].

16. *GUI (Graphical user interface)* – это графический интерфейс пользователя [30].

17. *GUID (Global unique identifier)* – это неизменяемый уникальный 128-битный идентификатор [31].

18. *ICMP (Internet control message protocol)* – это сетевой протокол, входящий в стек протоколов TCP/IP, предназначенный для обмена информацией о неполадках в сети [32].

19. *IP (internet protocol)* – это маршрутизируемый протокол сетевого уровня стека TCP/IP [33].

20. *IP-address* – это уникальный адрес, который идентифицирует устройство в сети [34].

21. *LDAP (Lightweight directory access protocol)* – это протокол прикладного уровня для доступа к службе каталогов [35].

22. *NET* – это платформа с открытым исходным кодом для создания приложений, которые могут работать в любой операционной системе [36].

23. *NIC (Network interface controller)* – это устройство, предназначенное для соединения компьютера с сетью [37].

24. *NOS (Network operation system)* – это сетевая операционная система [38].

25. *OS (Operating system)* – это программное обеспечение, управляющее аппаратным обеспечением компьютера [39].

26. *RDN (Relative distinguished name)* – относительное уникальное имя объекта службы каталогов [40].

27. *Shell* – это компьютерная программа (оболочка), которая предоставляет доступ к сервисам операционной системы или другим программам [41].

28. *SQL Server* – это коммерческая СУБД компании Microsoft [42].

29. *Windows* – это группа семейств коммерческий проприетарных операционных систем корпорации Microsoft [43].

30. *XML (Extensible markup language)* – это формат текстового документа, на котором создается разметка данных и описание ее языков [44].

31. *АРМ (Автоматизированное рабочее место)* – это информационное, программное и техническое обеспечение рабочего места сотрудника [45].

32. *Виртуальная машина* – это программная и/или аппаратная система, эмулирующая аппаратное обеспечение компьютера и исполняющая программы для гостевой платформы на хосте [46].

33. *Дерево* – это иерархическая структура одного домена [47].

34. *Домен* – это основная административная единица в сетевой инфраструктуре предприятия [48].

35. *Командлет (cmdlets)* – это специализированная команда Microsoft Powershell, которая реализует какую-либо функциональность [49].

36. *Коммутатор* – это сетевое устройство, предназначенное для соединения нескольких узлов компьютерной сети в пределах одного или нескольких сегментов сети [50].

37. *Лес* – это набор из одного или более деревьев, которые передают друг другу информацию и имеют транзитивное доверие [51].

38. *Маршрутизатор* – сетевое оборудования, предназначенное для объединения сегментов сети и ее элементов, а также для передачи пакетов между ними [52].

39. *Репозиторий* – это хранилище, в котором хранятся и поддерживаются данные [53].

40. *Служба каталогов* – это сетевой сервис, представляющий собой централизованные средства управления ресурсами автоматизированной системы [54].

41. *Хост* – это компьютер, подключенный к локальной или глобальной сети [55].

ОГЛАВЛЕНИЕ

ГЛОССАРИЙ	3
ВВЕДЕНИЕ	7
1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ	9
1.1. Служба каталогов	9
1.2. Powershell	15
2. РАЗРАБОТКА УТИЛИТЫ	26
2.1. Определение требований	26
2.2. Архитектура утилиты.....	26
2.3. Определение директории для хранения файлов вывода	28
2.4. Сбор имен компьютеров домена.....	28
2.5. Проверка доступности компьютеров домена.....	29
2.6. Сбор информации о CPU	30
2.7. Сбор информации об операционной системе	31
3. ОРГАНИЗАЦИЯ ТЕСТОВОЙ СРЕДЫ.....	33
3.1. Выбор имен виртуальных машин	34
3.2. Определение имени домена	35
3.3. Соответствие виртуальных машин и гостевых ОС	36
3.4. Аппаратное обеспечение виртуальных машин.....	36
4. ТЕСТИРОВАНИЕ	51
4.1. Проверка совместимости 5 и 7 версий Powershell.	51
4.2. Проверка основных функций.....	51
4.3. Проверка второстепенных функций.....	53
4.4. Проверка собранной информации	54
ЗАКЛЮЧЕНИЕ	56
ЛИТЕРАТУРА.....	57
ПРИЛОЖЕНИЕ. Код разработанной утилиты	63

ВВЕДЕНИЕ

Актуальность темы

В современной России многие организации цифровизированы, что создает спрос на специалистов по обслуживанию компьютерного аппаратного и программного обеспечения. Системный администратор отвечает за поддержку и сопровождение автоматизированных рабочих мест (АРМ) и серверов организации, обеспечивая бесперебойную работу информационной инфраструктуры.

Одной из основных задач системного администратора является поддержание порядка на автоматизированных рабочих местах пользователей. Именно системный администратор отвечает за состояние аппаратного обеспечения рабочих станций и за программное обеспечение. Так, например, в зоне ответственности одного системного администратора может быть более сотни автоматизированных рабочих мест, которые находятся в разных удаленных местах, и за которыми работают сотрудники с разным графиком работы, сменами и перерывами на обед. Таким образом физическое обслуживание рабочих станций становится очень сложной задачей.

В качестве примера обслуживания рабочих станций можно считать сбор информации о них, если в данный момент отсутствует документация. Системный администратор должен иметь доступ к базе данных, отражающей всю актуальную информацию о рабочих станциях и серверах для упрощения задач администрирования. Например, с ее помощью системный администратор может определить, какие рабочие станции совместимы с определенной операционной системой или программным обеспечением, какие рабочие станции необходимо обновить или списать, а какие требуют немедленного технического обслуживания.

Программное обеспечение для удаленного централизованного сбора информации о рабочих станциях пользователей существуют, однако они либо являются коммерческими продуктами и стоят средств, либо требуют

специального оборудования и навыков для их развертывания и администрирования. К сожалению, в данный момент не существует решения, которое бы было бесплатным для коммерческого использования и было максимально простым в управлении и обслуживании.

Постановка задачи

Целью выпускной квалификационной работы является разработка утилиты по сбору информации об участниках домена. Для достижения поставленной цели необходимо решить следующие задачи.

1. Провести анализ предметной области.
2. Спроектировать и реализовать утилиту по сбору информации об участниках домена.
3. Организовать тестовую среду.
4. Провести тестирование разработанного продукта.

Структура и содержание работы

Работа состоит из введения, четырех глав, заключения и списка литературы. Объем работы составляет 65 страниц, объем списка литературы – 55 источников.

В первой главе «Анализ предметной области» рассмотрены и описаны технологии, с помощью которых была проведена работа.

Во второй главе «Разработка утилиты» был описан процесс разработки программного продукта.

В третьей главе «Организация тестовой среды» был представлен процесс развертывания виртуальной инфраструктуры для тестирования разработанного продукта.

В четвертой главе «Тестирование» было проведено тестирование разработанной утилиты.

В приложении содержится листинг кода разработанной утилиты.

В глоссарии находятся значения технических терминов.

1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1. Служба каталогов

Служба каталогов (Directory service) – это инструмент представления информации и ресурсов в иерархическом виде. В роли ресурсов могут выступать сведения о материальном обеспечении организации, сотрудников и сетевых служб [1].

Каждый из ресурсов принадлежит к одному или более классам. Класс позволяет определить тип ресурса, будь то компьютер или учетная запись, а также отвечает за набор свойств. Совокупность классов называется схемой. Схема описывает типы ресурсов для каждой из предметных областей.

Служба каталогов в компьютерных сетях, представленная программным продуктом, позволяет системному администратору взаимодействовать с информацией в службе каталогов, которая хранится в едином кластере серверов.

Active Directory Domain Services

Самой популярной службой каталогов является Active Directory Domain Services (AD DS), программный продукт корпорации Microsoft [1]. По этой причине утилита будет разработана именно для работы с AD DS.

Active Directory (AD) – это сетевая операционная система (Network Operating System – NOS), разработанная корпорацией Microsoft, которая является встроенным компонентом операционных систем семейства Windows Server, начиная с версии Windows 2000 [1].

Active Directory позволяет системным администраторам управлять информацией организации из центрального хранилища (репозитория), которое может быть распределено глобально. Как только информация о пользователях, группах, компьютерах, принтерах, приложениях, сервисах и т.п. была добавлена в Active Directory, она становится доступной во всей инфраструктуре предприятия для всех членов организации. Информация может быть структурирована по той же топологии, по которой построена организация [1].

Хранение и идентификация объектов

Информация, содержащаяся в Active Directory, представлена для пользователя в иерархическом виде, который напоминает файловую систему. Каждый путь ведет к какому-либо объекту. На структурном уровне существуют два типа объектов: контейнеры (containers) и не-контейнеры (non-containers или так называемые leaf-objects, или конечные объекты). Контейнеры могут хранить другие ассоциированные с ними объекты в локальной структуре данных. Конечные объекты – это самые крайние логические компоненты в локальной структуре данных, которые не могут иметь ассоциированных с ними дочерних объектов [1].

Данные в Active Directory представлены в иерархическом виде, однако хранятся они в базе данных в виде строк и столбцов. Direction Information Tree (DIT), который представляет информацию, является файлом базы данных Extensible Storage Engine (ESE). Таким образом ESE является DIT технологией [1].

На рисунке 1 представлены отношения контейнеров и конечных объектов. Корневой объект является доменом и имеет два дочерних объекта, которые также являются контейнерами. Контейнер «Отдел информационных технологий» также содержит два других контейнера. В контейнере «Группа администрирования» могут находиться конечные объекты: компьютеры, учетные записи, группы, принтеры и т.д.

Основным типом контейнера является Organizational Unit (OU), который может быть добавлен, удален, перемещен, делегирован администратором домена [1]. Контейнеры по умолчанию, в которые добавляются новые объекты службы каталогов, называются «Container». Они не могут быть добавлены, удалены, перемещены или делегированы.

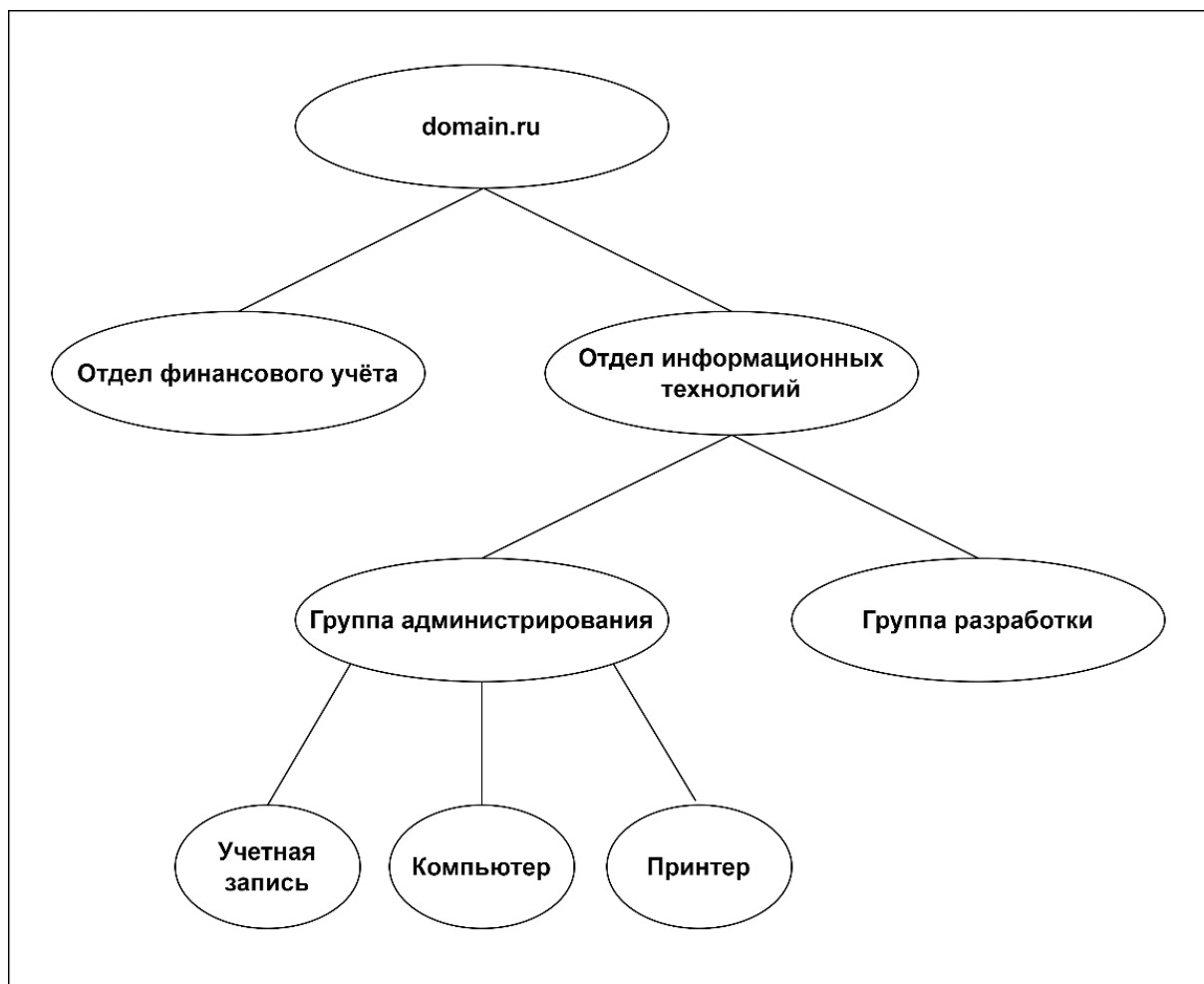


Рисунок 1 – Отношения контейнеров и конечных объектов

Уникально идентифицируемые объекты

Каждый объект в службе каталогов имеет свой уникальный идентификатор (Global unique identifier – GUID), который автоматически им назначается при добавлении объекта в базу данных. GUID – это 128-битное число, которое является реализацией universally unique identifier (UUID). GUID объекта присвоен ему до тех пор, пока объект не будет удален из базы данных, а перемещение в пределах DIT или изменение имени объекта не влияет на его изменение GUID. GUID также не будет изменен, если объект перемещается между доменами одного леса [1].

GUID объектов устойчивы и не изменяются, поэтому в случае изменения имени объекта или его перемещения запомнить этот объект по идентификатору является сложной задачей. Для ее облегчения разработан метод

определения имени объекта по иерархии – distinguished name (DN), который является основным для работы с объектами службы каталогов Active Directory [1].

Distinguished name (DN) – это иерархические пути, которые используются для идентификации объекта. DN определены в протоколе LDAP как стандарт, поэтому используется во всех службах каталогов, разработанных на основе протокола LDAP [1].

DN в Active Directory обычно предоставлены при помощи синтаксиса и правил, определенных стандартами LDAP.

На рисунке 2 показан DN корневого домена «susu.ru», каждая часть которого отделена запятой и имеет приставку со знаком равно и буквами «dc».

dc=susu,dc=ru

Рисунок 2 – Префикс и суффикс имени домена «susu.ru»

Приставка «dc» обозначает domain component и используется для определения части объектов в домене или приложении [1].

Relative distinguished name (RDN) – это имена, которые используются для уникальной идентификации объекта в родительском контейнере в директории [1].

На рисунке 3 показан пример RDN для учетной записи с именем Administrator, которая находится в контейнере Users в домене «susu.ru», где относительное имя учетной записи Administrator (cn=Administrator) является RDN.

cn=Administrator,cn=Users,dc=susu,dc=ru

Рисунок 3 – Пример RDN

RDN всегда должно быть уникальным в том контейнере, в котором оно находится. Невозможно иметь два объекта с одинаковым RDN в одном контейнере [1].

В таблице 1 показаны все ключи и соответствующие им типы атрибутов, согласно спецификации совместимости протокола LDAP RFC 2253 [2].

Таблица 1 – Ключи LDAP RFC 2253

Ключ (key)	Атрибут (attribute)
CN	Common name
L	Locality name
ST	State or province name
O	Organization name
OU	Organization unit name
C	Country name
STREET	Street address
DC	Domain Component
UID	User ID

Уровень домена

Логическая структура службы каталогов Active Directory построена на основе доменов [1].

Контроллер домена (Domain Controller – DC) может обслуживать только один домен. Невозможно разместить несколько доменов на одном контроллере домена.

Для примера рассмотрим домен «susu.ru», который был автоматически создан как корневой узел в иерархической структуре, названной деревом домена. Это буквально последовательность доменов, соединенных вместе в иерархическом представлении, где каждый из доменов использует смежные доменные имена. Если в домен «susu.ru» добавить поддомены (субдомены) «wiki», «info», «edu», тогда полные доменные имена будут иметь значения «edu.susu.ru», «wiki.susu.ru», «info.susu.ru». На рисунке 4 приведен пример логической структуры домена «susu.ru» и трех его поддоменов (субдоменов).

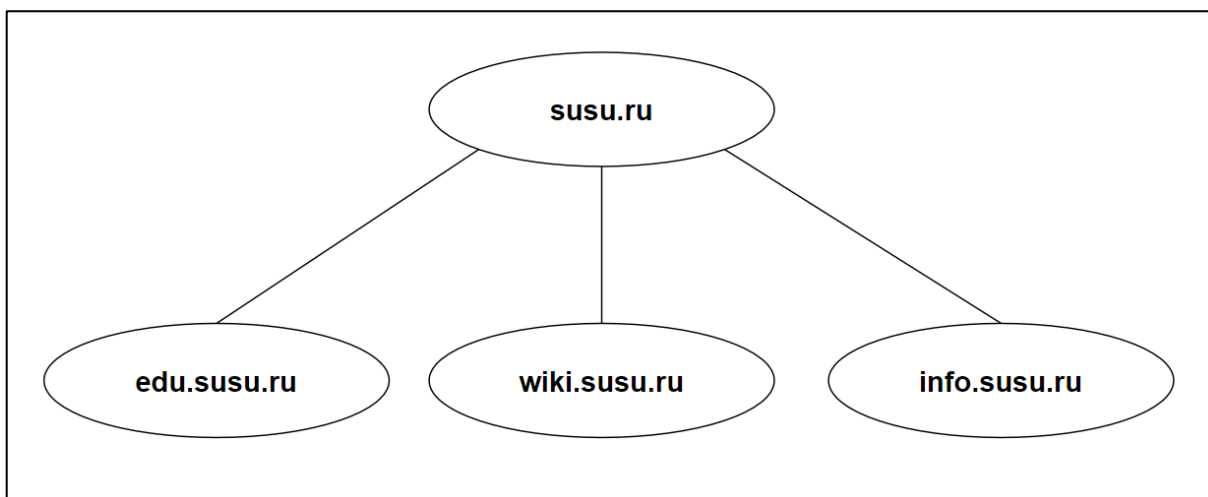


Рисунок 4 – Домен с субдоменами

Деревья домена позволяют иерархически управлять ресурсами и делегировать права доступа в доменной инфраструктуре. Ресурсы могут быть в составе нескольких доменов одного дерева, для этого существует транзитивное междоменное доверие (transitive trusts). В транзитивном доверии, если, к примеру, домен «edu.susu.ru» доверяет домену «wiki.susu.ru», а домен «wiki.susu.ru» доверяет домену «info.susu.ru», то домен «edu.susu.ru» и «info.susu.ru» будут автоматически доверять друг другу [1].

Уровень леса

Лес – это набор из одного или более деревьев. Деревья доменов в одном лесу передают друг другу информацию о содержимом контейнеров Schema и Configuration, если все деревья имеют транзитивное доверие. При создании нового домена необходимо добавить его в существующий лес или создать новый. Лес не имеет ограничения на количество деревьев, таким образом, лес – это самая большая доменная структура. Если при создании нового домена с ним создается и лес, то этот лес становится корневым [1].

В службе каталогов Active Directory невозможно удалить корневой лес домена, иначе будет удален весь лес со всеми сходящими в него доменами. Возможно изменить имя домена, однако невозможно изменить имя корневого домена в лесу. По умолчанию именем леса является имя корневого домена.

На рисунке 5 приведен пример леса, в котором находятся домены «susu.ru», «susu.edu» и «susu.net», а также транзитивные доверия между ними.

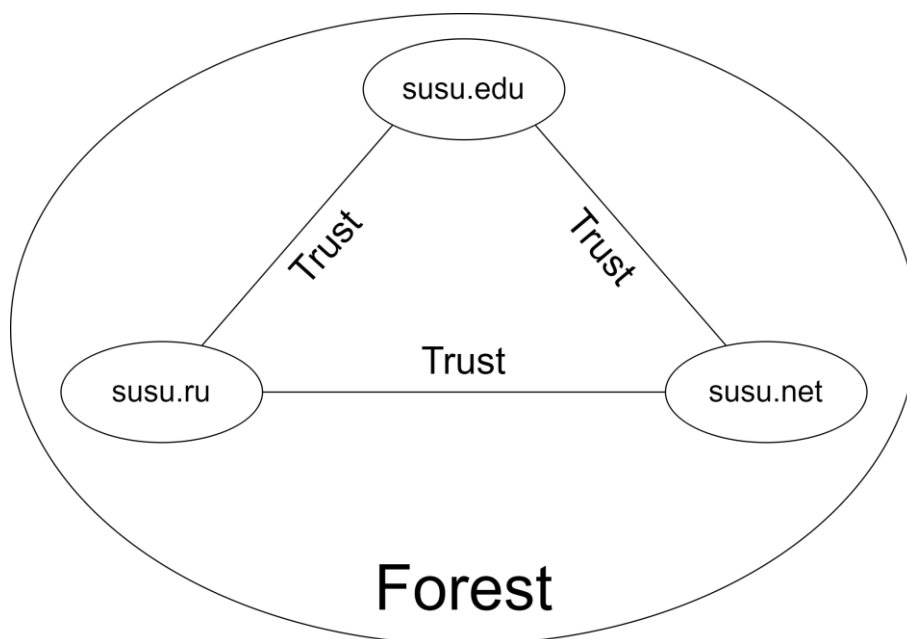


Рисунок 5 – Лес, состоящий из трех доменов

1.2. Powershell

Powershell – это решение, которое включает в себя оболочку (shell), одноименный скриптовый (сценарный) язык программирования и платформу управления конфигурацией [3].

Оболочка (shell) – это компьютерная программа, которая предоставляет доступ к сервисам операционной системы пользователю (человеку) или другим программам. Оболочкой может выступать как командный интерфейс (CLI – command line interface), так и графический интерфейс (GUI – graphical user interface). Командные оболочки требуют от пользователя наличия знаний команд и правил их написания (синтаксиса). Оболочка в Powershell принимает и возвращает не только текст, но и объекты .NET [3].

Powershell является скриптовым языком, и в основном используется для автоматизации задач управления системами. Powershell построен на платформе .NET на основе среды CLR (Common Language Runtime).

Платформа управления конфигурацией (Automation platform) – это экосистема модулей, при помощи которых возможно управление и развертывание программного обеспечения и технологий корпорации Microsoft, таких как Azure, Windows, Exchange, SQL и других [3].

Для разработки утилиты был выбран именно скриптовый язык программирования Powershell, поскольку он обладает следующими преимуществами.

1. Powershell работает со стандартными командами и приложениями операционных систем семейства Windows.

2. Команды Powershell представлены в отличающемся от других команд типе. Команды называются командлетами (cmdlets) и предоставляют Verb-Noun синтаксис.

3. Powershell поддерживает работу с объектами. Работа со структурируемыми объектами напрямую гораздо производительнее, чем работа со строками текста.

4. Powershell напрямую предназначен для системных администраторов инфраструктуры под управлением продуктов Microsoft.

5. Powershell поддерживает поиск. Используя определенные командлеты, пользователь может получить практически любую информацию о любых объектах системы.

Powershell поддерживает работу с другими технологиями, такими как .NET, COM, WMI, AD DS и другими.

Информация из Powershell может быть представлена в различных форматах, таких как CSV, JSON и XML.

Powershell позволяет работать с этими форматами данных с помощью специальных командлетов, таких как Import-Csv, ConvertTo-Json, ConvertTo-Xml и других. Это существенно упрощает работу с различными типами данных в сценариях Powershell и позволяет эффективно обмениваться информацией между различными приложениями и сервисами.

Версии Powershell

В таблице 2 приведены примеры основных версий Powershell и их описание [4].

Таблица 2 – Обзор версий Powershell

Версия Powershell	Примечание
1.0	Может быть установлен вручную, начиная с Windows Server SP1 и Windows XP
2.0	Предустановлен в Windows Server 2008 R2 и Windows 7
3.0	Предустановлен в Windows Server 2012 и Windows 8
4.0	Предустановлен в Windows Server 2012 R2 и Windows 8.1
5.0	Предустановлен в Windows 10 RTM и способен автоматически обновиться до версии 5.1 через службу Windows Update
5.1	Предустановлен в Windows 10 (начиная с 1709) и Windows Server 2016
Core 6.0 и 6.1	Добавлена поддержка MacOS, CentOS, RHEL, Debian, Ubuntu, openSUSE.
7.0 и выше	Является последней версией на текущий момент.

Согласно таблице 2, последней версией PowerShell является 7, именно на ее основе будет разработана утилита. Также необходима проверка разработанной утилиты на совместимость с 5 версией, так как она идет по умолчанию в современных версиях операционных систем Windows.

Модуль Powershell Active Directory

Модуль Powershell Active Directory позволяет взаимодействовать с объектами службы каталогов Active Directory и состоит из группы командлетов [5]. Модуль PowerShell Active Directory значительно упрощает автоматизацию, позволяя быстро и эффективно выполнять повседневные операции в области управления доменами, лесами и объектами Active Directory. В таблице 3 показаны некоторые из командлетов и их описание.

Таблица 3 – Примеры командлетов PowerShell

Командлет	Описание
Get-ADComputer	Получает информацию о компьютере домена
Get-ADUser	Получает информацию об учетной записи домена
New-ADUser	Создает новую учетную запись домена
Remove-ADUser	Удаляет существующую учетную запись домена
Set-ADDomain	Изменяет параметры существующего домена

Командлет `Get-ADComputer`

Командлет `Get-ADComputer` позволяет получить CN (имена) одного или более компьютеров, входящих в состав домена AD DS [6].

Параметры командлета, которые могут быть использованы при разработке программного продукта, представлены далее.

1. `AuthType` – определяет используемый метод аутентификации. Допустимыми значениями для этого параметра являются `Negotiate` (0) и `Basic` (1). По умолчанию используется метод аутентификации `Negotiate`. Для того, чтобы использовать метод аутентификации `Basic`, необходимо соединение SSL (Secure Sockets Layer).

2. `Credential` – определяет учетную запись пользователя, от имени которого будет выполнен командлет. По умолчанию используется учетная запись пользователя, который уже вошел в систему. Для того, чтобы указать этот параметр, необходимо ввести имя пользователя в виде «User», «Domain\User» или указать объект `PSCredential`. Если учетной записи назначен пароль, то командлет после вызова запросит пароль. Учетная запись должна обладать правами на уровне каталога.

3. `Filter` – определяет параметры, по которым будут отобраны и возвращены объекты Active Directory. Строка использует синтаксис Windows PowerShell Expression Language.

4. `Identity` – определяет объект (компьютер) Active Directory по предоставляемым параметрам. Параметрами могут быть DN, GUID, SID, SAM.

5. `LDAPFilter` – определяет LDAP-запрос в виде строки, которая используется для сбора объектов AD. Этот параметр используется для запуска существующих запросов LDAP.

6. `Partition` – определяет DN раздела AD, в котором хранятся объекты. DN должен быть одним из существующих контекстных имен в базе данных контроллера домена.

7. `Properties` – определяет свойства получаемого объекта. Этот параметр необходимо использовать в случае, когда свойства объекта не включены в набор свойств по умолчанию.

8. `ResultPageSize` – определяет количество объектов, включаемых в консольный вывод на одной странице выполненного запроса AD DS. Значение по умолчанию – 256 объектов на страницу.

9. `ResultSetSize` – определяет максимальное количество объектов, возвращаемых при выполнении запроса AD DS. По умолчанию используется значение `$Null`, которое возвращает все объекты.

10. `SearchBase` – определяет путь для поиска в Active Directory. Когда командлет запущен на диске, который является носителем базы данных AD, значением по умолчанию является путь к диску, с которого был запущен командлет. Этот параметр необходим, чтобы командлет смог быть запущен с диска, который не является носителем базы данных AD.

11. `SearchScope` – определяет область поиска объектов в AD. Принимаемыми параметрами являются `Base` (0), который направляет запрос поиска только по текущему пути к объекту, `OneLevel` (1), который направляет запрос на непосредственные дочерние элементы заданного пути или объекта, а также `SubTree` (2), который направляет запрос на все дочерние элементы.

12. `Server` – определяет экземпляр сервера AD (контроллера домена), к базе данных которого происходит подключение. Контроллер домена может быть задан по NetBIOS или FQDN с портом или без.

Командлет Get-WmiObject

Windows Management Instrumentation (WMI) – это инфраструктура и инструментарий для управления данными и операциями в операционных системах Windows. При помощи WMI возможно написание скриптов или приложений для автоматизации задач администрирования удаленных компьютеров. WMI полностью поддерживается корпорацией Microsoft [7].

Командлет `Get-WmiObject` позволяет получать информацию об экземплярах классов или информацию о доступных классах.

Параметры командлета, которые могут быть использованы при разработке программного продукта, представлены далее.

1. `Amended` – получает или задает значение, которое определяет, должны ли объекты, возвращаемые из WMI, содержать исправленную информацию. Обычно исправленная информация – это локализуемая информация, такая как описание объектов и свойств, которые закреплены за объектом WMI.

2. `AsJob` – запускает команду как фоновый процесс. Этот параметр можно использовать, к примеру, для вызовов, обработка которых занимает много времени. При использовании этого параметра команда возвращает объект, который представляет из себя фоновый процесс, и только затем отображает вывод командной строки.

3. `Authentication` – определяет уровень аутентификации, который используется при создании WMI соединения.

4. `Authority` – определяет метод аутентификации, который используется при подключении к WMI соединению. Допустимыми методами являются NTLM и Kerberos.

5. `Class` – задает имя класса WMI. Если имя класса задано, то командлет возвращает экземпляры заданного WMI класса.

6. `ComputerName` – задает имя целевого компьютера, над которым будет произведена операция. Имя целевого компьютера указывается CN или FQDN (если удаленный компьютер находится в другом домене) или NetBIOS, а также возможно обращение по IP-адресу. По умолчанию обращение происходит к локальному компьютеру. Имя локального компьютера можно задать по CN, точкой или строкой `localhost`. При обращении к удаленному компьютеру учетная запись, от имени которой была выполнена команда, должна иметь соответствующие разрешения для доступа к информации.

7. `Credential` – определяет учетную запись, от имени которой будет выполнена команда. Имя учетной записи может быть задано в виде «User», «Domain\User» или «User@Domain.ru». Также допустим ввод объекта `PSCredential`, к примеру объект, который возвращается командлетом с именем `Get-Credential`. При введении учетной записи пользователя оболочка предложит ввести пароль.

8. `DirectRead` – определяет, был ли запрошен прямой доступ к поставщику WMI для указанного класса без учета его базового или производных классов.

9. `EnableAllPrivileges` – подключает все привилегии текущего пользователя до того, как команда выполнит вызов WMI.

10. `Filter` – определяет условия и параметры, по которым будут отобраны запрашиваемые объекты. Использует синтаксис языка запросов WMI (WQL).

11. `Impersonation` – определяет уровень выполнения роли при вызове команды.

12. `List` – получает все имена классов в репозитории имен WMI, которое определяется параметром `Namespace` (пространством имен).

13. `Locale` – определяет предпочитаемую локализацию объектов WMI. Задается в формате `MS_<LCID>`.

14. `Namespace` – при использовании с параметром `Class`, этот параметр определяет репозиторий имен WMI.

15. `Property` – определяет свойства класса WMI, из которого командлет считывает информацию. Значением параметра являются имена свойств.

16. `Query` – запускает определенное WMI Query Language (WQL) состояние.

17. `Recurse` – анализирует заданное пространство имени и все другие пространства имен на совпадение имени класса, который задан в другом параметре `Class`.

18. `ThrottleLimit` – определяет максимальное количество операций WMI, которые могут быть выполнены одновременно. Этот параметр активен только, если другой параметр `AsJob` задан и используется при вызове команды.

Класс `Win32_Processor`

Класс `Win32_Processor` используется для сбора информации о CPU. В качестве собираемой информации могут выступать такие параметры CPU как: модель и производитель, количество физических, логических и включенных ядер, а также текущая и максимальная частоты работы CPU [8]. Иные параметры CPU являются второстепенными и не играют роли при просчете производительности, планировании вычислительных ресурсов или анализа совместимости с программным обеспечением.

Производитель и модель CPU в классе `Win32_Processor` представлены одним свойством `Name`, которое содержит в себе название модели и производителя процессора. Этот параметр является самым важным, т.к. зная модель, можно узнать всю основную информацию о процессоре на официальных интернет-ресурсах производителей CPU.

Количество физических ядер (`Physical Cores`) процессора. Этот параметр является второстепенным, т.к. информацию о количестве ядер текущего CPU можно узнать по его модели. За сбор информации о количестве физических ядер процессора отвечает свойство `NumberOfCores`.

Количество логических ядер (`Logical Cores`) процессора. Знать информацию о количестве логических ядер необходимо, чтобы системный администратор мог сопоставить количество физических и логических ядер, и тем самым определить, включена ли технология многопоточности у процессора,

если процессор ее поддерживает. За сбор информации о количестве логических ядер процессора отвечает свойство `NumberOfLogicalProcessors`.

Количество включенных ядер. Знать информацию о количестве включенных физических ядер необходимо, чтобы администратор мог сопоставить количество включенных и физических ядер, и тем самым определить, включены ли все ядра процессора. За сбор информации о количестве включенных ядер процессора отвечает свойство `NumberOfEnabledCore`.

Текущая частота – это текущая скорость работы процессора, измеряемая в МГц (MHz). Этот параметр является важным, так как процессор может регулировать свою скорость работы. Так, например, процессор может снижать частоту своей работы из-за превышения допустимых температур, что может говорить о том, что система охлаждения компьютера не справляется со своей задачей. За сбор информации о текущей частоте отвечает свойство `CurrentClockSpeed`.

Максимальная частота процессора – это максимальная скорость работы процессора, с которой он может работать, измеряется в МГц (MHz). Эта информация является критически важной для серверов, процессоры которых настроены на снижение максимальной частоты работы для снижения энергопотребления при простое. За сбор информации о текущей частоте отвечает свойство `MaxClockSpeed`.

В таблице 4 показано соответствие характеристик CPU (центрального процессора) и свойств класса `Win32_Processor`.

Таблица 4 – Соответствие параметров CPU и свойств класса `Win32_Processor`

Характеристика CPU	Свойство
Производитель и модель CPU	<code>Name</code>
Количество физических ядер	<code>NumberOfCores</code>
Количество логических ядер	<code>NumberOfLogicalProcessors</code>
Количество включенных ядер	<code>NumberOfEnabledCore</code>
Текущая частота	<code>CurrentClockSpeed</code>
Максимальная частота процессора	<code>MaxClockSpeed</code>

Класс Win32_OperatingSystem

Класс Win32_OperatingSystem используется для сбора информации об установленной ОС. В качестве собираемой информации могут выступать такие параметры ОС как: семейство и поколение, версия, разрядность, редакция, дата установки, корневой раздел, локализация и временная зона. Иные параметры ОС являются второстепенными и не играют роли в процессе администрирования рабочих станций.

Название (имя) ОС определяется параметром Name. Параметр является строкой, в которой определены семейство операционной системы, ее поколение, редакция, а также, предположительно, раздел, отвечающий за восстановление системы.

Версия (информация о выпуске) ОС определяется параметром Version, который содержит идентификатор платформы и версии, разделенные точкой.

Разрядность (bit) ОС определяется параметром OSArchitecture. Этот параметр очень важен, особенно для устаревающих версий ОС Windows. Однако все новейшие поколения операционных систем семейства Windows являются 64-разрядными. Иметь информацию о разрядности ОС крайне важно при установке программного обеспечения на устаревающие версии ОС семейства Windows.

Локализация определяется параметром OsLocale. Локализация – это текущий язык интерфейса ОС. Параметром является число в десятичной системе счисления, которое представлено в виде языкового кода.

Временная зона ОС определяется параметром CurrentTimeZone. Корректно заданная временная зона необходима для взаимодействия рабочей станции с контроллерами домена и доменными службами. Параметром является количество минут смещения временной зоны.

Дата установки ОС определяется параметром InstallDate, и она является важной, поскольку позволяет определить, когда именно ОС была

установлена. Параметром является строка, первые четыре цифры которой являются годом установки, следующие две – номером месяца, еще две – числом месяца, а далее часы, минуты, секунды.

Корневой системный раздел – это директория, в которую была установлена ОС. Определяется параметром `SystemDirectory`. Эта информация крайне важна при централизованном изменении содержимого каталогов на системных разделах компьютеров домена.

В таблице 5 показано соответствие характеристик ОС (операционной системы) и свойств класса `Win32_OperatingSystem`.

Таблица 5 – Соответствие характеристик ОС и свойств класса `Win32_OperatingSystem`

Характеристика ОС	Свойство
Имя	<code>Name</code>
Версия	<code>Version</code>
Разрядность (битность)	<code>OSArchitecture</code>
Локализация	<code>OSLocale</code>
Временная зона	<code>CurrentTimeZone</code>
Дата установки	<code>InstallDate</code>
Корневой системный раздел	<code>SystemDirectory</code>

Выводы по первой главе

В этой главе был проведен предметный анализ таких технологий, как служба каталогов `Active Directory` и командная оболочка `Powershell`. Служба каталогов является фундаментом разрабатываемой утилиты, поскольку именно на ее основе будет производиться сбор информации. Командная оболочка является инструментом, при помощи которого будет разработана утилита.

2. РАЗРАБОТКА УТИЛИТЫ

2.1. Определение требований

Функциональные требования к разрабатываемой утилите

1. Перед запуском утилиты у пользователя должна быть возможность задать директорию, в которой будут храниться файлы с информацией.
2. Результатом работы утилиты должны стать файлы, в которых храниться различная информация о компьютерах домена.
3. Ввод и вывод текстовой информации, а также интерфейс должны быть только на английском языке.

Нефункциональные требования к разрабатываемой утилите

1. Утилита должна быть совместима с 5 и 7 версиями PowerShell.
2. Вывод текстовой информации должен быть на английском языке.

2.2. Архитектура утилиты

Утилита состоит из модулей (функций), которые выполняются последовательно друг за другом. Каждая из функций выполняет определенную, уникальную роль, и не конфликтует с другими функциями. В таблице 6 показаны все функции утилиты.

Таблица 6 – Функции (модули) утилиты

Название функции	Назначение функции
func_Directory	Задаёт и проверяет директорию, в которую будут записываться файлы вывода.
func_AD_Computers_Names_Collection	Собирает CN компьютеров домена.
func_AD_Computers_Availability	Проверяет доступность компьютеров домена.
func_AD_Computers_CPU_Info_Collection	Собирает информацию об аппаратном обеспечении доступных компьютеров домена (а именно – CPU).
func_AD_Computers_OS_Info_Collection	Собирает информацию об операционных системах доступных компьютеров домена.

Модули утилиты разделены на две категории: основные и второстепенные. Основные функции – это те функции, которые отвечают за работу утилиты и сбор основной информации об участниках домена. Второстепен-

ные функции отвечают за сбор конкретной информации о параметрах участников домена. В роли параметров выступают характеристики подсистем программного и аппаратного обеспечения. Планируется разработка двух второстепенных модулей, которые будут отвечать за сбор информации о процессоре (CPU) и установленной операционной системе на клиенте соответственно. Эти параметры были выбраны в первую очередь по причине их наличия в единственном экземпляре на клиенте, что не допускает неоднородности собираемой информации. На рисунке 6 представлена диаграмма, отражающая все функции утилиты, их категории, а также последовательность выполнения модулей.

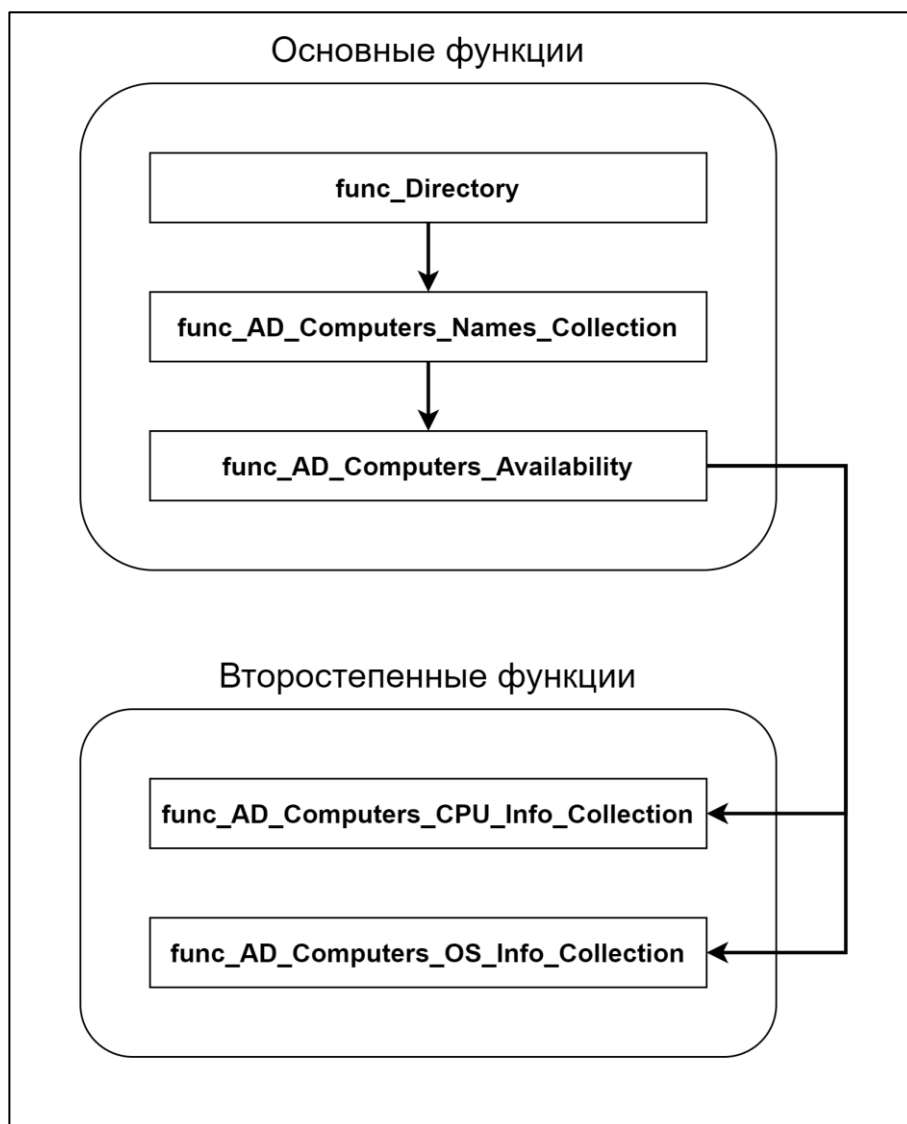


Рисунок 6 – Архитектура утилиты

2.3. Определение директории для хранения файлов вывода

Функция `func_Directory` позволяет пользователю определить директорию, в которой будут записаны или перезаписаны файлы вывода, содержащие информацию.

Директория задается в глобальной переменной `$global:Directory`.

Функция проверяет доступность заданной директории при помощи командлета `Test-Path`.

В случае успешной проверки запускается следующая функция `func_AD_Computers_Names_Collection`.

В случае провала проверки выводится сообщение об ошибке «The directory you specified does not exist. The utility start was interrupt.».

Значение по умолчанию равно «C:\».

2.4. Сбор имен компьютеров домена

Функция `func_AD_Computers_Names_Collection` собирает `Common Name` всех компьютеров домена. Опрос клиентов будет производиться по их доменным именам. В этой функции определены несколько глобальных переменных, представленных в таблице 7.

Таблица 7 – Переменные функции сбора имен

Переменная	Назначение
<code>\$global:AD_Computers_Names</code>	Содержит все имена компьютеров домена
<code>\$global:AD_Computers_All_Online</code>	Содержит логическое выражение, определяющее доступность всех компьютеров домена
<code>\$global:AD_Computers_Online_List</code>	Содержит список имен доступных компьютеров домена
<code>\$global:AD_Computers_Offline_List</code>	Содержит список имен недоступных компьютеров домена

У каждой из объявленных переменных есть свои значения по умолчанию, представленные в таблице 8.

Таблица 8 – Значения по умолчанию

Переменная	Значение по умолчанию
\$global:AD_Computers_Names	Get-ADComputer -Filter * -Properties Name Select-Object -ExpandProperty Name
\$global:AD_Computers_All Online	true
\$global:AD_Computers_Online_List	null
\$global:AD_Computers_Offline_List	null

Все имена компьютеров домена записываются в массив \$global:AD_Computers_Names при помощи командлета Get-ADComputer, в котором через фильтр были отобраны все имена по параметру Name, который содержит CN компьютера домена:

```
$global:AD_Computers_Names = Get-ADComputer -Filter * -Properties Name | Select-Object -ExpandProperty Name;
```

После сбора всех CN компьютеров домена вызывается следующая функция func_AD_Computers_Availability.

2.5. Проверка доступности компьютеров домена

Функция func_AD_Computers_Availability проверяет доступность компьютеров домена, фильтрует доступные компьютеры от недоступных, а также выводит пользователю список недоступных компьютеров в случае их наличия.

Исходные данные берутся из массива \$global:AD_Computers_Names, который был заполнен в функции func_AD_Computers_Names_Collection.

Проверка доступности осуществляется при помощи командлета Test-Connection, который отправляет эхо-запросы по протоколу ICMP. Размер пакета – 32 байта (bytes), количество пакетов – 1 (один), и также добавлен параметр -Quiet, чтобы запретить вывод информации о проверке доступности.

Эхо-запросы отправляются поочередно по всем компьютерам, находящимся в массиве \$global:AD_Computers_Names. Если ответ был получен и компьютер доступен, он добавляется в массив \$global:AD_Computers_

`Online_List`. Если ответ не был получен, то компьютер добавляется в массив `$global:AD_Computers_Offline_List`, а также переменная `$global:AD_Computers_All_Online` получает значение `false`.

Переменная `$global:AD_Computers_All_Online` – это глобальная логическая переменная, которая отвечает за доступность всех компьютеров домена. Если она равна значению `true`, то программа сразу приступает к процессу сбора информации, не информируя пользователя о недоступных компьютерах домена, т.к. они отсутствуют. Если значение равно `false`, то пользователю в консоль выводятся построчно имена недоступных компьютеров домена из массива `$global:AD_Computers_Offline_List`, а после предлагается ввести значение «yes» или «no», чтобы продолжить или прервать работу программы соответственно. Введенное пользователем значение проверяется после каждого ввода, и будет зациклено до тех пор, пока пользователь не введет значение «yes» или «no».

Если пользователь ввел значение «yes», то утилита запускает все остальные функции, отвечающие за сбор информации.

Если пользователь ввел значение «no», то утилита завершает работу.

2.6. Сбор информации о CPU

Функция `func_AD_Computers_CPU_Info_Collection` собирает информацию о CPU компьютеров, расположенных в переменной `$global:AD_Computers_Online_List` при помощи класса `Win32_Processor`.

Запись собираемой информации будет производиться в файл, имя которого определено в переменной `AD_Computers_CPU_info_file_name`. Имя файла по умолчанию – `AD_Computers_CPU_info.txt`.

Функция содержит массив `AD_Computer_CPU_parameters`, содержащий следующие названия параметров CPU: `Name`, `NumberOfCores`, `NumberOfLogicalProcessors`, `NumberOfEnabledCore`, `CurrentClockSpeed`, `MaxClockSpeed`. Всего параметров 6 единиц.

Запись собираемой информации в файл происходит построчно, по одной строке для каждого из компьютеров домена. Первая строка файла содержит заголовок, содержащий CN и названия параметров CPU, разделенных точкой с запятой. Все последующие строки файла – это CN и значения параметров CPU опрошенных компьютеров домена.

Заголовок файла заполняется при помощи командлета `Add-Content` путем записи строковой переменной `AD_Computer_CPU_parameters_string`, которая перед записью в файл изменяется путем добавления в конец строки имен параметров CPU, разделенных точкой с запятой.

Все параметры CPU компьютеров домена, а также их CN определяются в строковой переменной `AD_Computer_CPU_info`, в которую добавляются CN и значения параметров CPU компьютера домена, после чего строка записывается в файл и ей присваивается значение `null`, чтобы записать в переменную информацию о следующем компьютере домена из массива `AD_Computers_Online_List`.

2.7. Сбор информации об операционной системе

Функция `func_AD_Computers_OS_Info_Collection` собирает информацию о параметрах операционных систем доступных компьютеров домена, расположенных в переменной `$global:AD_Computers_Online_List` при помощи класса `Win32_OperatingSystem`.

Запись собираемой информации будет производиться в файл, имя которого определено в переменной `AD_Computers_OS_info_file_name`. Имя файла по умолчанию – `AD_Computers_OS_info.txt`.

Функция содержит массив `AD_Computer_OS_parameters`, содержащий следующие названия параметров ОС: `Name`, `Version`, `OSArchitecture`, `OSLanguage`, `CurrentTimeZone`, `InstallDate`, `SystemDirectory`. Всего параметров 7 единиц.

Запись собираемой информации в файл происходит построчно, по одной строке для каждого из компьютеров домена. Первая строка файла содержит заголовок, содержащий CN и названия параметров ОС, разделенных точкой с запятой. Все последующие строки файла – это CN и значения параметров ОС опрошенных компьютеров домена.

Заголовок файла заполняется при помощи командлета `Add-Content` путем записи строковой переменной `AD_Computer_OS_parameters_string`, которая перед записью в файл изменяется путем добавления в конец строки имен параметров ОС, разделенных точкой с запятой.

Все параметры ОС компьютеров домена, а также их CN определяются в строковой переменной `AD_Computer_OS_info`, в которую добавляются CN и значения параметров ОС компьютера домена, после чего строка записывается в файл и ей присваивается значение `null`, чтобы записать в переменную информацию о следующем компьютере домена из массива `AD_Computers_Online_List`.

Выводы по второй главе

В этой главе был представлен процесс разработки утилиты, а также описаны ее модули. Архитектура утилиты является модульной. Модули разделены на две категории: основные и второстепенные. Основные модули отвечают за функционирование самой утилиты, а также за процесс сбора имен участников домена и проверки их доступности. Второстепенные модули отвечают за процесс сбора информации о каждой из конкретных подсистем каждого из участников домена. В данный момент разработано только два второстепенных модуля, отвечающих за сбор информации о процессоре и установленной операционной системе клиента соответственно.

3. ОРГАНИЗАЦИЯ ТЕСТОВОЙ СРЕДЫ

Обзор клиентских и серверных операционных систем

В таблицах 9 и 10 показаны клиентские и серверные операционные системы семейства Windows.

В роли гостевых операционных систем виртуальных машин для тестовой среды были выбраны именно те операционные системы, поддержка которых заканчивается не раньше 2024 года.

Таблица 9 – Версии клиентской ОС Windows

Операционная система	Разрядность (битность)	Редакции	Конец поддержки
Windows 11	X64 (only)	Home, Pro, Enterprise, Education	неизвестно
Windows 10	x32-64	Home, Pro, S, Enterprise, Education	14.10.2025

В тестовой среде не будут участвовать версии Home и их производные, так как они не имеют возможности интеграции со службами каталогов AD DS. В тестовой среде также не будут участвовать версии S, Enterprise, Education и их производные, так как они не доступны для ознакомления и могут быть получены только через партнеров Microsoft.

Таблица 10 – Версии серверной ОС Windows Server

Операционная система	Разрядность (битность)	Редакции	Конец поддержки
Windows Server 2022	x64	Standard, Enterprise, Essentials	13.10.2026

В тестовой среде не будет участвовать версия Essentials, т.к. она не доступна для ознакомления.

В таблице 11 показан итоговый выбор операционных систем и их редакций, которые будут участвовать в тестировании.

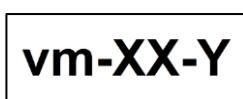
Таблица 11 – Версии выбранных операционных систем

Операционная система	Разрядность	Редакции
Windows 10	x32/x64	Pro
Windows 11	x64	Pro
Windows Server 2022	x64	Standard, Enterprise

3.1. Выбор имен виртуальных машин

Перед созданием виртуальных машин для тестового стенда необходимо определиться с их именами. Следуя лучшим практикам наименования виртуальных машин VMWare [14], в качестве их имен будут использоваться FQDN. Правильный выбор имен виртуальных машин позволит лучше ориентироваться в виртуальной среде, а также упростит процесс управления виртуальными ресурсами и повысит их уровень организации и согласованности. Выбор правильных имен для виртуальных машин может повысить эффективность управления виртуальными ресурсами, сделать их более распознаваемыми и упростить взаимодействие с другими элементами ИТ-инфраструктуры.

На рисунке 7 представлен порядок определения имен виртуальных машин: «vm» – показывает, что объект – виртуальная машина, «-» – разделители, «XX» – две цифры, которые обозначают версию Windows, установленную в единственном экземпляре на виртуальный жесткий диск, присоединенный к этой машине, «Y» – число, которое является порядковым номером различных виртуальных машин, на которую установлена одна и та же операционная система, но с различными ее редакциями или аппаратным обеспечением.



vm-XX-Y

Рисунок 7 – Представление имен виртуальных машин

В таблице 12 показано соответствие гостевых ОС заданным значениям.

Таблица 12 – Соответствие операционных систем заданным значениям

Значение	Соответствие
10	Windows 10
11	Windows 11
22	Windows Server 2022

3.2. Определение имени домена

Выбор имени домена очень важен, так как неправильный его выбор не приведет к сбою системы в целом, но создаст значительные ограничения при развертывании инфраструктуры и ее интеграции с другими сервисами.

Доменное имя должно содержать суффикс и префикс, разделенные точкой.

Суффикс при развертывании нового корневого домена обычно отражает имя организации. В нашем случае будут использоваться фамилия и инициалы автора работы – trubetskoyga.

Префикс же должен отражать имя регистратора домена, который будет обслуживать наш домен в сети Интернет. Поскольку ЮУрГУ – это университет, расположенный на территории Российской Федерации, будет использован префикс «ru».

На рисунке 8 показано разделение домена trubetskoyga.ru.



Рисунок 8 – Префикс и суффикс имени домена

Зная имя домена и имена будущих членов домена, возможно определить полные доменные имена виртуальных машин. Соответствие CN (простых имен) и FQDN (полных доменных имен) виртуальных машин показано в таблице 13.

Таблица 13 – Соответствие CN и FQDN виртуальных машин

VM name (CN)	VM name (FQDN)
vm-10-1	vm-10-1.trubetskoyga.ru
vm-10-2	vm-10-2.trubetskoyga.ru
vm-10-3	vm-10-3.trubetskoyga.ru
vm-10-4	vm-10-4.trubetskoyga.ru
vm-11-1	vm-11-1.trubetskoyga.ru
vm-11-2	vm-11-2.trubetskoyga.ru
vm-22-1	vm-22-1.trubetskoyga.ru
vm-22-2	vm-22-2.trubetskoyga.ru

3.3. Соответствие виртуальных машин и гостевых ОС

Для тестирования разработанной утилиты необходимо большое разнообразие гостевых операционных систем у клиентов домена.

Проанализировав таблицы 12 и 13, можно определить, какая версия и редакция гостевой ОС будет установлена на виртуальные машины. В таблице 14 показаны параметры гостевых операционных систем виртуальных машин.

Таблица 14 – Соответствие имен виртуальных машин и версий гостевых операционных систем

Имя VM	ОС	Версия (редакция)	Разрядность
vm-10-1.trubetskoyga.ru	Windows 10	Pro	x32
vm-10-2.trubetskoyga.ru	Windows 10	Pro	x64
vm-10-3.trubetskoyga.ru	Windows 10	Pro for workstation	x32
vm-10-4.trubetskoyga.ru	Windows 10	Pro for workstation	x64
vm-11-1.trubetskoyga.ru	Windows 11	Pro	x64
vm-11-2.trubetskoyga.ru	Windows 11	Pro	x64
vm-22-1.trubetskoyga.ru	Windows Server 2022	Standard	x64
vm-22-2.trubetskoyga.ru	Windows Server 2022	Enterprise	x64

3.4. Аппаратное обеспечение виртуальных машин

Для тестирования разработанной утилиты необходимо большое разнообразие аппаратных характеристик у клиентов домена.

Согласно минимальным системным требованиям для Windows Server 2022 [9], необходимо как минимум 2 GB оперативной памяти и 32 GB дискового пространства (вне зависимости от редакции). Для каждой виртуальной машины назначен различный объем аппаратных ресурсов, который больше или равен минимальному.

Согласно минимальным системным требованиям для Windows 10 [10], для 32-битной версии необходимо как минимум 1 GB RAM и 16 GB дискового пространства, а для 64-битной 2 GB RAM и 32 GB дискового пространства.

Согласно минимальным системным требованиям для Windows 11 [11], необходимо как минимум 4 GB RAM и 64 GB дискового пространства.

Аппаратные характеристики некоторых виртуальных машин могут повторяться, а объем назначенной оперативной памяти и виртуального жесткого диска могут не соответствовать своим физическим аналогам по причине того, что физические аппаратные ресурсы гипервизора расходовались экономно ради большего количества и разнообразия одновременно работающих виртуальных машин.

Каждой виртуальной машине назначено по одному виртуальному жесткому диску по причине того, что место на физическом диске гипервизора ограничено. Аппаратные ресурсы виртуальных машин приведены в таблице 15, а параметры виртуальных жестких дисков виртуальных машин приведены в таблице 16.

Таблица 15 – Аппаратные ресурсы виртуальных машин

VM name	vCPU count	RAM (GB)	Disk space (GB)
vm-10-1.trubetskoyga.ru	1	1	20
vm-10-2.trubetskoyga.ru	4	2	40
vm-10-3.trubetskoyga.ru	4	2	30
vm-10-4.trubetskoyga.ru	8	4	60
vm-11-1.trubetskoyga.ru	2	4	70
vm-11-2.trubetskoyga.ru	4	6	80
vm-22-1.trubetskoyga.ru	2	2	40
vm-22-2.trubetskoyga.ru	8	4	60

Таблица 16 – Параметры виртуальных жестких дисков виртуальных машин

VM name	VHD space	VHD name	VHD format	VHD type
vm-10-1.trubetskoyga.ru	20	vm-10-1-vhd	vhdx	fixed-size
vm-10-2.trubetskoyga.ru	40	vm-10-2-vhd	vhdx	fixed-size
vm-10-3.trubetskoyga.ru	30	vm-10-3-vhd	vhdx	fixed-size
vm-10-4.trubetskoyga.ru	60	vm-10-4-vhd	vhdx	fixed-size
vm-11-1.trubetskoyga.ru	70	vm-11-1-vhd	vhdx	fixed-size
vm-11-2.trubetskoyga.ru	80	vm-11-2-vhd	vhdx	fixed-size
vm-22-1.trubetskoyga.ru	40	vm-22-1-vhd	vhdx	fixed-size
vm-22-2.trubetskoyga.ru	60	vm-22-2-vhd	vhdx	fixed-size

Виртуальная локальная сеть

Каждая из виртуальных машин имеет виртуальный сетевой адаптер, который подключен к внешнему виртуальному коммутатору с именем `evs`.

На рисунке 9 изображена схема виртуальной локальной сети, из которой видно, что все восемь виртуальных машин имеют виртуальный сетевой адаптер, который подключен к внешнему виртуальному коммутатору. Все виртуальные машины находятся в одной виртуальной локальной сети IPv4 `192.168.1.0/24`. Адрес сетевого адаптера гостевой операционной системы виртуальной машины `vm-22-1.trubetskoyga.ru` является статическим и также входит в сеть `192.168.1.0/24`. Сетевые адреса всех остальных узлов сети являются динамическими, их сетевые интерфейсы получают адреса от DHCP сервера `vEthernet`, виртуального сетевого коммутатора, которым является физическое устройство маршрутизатор.

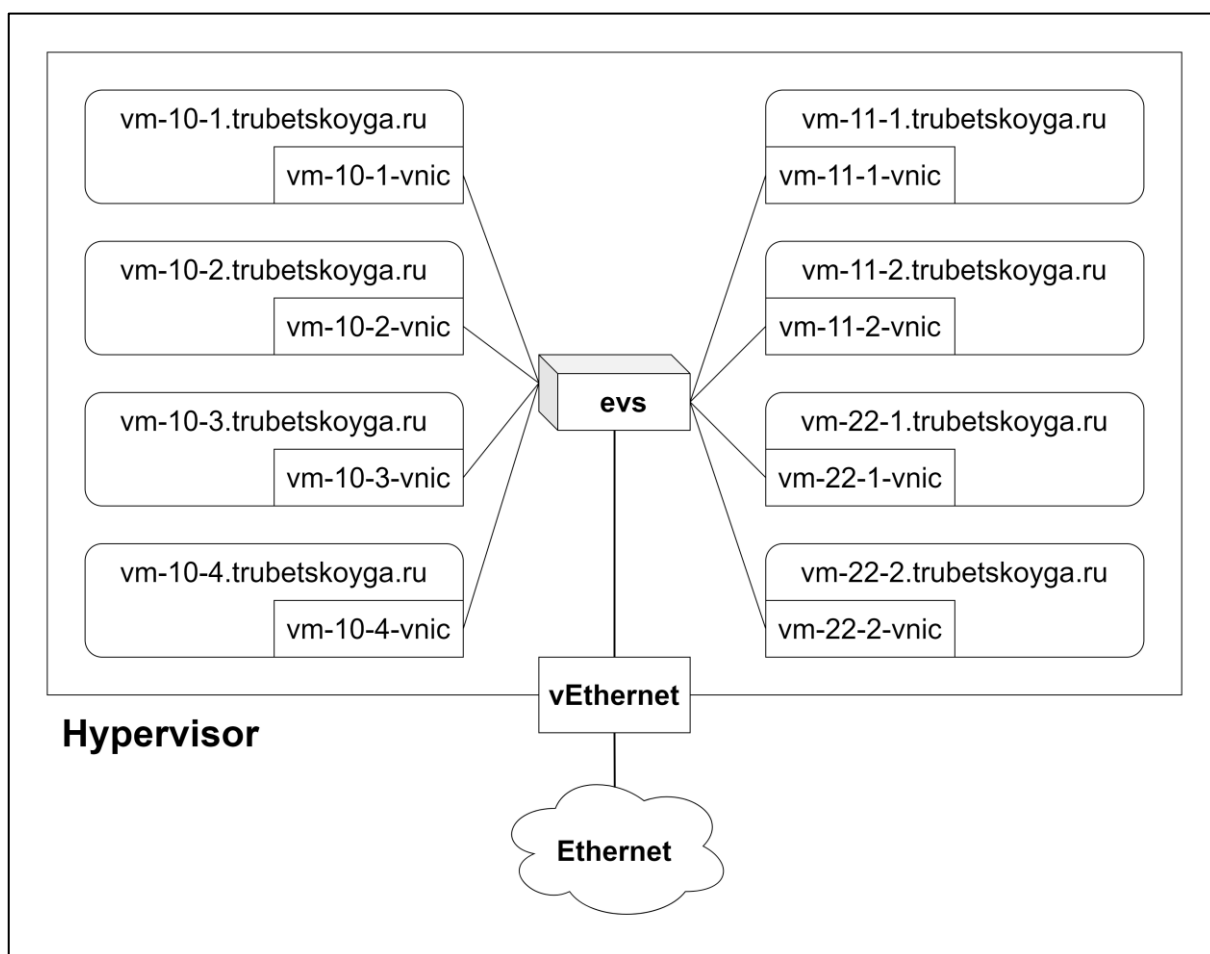


Рисунок 9 – Схема виртуальной локальной сети

В таблице 17 показаны имена виртуальных сетевых адаптеров виртуальных машин и виртуальных коммутаторов, к которым они подключены.

Таблица 17 – Сетевые параметры виртуальных машин

VM name	vNIC name	Connect to
vm-10-1.trubetskoyga.ru	vm-10-1-vnic	evs
vm-10-2.trubetskoyga.ru	vm-10-2-vnic	
vm-10-3.trubetskoyga.ru	vm-10-3-vnic	
vm-10-4.trubetskoyga.ru	vm-10-4-vnic	
vm-11-1.trubetskoyga.ru	vm-11-1-vnic	
vm-11-2.trubetskoyga.ru	vm-11-2-vnic	
vm-22-1.trubetskoyga.ru	vm-22-1-vnic	
vm-22-2.trubetskoyga.ru	vm-22-1-vnic	

Локальные учетные записи

После установки операционной системы необходимо создать локальную учетную запись.

Именем локальной учетной записи каждой гостевой ОС будет CN (без имени домена) виртуальной машины с приставкой «-lu» после имени.

Пароль учетной записи будет один и тот же. Это необходимо для упрощения работы, т.к. это тестовая среда, а не продуктивная. Одинаковый пароль на всех локальных и доменных учетных записях облегчит процесс аутентификации и авторизации, благодаря чему тестирование будет проведено быстрее. В продуктивной среде подобные практики не рекомендуются [12]. В таблице 18 показаны логины и пароли локальный учетных записей гостевых ОС.

Таблица 18 – Локальные учетные записи гостевых ОС

VM name	Login	Password
vm-10-1.trubetskoyga.ru	Vm-10-1-lu	1234QWer
vm-10-2.trubetskoyga.ru	Vm-10-2-lu	
vm-10-3.trubetskoyga.ru	Vm-10-3-lu	
vm-10-4.trubetskoyga.ru	Vm-10-4-lu	
vm-11-1.trubetskoyga.ru	Vm-11-1-lu	
vm-11-2.trubetskoyga.ru	Vm-11-2-lu	
vm-22-1.trubetskoyga.ru	Администратор	
vm-22-2.trubetskoyga.ru	Администратор	

3.5. Планирование домена

Для развертывания домена AD DS в организации необходим как минимум один контроллер домена.

В роли контроллера домена AD DS и DNS сервера будет выступать виртуальная машина `vm-22-1.trubetskoyga.ru`.

Согласно рекомендациям Microsoft [13], перед установкой роли AD DS на сервере, необходимо выполнить следующие действия:

- 1) установить последние обновления;
- 2) задать имя сервера (CN);
- 3) установить статический IP-адрес.

Далее необходимо ввести в домен остальные машины, но перед вводом необходимо также выполнить определенный порядок действий, рекомендуемый Microsoft:

- 1) изменить имя компьютера;
- 2) установить в качестве DNS сервера контроллер домена;
- 3) проверить доступность контроллера домена;
- 4) ввести компьютер в домен;
- 5) проверить, является ли компьютер членом домена.

В таблице 19 показано соответствие имен виртуальных машин и гостевых операционных систем, которые установлены на виртуальных жестких дисках, им назначенных.

Таблица 19 – Соответствие имен

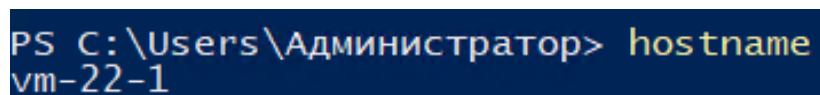
VM name	Guest OS CN
<code>vm-10-1.trubetskoyga.ru</code>	<code>vm-10-1</code>
<code>vm-10-2.trubetskoyga.ru</code>	<code>vm-10-2</code>
<code>vm-10-3.trubetskoyga.ru</code>	<code>vm-10-3</code>
<code>vm-10-4.trubetskoyga.ru</code>	<code>vm-10-4</code>
<code>vm-11-1.trubetskoyga.ru</code>	<code>vm-11-1</code>
<code>vm-11-2.trubetskoyga.ru</code>	<code>vm-11-2</code>
<code>vm-22-1.trubetskoyga.ru</code>	<code>vm-22-1</code>
<code>vm-22-2.trubetskoyga.ru</code>	<code>vm-22-2</code>

Настройка контроллера домена

Согласно порядку из предыдущего пункта, была произведена настройка гостевой операционной системы виртуальной машины с именем: `vm-22-1.trubetskoyga.ru`.

Также было задано новое имя операционной системы, которое было определено в таблице 13, после чего ОС была перезагружена для принятия изменений. Для этого была введена команда Powershell: `Rename-Computer -NewName "vm-22-1" -Restart`.

На рисунке 10 показана проверка принятых изменений командой `hostname`.

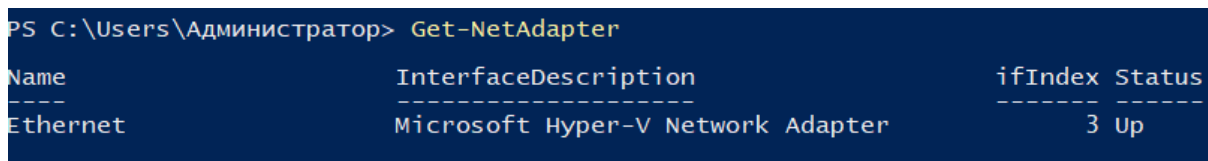


```
PS C:\Users\Администратор> hostname
vm-22-1
```

Рисунок 10 – Проверка изменения имени хоста

Перед тем, как задать статический IP-адрес сетевому адаптеру, необходимо узнать имя сетевого адаптера, которому этот адрес планируется назначить.

Узнать информацию об установленных сетевых параметрах можно при помощи команды Powershell `Get-NetAdapter`. На рисунке 11 показан результат выполнения команды.



```
PS C:\Users\Администратор> Get-NetAdapter
```

Name	InterfaceDescription	ifIndex	Status
Ethernet	Microsoft Hyper-V Network Adapter	3	Up

Рисунок 11 – Информация о сетевых адаптерах

Из рисунка 11 видно, что в системе установлен только один сетевой адаптер с именем Ethernet, который находится в рабочем состоянии.

Зная имя сетевого адаптера, возможно увидеть его параметры. Для этого была введена команда: `Get-NetIPConfiguration -InterfaceAlias Ethernet`.

На рисунке 12 показан результат выполнения команды.

```
PS C:\Users\Администратор> Get-NetIPConfiguration -InterfaceAlias ethernet

InterfaceAlias      : Ethernet
InterfaceIndex      : 3
InterfaceDescription : Microsoft Hyper-V Network Adapter
NetProfile.Name     : Сеть
IPv4Address          : 192.168.1.194
IPv6DefaultGateway  :
IPv4DefaultGateway  : 192.168.1.1
DNSServer            : 192.168.1.1
```

Рисунок 12 – Параметры сетевого адаптера Ethernet

Из рисунка 12 видно, что интерфейсу Ethernet уже назначены IP-адрес 4 версии (IPv4), а также адреса шлюза по умолчанию и адрес DNS сервера, который ссылается на маршрутизатор, являющийся DNS-сервером.

Перед тем, как задать статический IP-адрес и маршрут по умолчанию, необходимо удалить уже существующие маршруты по умолчанию, что можно сделать при помощи команды: `Remove-NetRoute -InterfaceAlias Ethernet`.

После очистки конфигурации сетевого интерфейса назначим ему IPv4-адрес в нотации CIDR `192.168.1.10/24` и адрес DNS сервера равным `192.168.1.1`.

Команда для добавления новых сетевых настроек: `Get-NetIpAddress -InterfaceAlias Ethernet | New-NetIpAddress -IpAddress 192.168.1.10 -PrefixLength 24 -DefaultGateway 192.168.1.1`.

На рисунке 13 показан результат выполнения команды.

```
PS C:\Users\Администратор> Get-NetIPConfiguration -InterfaceAlias ethernet

InterfaceAlias      : Ethernet
InterfaceIndex      : 3
InterfaceDescription : Microsoft Hyper-V Network Adapter
NetProfile.Name     : Сеть
IPv4Address          : 192.168.1.10
IPv6DefaultGateway  :
IPv4DefaultGateway  : 192.168.1.1
DNSServer            : fec0:0:0:ffff::1
                   : fec0:0:0:ffff::2
                   : fec0:0:0:ffff::3
```

Рисунок 13 – Измененные параметры сетевого адаптера Ethernet

Ввод клиентов в домен

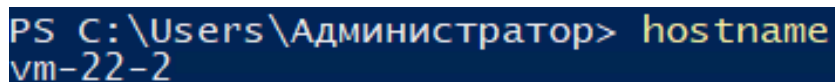
После настройки контроллера домена, необходимо такое провести настройку необходимых параметров операционных систем будущих клиентов домена.

Перед вводом в домен потенциальных клиентов необходимо для каждого клиента задать его имя, проверить доступность контроллера домена (DC) в локальной сети и задать в качестве основного DNS сервера IP-адрес контроллера домена.

Для виртуальной машины с именем `vm-22-2.trubetskoyga.ru` отдельно продумаем все эти операции в целях проверки возможности их выполнения.

Изменим CN гостевой ОС виртуальной машины командой: `Ren-
name-Computer -NewName "vm-22-2"`.

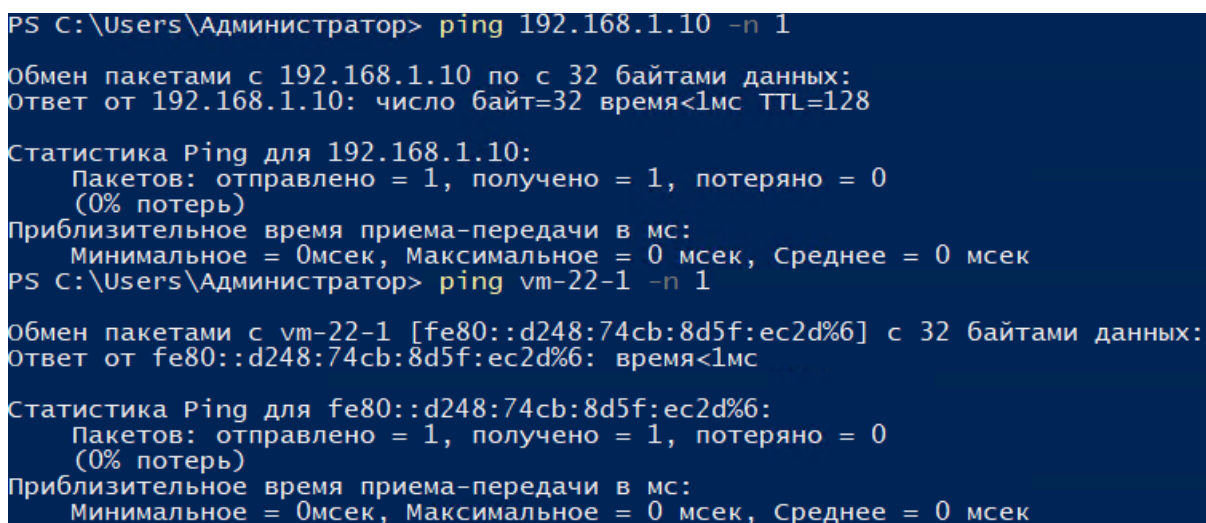
Проверка CN командой `hostname` показана на рисунке 14.



```
PS C:\Users\Администратор> hostname  
vm-22-2
```

Рисунок 14 – Проверка CN

На рисунке 15 показан результат проверки DC по его IP-адресу и CN командой `ping`.



```
PS C:\Users\Администратор> ping 192.168.1.10 -n 1  
Обмен пакетами с 192.168.1.10 по с 32 байтами данных:  
Ответ от 192.168.1.10: число байт=32 время<1мс TTL=128  
  
Статистика Ping для 192.168.1.10:  
Пакетов: отправлено = 1, получено = 1, потеряно = 0  
(0% потерь)  
Приблизительное время приема-передачи в мс:  
Минимальное = 0мсек, Максимальное = 0 мсек, Среднее = 0 мсек  
PS C:\Users\Администратор> ping vm-22-1 -n 1  
Обмен пакетами с vm-22-1 [fe80::d248:74cb:8d5f:ec2d%6] с 32 байтами данных:  
Ответ от fe80::d248:74cb:8d5f:ec2d%6: время<1мс  
  
Статистика Ping для fe80::d248:74cb:8d5f:ec2d%6:  
Пакетов: отправлено = 1, получено = 1, потеряно = 0  
(0% потерь)  
Приблизительное время приема-передачи в мс:  
Минимальное = 0мсек, Максимальное = 0 мсек, Среднее = 0 мсек
```

Рисунок 15 – Проверка доступности DC

Назначение в сетевых параметрах IP-адреса DC в роли основного DNS сервера при помощи команды: `Set-DNSClientServerAddress Ethernet -ServerAddresses ("192.168.1.10")`.

Результат выполнения команды показан на рисунке 16.

```
PS C:\Users\Администратор> Get-NetIPConfiguration -InterfaceAlias Ethernet
InterfaceAlias      : Ethernet
InterfaceIndex      : 6
InterfaceDescription : Microsoft Hyper-V Network Adapter
NetProfile.Name     : Сеть
IPv4Address         : 192.168.1.195
IPv6DefaultGateway  :
IPv4DefaultGateway  : 192.168.1.1
DNSServer           : 192.168.1.10
```

Рисунок 16 – Проверка DNS сервера

Ввод в домен рабочей станции осуществляется командлетом `Add-Computer` с определением имени домена, а также последующей перезагрузкой.

Для ввода клиента в домен и перезагрузки использовалась команда: `Add-Computer -DomainName "trubetskoyga.ru" -Restart`.

После ввода клиента в домен и перезагрузки ОС необходимо проверить, была ли рабочая станция введена в домен. Узнать это можно при помощи опроса FQDN хоста. Для этого использовалась команда: `[System.Net.Dns]::GetHostByName($env:computerName)`. Результат выполнения команды показан на рисунке 17.

```
PS C:\Users\Администратор> [System.Net.Dns]::GetHostByName($env:computerName)
HostName           Aliases AddressList
-----
vm-22-2.trubetskoyga.ru {} {192.168.1.195}
```

Рисунок 17 – Проверка членства в домене

По итогу рабочая станция `vm-22-2` была введена в домен `trubetskoyga.ru` успешно. В таблицах 20–24 отражены результаты выполнения тех же операций над остальными рабочими станциями.

Таблица 20 – Изменение имен рабочих станций

CN	Powershell команды	Результат
vm-10-1	Rename-Computer -NewName "vm-10-1" -Restart;	OK
vm-10-2	Rename-Computer -NewName "vm-10-2" -Restart;	OK
vm-10-3	Rename-Computer -NewName "vm-10-3" -Restart;	OK
vm-10-4	Rename-Computer -NewName "vm-10-4" -Restart;	OK
vm-11-1	Rename-Computer -NewName "vm-11-1" -Restart;	OK
vm-11-2	Rename-Computer -NewName "vm-11-2" -Restart;	OK

Из таблицы 20 видно, что имена рабочих станций изменены успешно.

Все имена корректны.

Таблица 21 – Проверка доступности DC для рабочих станций

CN	Powershell команды	Результат
vm-10-1	ping 192.168.1.10 -n 1; ping vm-22-1 -n 1;	OK
vm-10-2		OK
vm-10-3		OK
vm-10-4		OK
vm-11-1		OK
vm-11-2		OK

Таблица 22 – Настройка основного DNS сервера для рабочих станций

CN	Powershell команды	Результат
vm-10-1	Set-DNSClientServerAddress Ethernet -ServerAddresses ("192.168.1.10"); Get-NetIPConfiguration -InterfaceAlias Ethernet;	OK
vm-10-2		OK
vm-10-3		OK
vm-10-4		OK
vm-11-1		OK
vm-11-2		OK

Таблица 23 – Ввод в домен рабочих станций

CN	Powershell команды	Результат
vm-10-1	Add-Computer -DomainName "trubetskoyga.ru" -Restart;	OK
vm-10-2		OK
vm-10-3		OK
vm-10-4		OK
vm-11-1		OK
vm-11-2		OK

Из таблиц 21, 22 и 23 видно, что рабочие станции введены в домен без ошибок, а контроллер домена принимает запросы от клиентов.

Таблица 24 – Проверка FQDN рабочих станций

CN	Powershell команды	Результат
vm-10-1	[System.Net.Dns]::GetHostByName(\$env:computerName);	OK
vm-10-2		OK
vm-10-3		OK
vm-10-4		OK
vm-11-1		OK
vm-11-2		OK

Заключительным этапом проверки будет запрос всех членов домена непосредственно на контроллере домена при помощи командлета `Get-AD-Computer`. На рисунке 18 показан результат выполнения команды, из которого видно, что все компьютеры были успешно введены в домен.

```
PS C:\Users\Администратор> Get-ADComputer -Filter *
>> -Properties Name | select-object
>> -Expandproperty Name
VM-22-1
VM-22-2
VM-10-2
VM-10-1
VM-10-3
VM-11-1
VM-11-2
VM-10-4
PS C:\Users\Администратор>
```

Рисунок 18 – Проверка компьютеров домена

Из рисунка 18 видно, что все компьютеры были успешно введены в домен.

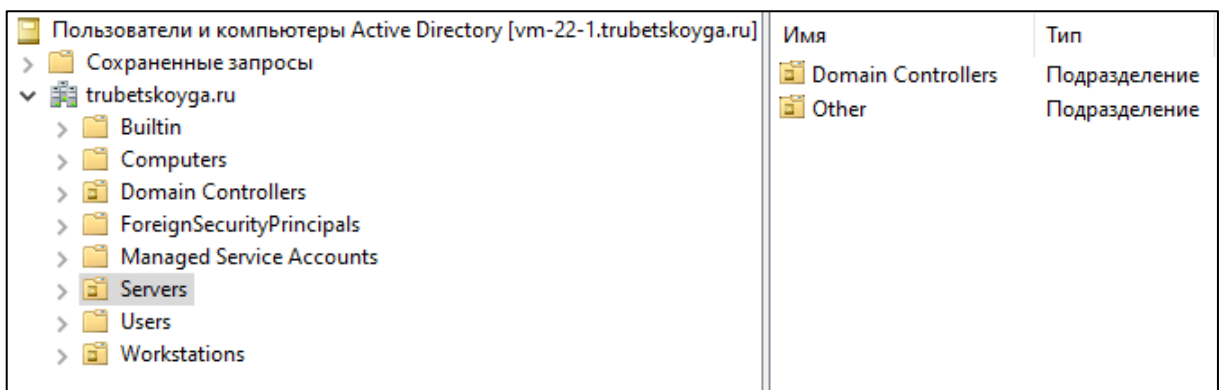
Распределение компьютеров домена

По умолчанию все компьютеры, введенные в домен, находятся в контейнере `Computers`. Для упрощения задач администрирования необходимо перераспределить их в новые организационные подразделения (OU). В корне домена созданы два OU с именами `Servers` и `Workstations`, которые содержат вложенные OU `Domain Controllers`, `Other` и `Windows 10`, `Windows 11` соответственно при помощи командлета `New-ADOrganizationalUnit`. Команды для создания новых OU приведены в листинге 1.

Листинг 1 – Команды для создания OU

```
New-ADOrganizationalUnit -Name "Servers" -Path "DC=trubetskoyga,DC=ru";
New-ADOrganizationalUnit -Name "Domain Controllers" -Path "OU=Servers,DC=trubetskoyga,DC=ru";
New-ADOrganizationalUnit -Name "Other" -Path "OU=Servers,DC=trubetskoyga,DC=ru";
New-ADOrganizationalUnit -Name "Workstations" -Path "DC=trubetskoyga,DC=ru";
New-ADOrganizationalUnit -Name "Windows 10" -Path "OU=Workstations,DC=trubetskoyga,DC=ru";
New-ADOrganizationalUnit -Name "Windows 11" -Path "OU=Workstations,DC=trubetskoyga,DC=ru";
```

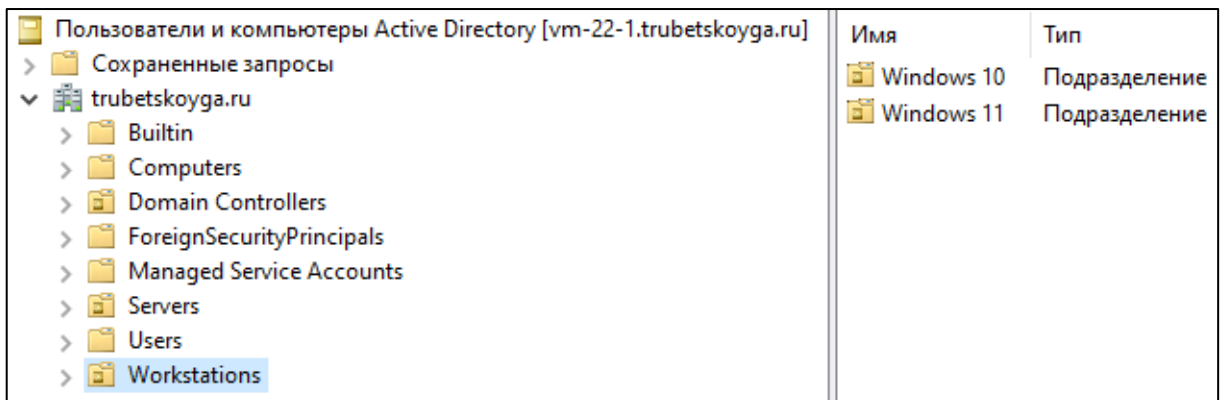
На рисунках 19 и 20 показана проверка наличия созданных OU при помощи оснастки AD UC.



The screenshot shows the Active Directory console for the domain trubetskoyga.ru. The left pane shows the tree structure with 'Servers' selected. The right pane displays a table of the selected OU's contents.

Имя	Тип
Domain Controllers	Подразделение
Other	Подразделение

Рисунок 19 – Проверка OU Servers



The screenshot shows the Active Directory console for the domain trubetskoyga.ru. The left pane shows the tree structure with 'Workstations' selected. The right pane displays a table of the selected OU's contents.

Имя	Тип
Windows 10	Подразделение
Windows 11	Подразделение

Рисунок 20 – Проверка OU Workstations

После успешного создания OU, необходимо переместить компьютеры домена в соответствующие им OU при помощи команд, представленных в листинге 2.

Листинг 2 – Команды для перемещения в OU

```
Get-ADComputer -Identity vm-22-1 | Move-ADObject -TargetPath "OU=Domain Controllers,OU=Servers,DC=trubetskoyga,DC=ru";
Get-ADComputer -Identity vm-22-2 | Move-ADObject -TargetPath "OU=Other,OU=Servers,DC=trubetskoyga,DC=ru";
Get-ADComputer -Identity vm-10-1 | Move-ADObject -TargetPath "OU=Windows 10,OU=Workstations,DC=trubetskoyga,DC=ru";
Get-ADComputer -Identity vm-10-2 | Move-ADObject -TargetPath "OU=Windows 10,OU=Workstations,DC=trubetskoyga,DC=ru";
Get-ADComputer -Identity vm-10-3 | Move-ADObject -TargetPath "OU=Windows 10,OU=Workstations,DC=trubetskoyga,DC=ru";
Get-ADComputer -Identity vm-10-4 | Move-ADObject -TargetPath "OU=Windows 10,OU=Workstations,DC=trubetskoyga,DC=ru";
Get-ADComputer -Identity vm-11-1 | Move-ADObject -TargetPath "OU=Windows 11,OU=Workstations,DC=trubetskoyga,DC=ru";
Get-ADComputer -Identity vm-11-2 | Move-ADObject -TargetPath "OU=Windows 11,OU=Workstations,DC=trubetskoyga,DC=ru";
```

На рисунке 21 показан процесс проверки перемещенных объектов при помощи командлета `Get-ADComputer` с запрашиваемыми параметрами имени (Name), уникального имени (DN), а также типа объекта (Object Class).

```
PS C:\Users\Администратор> Get-ADComputer -Filter * | Select-Object Name,DistinguishedName,ObjectClass
Name      DistinguishedName      ObjectClass
-----
VM-22-1   CN=VM-22-1,OU=Domain  computer
          Controllers,OU=Servers,DC=trubetskoyga,DC=ru
VM-22-2   CN=VM-22-2,OU=Other,  computer
          OU=Servers,DC=trubetskoyga,DC=ru
VM-10-2   CN=VM-10-2,OU=Windows  computer
          10,OU=Workstations,DC=trubetskoyga,DC=ru
VM-10-1   CN=VM-10-1,OU=Windows  computer
          10,OU=Workstations,DC=trubetskoyga,DC=ru
VM-10-3   CN=VM-10-3,OU=Windows  computer
          10,OU=Workstations,DC=trubetskoyga,DC=ru
VM-11-1   CN=VM-11-1,OU=Windows  computer
          11,OU=Workstations,DC=trubetskoyga,DC=ru
VM-11-2   CN=VM-11-2,OU=Windows  computer
          11,OU=Workstations,DC=trubetskoyga,DC=ru
VM-10-4   CN=VM-10-4,OU=Windows  computer
          10,OU=Workstations,DC=trubetskoyga,DC=ru
```

Рисунок 21 – Проверка перемещенных компьютеров

Из рисунка 21 видно, что все компьютеры домена были успешно перемещены в соответствующие им организационные подразделения.

Разрешение входящих подключений

По умолчанию в операционных системах Windows 10, 11 и Windows Server 2022 запрещены входящие ICMP подключения в правилах брандмауэра. На рисунках 22, 23 и 24 показаны результаты отправки по одному пакету с контроллера домена `vm-22-1` на компьютеры домена `vm-10-1`, `vm-11-1` и `vm-22-2` соответственно.


```
PS C:\Users\Администратор> ping -n 1 vm-10-1
Обмен пакетами с vm-10-1.trubetskoyga.ru [192.168.1.197] с 32 байтами данных:
Превышен интервал ожидания для запроса.
Статистика Ping для 192.168.1.197:
    Пакетов: отправлено = 1, получено = 0, потеряно = 1
    (100% потеря)
PS C:\Users\Администратор>
```

Рисунок 22 – Отправка пакета на vm-10-1

```
PS C:\Users\Администратор> ping -n 1 vm-11-1
Обмен пакетами с vm-11-1.trubetskoyga.ru [192.168.1.193] с 32 байтами данных:
Превышен интервал ожидания для запроса.
Статистика Ping для 192.168.1.193:
    Пакетов: отправлено = 1, получено = 0, потеряно = 1
    (100% потеря)
PS C:\Users\Администратор>
```

Рисунок 23 – Отправка пакета на vm-11-1

```
PS C:\Users\Администратор> ping -n 1 vm-22-2
Обмен пакетами с vm-22-2.trubetskoyga.ru [192.168.1.195] с 32 байтами данных:
ответ от 192.168.1.10: Заданный узел недоступен.
Статистика Ping для 192.168.1.195:
    Пакетов: отправлено = 1, получено = 1, потеряно = 0
    (0% потеря)
PS C:\Users\Администратор>
```

Рисунок 24 – Отправка пакета на vm-22-2

Из рисунков 22, 23 и 24 видно, что клиенты домена под управлением операционных систем Windows 10, 11 и Windows Server 2022 не принимают входящие пакеты по протоколу ICMP.

Для решения этой проблемы была создана групповая политика для всех клиентов домена, которая разрешает входящие ICMP-подключения по IPv4 и IPv6.

После применения групповой политики путем перезапуска клиентов доступность клиентов домена была проверена снова. На рисунках 25, 26 и 27 показаны результаты отправки по одному пакету с контроллера домена vm-22-1 на компьютеры домена vm-10-1, vm-11-1 и vm-22-2 соответственно.

```
PS C:\Users\Администратор> ping -n 1 vm-10-1
Обмен пакетами с vm-10-1.trubetskoyga.ru [192.168.1.197] с 32 байтами данных:
Ответ от 192.168.1.197: число байт=32 время<1мс TTL=128

Статистика Ping для 192.168.1.197:
    Пакетов: отправлено = 1, получено = 1, потеряно = 0
    (0% потерь)
Приблизительное время приема-передачи в мс:
    Минимальное = 0мсек, Максимальное = 0 мсек, Среднее = 0 мсек
PS C:\Users\Администратор>
```

Рисунок 25 – Повторная отправка пакета на vm-10-1

```
PS C:\Users\Администратор> ping -n 1 vm-11-1
Обмен пакетами с vm-11-1.trubetskoyga.ru [192.168.1.193] с 32 байтами данных:
Ответ от 192.168.1.193: число байт=32 время<1мс TTL=128

Статистика Ping для 192.168.1.193:
    Пакетов: отправлено = 1, получено = 1, потеряно = 0
    (0% потерь)
Приблизительное время приема-передачи в мс:
    Минимальное = 0мсек, Максимальное = 0 мсек, Среднее = 0 мсек
PS C:\Users\Администратор>
```

Рисунок 26 – Повторная отправка пакета на vm-11-1

```
PS C:\Users\Администратор> ping -n 1 vm-22-2
Обмен пакетами с vm-22-2.trubetskoyga.ru [192.168.1.206] с 32 байтами данных:
Ответ от 192.168.1.206: число байт=32 время<1мс TTL=128

Статистика Ping для 192.168.1.206:
    Пакетов: отправлено = 1, получено = 1, потеряно = 0
    (0% потерь)
Приблизительное время приема-передачи в мс:
    Минимальное = 0мсек, Максимальное = 0 мсек, Среднее = 0 мсек
PS C:\Users\Администратор>
```

Рисунок 27 – Повторная отправка пакета на vm-22-2

Из рисунков 25, 26 и 27 видно, что групповая политика была применена успешно.

Выводы по третьей главе

В этой главе был описан процесс организации тестовой среды, которая симулирует домен организации. Всем участникам домена были определены различные параметры, благодаря чему дальнейший процесс тестирования будет проходить быстрее.

4. ТЕСТИРОВАНИЕ

4.1. Проверка совместимости 5 и 7 версий Powershell.

Этот тест необходим, чтобы проверить совместимость разработанной утилиты с 5 и 7 основными (major) версиями Powershell.

Проверка версии Powershell осуществляется путем вывода информации в CLI при помощи команды `$PSVersionTable.PSVersion`. На рисунках 28 и 29 показан пример вывода информации об установленной версии.

```
PS C:\Users\Администратор> $PSVersionTable.PSVersion
Major  Minor  Build  Revision
-----
5      1      20348  2227
```

Рисунок 28 – Запрос версии в powershell.exe

```
PS C:\Users\Администратор> $PSVersionTable.PSVersion
Major  Minor  Patch  PreReleaseLabel  BuildLabel
-----
7      4      2
```

Рисунок 29 – Запрос версии в pwsh.exe

В таблице 25 приведена информация о прохождении тестов на проверку совместимости между различными версиями Powershell.

Таблица 25 – Тестирование на совместимость

№	Название теста	Ожидаемый результат	Тест пройден?
1.	Проверка на совместимость с 5 версией	Утилита и все основные функции выполняются корректно	Да
2.	Проверка на совместимость с 7 версией	Утилита и все основные функции выполняются корректно	Да

4.2. Проверка основных функций

Основные функции – это те функции, которые выполняются перед функциями сбора информации и которые необходимы для корректного запуска утилиты.

В таблицах 26–28 приведена информация о тестировании основных функций утилиты.

Таблица 26 – Тестирование `func_Directory`

№	Название теста	Ожидаемый результат	Тест пройден?
1.	Заданная директория существует	Заданная в переменной <code>\$global:Directory</code> директория существует и вызвана функция с именем <code>func_AD_Computers_Names_Collection</code>	Да
2.	Заданная директория не существует	Заданная в переменной <code>\$global:Directory</code> директория не существует и выведено корректное сообщение об ошибке	Да

Таблица 27 – Тестирование `func_AD_Computers_Names_Collection`

№	Название теста	Ожидаемый результат	Тест пройден?
1.	Сбор имен	Все имена компьютеров домена собраны, включая локальный контроллер домена	Да
2.	Сортировка имен	Все элементы массива <code>\$global:AD_Computers_Names</code> отсортированы в алфавитном порядке	Да
3.	Запуск последующей функции	Функция <code>func_AD_Computers_Availability</code> была вызвана корректно	Да

Таблица 28 – Тестирование `func_AD_Computers_Availability`

№	Название теста	Ожидаемый результат	Тест пройден?
1.	Проверка доступности	ICMP пакеты были доставлены корректно и был получен ответ о доступности или не доступности удаленного компьютера	Да
2.	Определение доступных компьютеров	Все доступные компьютеры были добавлены в массив <code>\$global:AD_Computers_Online_List</code>	Да
3.	Определение недоступных компьютеров	Все недоступные компьютеры были добавлены в массив <code>\$global:AD_Computers_Offline_List</code>	Да
4.	Определение доступности всех компьютеров	Переменной <code>\$global:AD_Computers_All_Online</code> присваивается значение <code>false</code> , если массив <code>\$global:AD_Computers_Offline_List</code> содержит хотя бы один элемент	Да
5.	Сортировка массивов	Элементы массивов, содержащих доступные и недоступные компьютеры соответственно, были отсортированы в алфавитном порядке	Да
6.	Вызов функций по сбору информации	Функции по сбору информации были вызваны корректно при условии, что все компьютеры домена были доступны	Да

№	Название теста	Ожидаемый результат	Тест пройден?
7.	Определение недоступных компьютеров	При наличии недоступных компьютеров, было выведено корректное предупреждение и построчный список всех недоступных компьютеров	Да
8.	Ожидание корректного ввода пользователем	При наличии недоступных компьютеров программа выводит предупреждение до тех пор, пока пользователь не введет значение <code>yes</code> или <code>no</code> .	Да
9.	Вызов функций по сбору информации при согласии пользователя	Функции, отвечающие за сбор информации, были вызваны корректно после того, как пользователь ввел значение <code>yes</code> .	Да

4.3. Проверка второстепенных функций

Второстепенные функции – это те функции, которые отвечают за сбор информации и которые выполняются после успешного выполнения основных функций.

В таблицах 29–30 приведена информация о тестировании функций по сбору информации.

Таблица 29 – Тестирование `func_AD_Computers_CPU_Info_Collection`

№	Название теста	Ожидаемый результат	Тест пройден?
1.	Запись названий параметров	Имена параметров записаны в файл корректно	Да
2.	Опрос компьютеров	Опрошены все компьютеры, имена которых находятся в переменной <code>%global:AD_Computers_Online_List</code>	Да
3.	Сбор параметров	Все параметры CPU были считаны корректно	Да
4.	Запись параметров	Все параметры (включая CN) CPU были записаны корректно в файл	Да
5.	Определение файла для записи	Файл был создан, он существует и доступен для чтения	Да

Таблица 30 – Тестирование `func_AD_Computers_OS_Info_Collection`

№	Название теста	Ожидаемый результат	Тест пройден?
1.	Запись названий параметров	Имена параметров записаны в файл корректно	Да

№	Название теста	Ожидаемый результат	Тест пройден?
2.	Опрос компьютеров	Опрошены все компьютеры, имена которых находятся в переменной \$global:AD_Computers_Online_List, включая локальный контроллер домена	Да
3.	Сбор параметров	Все параметры ОС были считаны корректно	Да
4.	Запись параметров	Все параметры (включая CN) ОС были записаны корректно в файл	Да
5.	Определение файла для записи	Файл был создан, он существует и доступен для чтения	Да

4.4. Проверка собранной информации

В листинге 3 показано содержимое файла, в который была записана информация о CPU. В листинге 4 показано содержимое файла, в который была записана информация об ОС.

Листинг 3 – Записанная информация о CPU

```
CN;Name;NumberOfCores;NumberOfLogicalProcessors;NumberOfEnabledCore;CurrentClockSpeed;MaxClockSpeed;
VM-10-1;AMD Ryzen 5 4500 6-Core Processor ;1;1;;3593;3593;
VM-10-2;AMD Ryzen 5 4500 6-Core Processor ;2;4;4;3593;3593;
VM-10-3;AMD Ryzen 5 4500 6-Core Processor ;2;4;;3593;3593;
VM-10-4;AMD Ryzen 5 4500 6-Core Processor ;4;8;8;3593;3593;
VM-11-1;AMD Ryzen 5 4500 6-Core Processor ;1;2;2;3593;3593;
VM-11-2;AMD Ryzen 5 4500 6-Core Processor ;2;4;4;3593;3593;
VM-22-1;AMD Ryzen 5 4500 6-Core Processor ;1;2;2;3593;3593;
VM-22-2;AMD Ryzen 5 4500 6-Core Processor ;4;8;8;3593;3593;
```

Листинг 4 – Записанная информация об ОС

```
CN;Name;Version;OSArchitecture;OSLanguage;CurrentTimeZone;InstallDate;SystemDirectory;
VM-10-1;Майкрософт Windows 10 Pro|C:\Windows|\Device\Harddisk0\Partition2;10.0.19045;32-разрядная;1049;300;20240130121339.000000+300;C:\Windows\system32;
VM-10-2;Майкрософт Windows 10 Pro|C:\Windows|\Device\Harddisk0\Partition3;10.0.19045;64-разрядная;1049;300;20240130122750.000000+300;C:\Windows\system32;
VM-10-3;Майкрософт Windows 10 Pro для рабочих станций|C:\Windows|\Device\Harddisk0\Partition2;10.0.19045;32-разрядная;1049;300;20240130121617.000000+300;C:\Windows\system32;
VM-10-4;Майкрософт Windows 10 Pro для рабочих станций|C:\Windows|\Device\Harddisk0\Partition3;10.0.19045;64-разрядная;1049;300;20240130124158.000000+300;C:\Windows\system32;
VM-11-1;Майкрософт Windows 11 Pro|C:\Windows|\Device\Harddisk0\Partition3;10.0.22631;64-разрядная;1049;180;20240130094647.000000+180;C:\Windows\system32;
```

```
VM-11-2;Майкрософт Windows 11 Pro|C:\Windows|\Device\Harddisk0\Parti-  
tion3;10.0.22631;64-разрядная;1049;180;20240130094730.000000+180;C:\Win-  
dows\system32;  
VM-22-1;Майкрософт Windows Server 2022 Standard|C:\Windows|\De-  
vice\Harddisk0\Partition3;10.0.20348;64-  
разрядная;1049;300;20240130113614.000000+300;C:\Windows\system32;  
VM-22-2;Майкрософт Windows Server 2022 Datacenter|C:\Windows|\De-  
vice\Harddisk0\Partition3;10.0.20348;64-  
разрядная;1049;300;20240130113624.000000+300;C:\Windows\system32;
```

Проанализировав листинг 3 и таблицу 15, можно сделать вывод о том, что собранная информация о CPU является корректной. Модель и производитель CPU совпадают с CPU хоста гипервизора. Количество логических ядер соответствует количеству заданных vCPU для каждой из VM.

Проанализировав листинг 4 и таблицу 14, можно сделать вывод о том, что собранная информация об ОС является корректной. Семейство и название установленной ОС, а также ее разрядность и язык интерфейса соответствуют гостевым ОС. Дата установки является верной и не была скомпрометирована. Временная зона не на всех гостевых ОС является корректной для текущего часового пояса (UTC + 5:00).

Проанализировав листинг 3, можно сделать вывод о том, что собранная информация о производителе и модели (имени) CPU содержит лишние пробелы в конце строки, следовательно, выводимая информация требует дальнейшей обработки.

Проанализировав листинг 4, можно сделать вывод о том, что собранная информация не выводится в привычном для пользователя виде и требует дальнейшей обработки.

Выводы по четвертой главе

В этой главе был представлен процесс тестирования разработанной утилиты. Тестирование проведено успешно. Все основные модули обеспечивают стабильную работу утилиты. Все второстепенные модули собирают информацию корректно. Собранная информация является полезной и готова для реализации в продуктивной среде.

ЗАКЛЮЧЕНИЕ

В рамках выпускной квалификационной работы была разработана утилита для сбора информации об участниках домена и решены следующие задачи.

1. Проведен анализ предметной области. Изучены технологии, необходимые для успешной реализации программного продукта.

2. Утилита была успешно спроектирована и реализована. Архитектура утилиты позволяет добавлять новые модули и упрощает отладку и процесс ввода будущих нововведений.

3. Организована тестовая среда, которая симулирует домен организации для проведения последующего тестирования.

4. Успешно проведено тестирование разработанной утилиты.

Таким образом, поставленные задачи решены, а цель выпускной квалификационной работы была достигнута.

В дальнейшем планируется тестирование и отладка существующих основных и второстепенных модулей, а также добавление новых второстепенных модулей для сбора информации о других подсистемах компьютеров домена. Также рассматривается возможность добавления графического интерфейса пользователя и интеграция с другими операционными системами и службами каталогов.

ЛИТЕРАТУРА

1. Desmond B., Richards J., Allen R., Lowe-Norris A. G., Designing, Deploying and Running Active Directory. // 5th edition, 2013. – 735 с.
2. Спецификация протокола LDAP. [Электронный ресурс] URL: <https://datatracker.ietf.org/doc/html/rfc2253> (дата обращения: 25.04.2024 г.).
3. Обзор Powershell. [Электронный ресурс] URL: <https://learn.microsoft.com/ru-ru/powershell/scripting/overview?view=powershell-7.4> (дата обращения: 25.04.2024 г.).
4. Жизненный цикл Powershell. [Электронный ресурс] URL: <https://learn.microsoft.com/en-us/powershell/scripting/install/powershell-support-lifecycle?view=powershell-7.4> (дата обращения: 25.04.2024 г.).
5. Модуль Powershell для AD. [Электронный ресурс] URL: <https://learn.microsoft.com/en-us/powershell/module/activedirectory/?view=windowsserver2022-ps> (дата обращения: 25.04.2024 г.).
6. Командлет Get-ADComputer. [Электронный ресурс] URL: <https://learn.microsoft.com/en-us/powershell/module/activedirectory/get-adcomputer?view=windowsserver2022-ps> (дата обращения: 25.04.2024 г.).
7. Командлет Get-WmiObject. [Электронный ресурс] URL: <https://learn.microsoft.com/en-us/powershell/module/microsoft.powershell.management/get-wmiobject?view=powershell-5.1> (дата обращения: 25.04.2024 г.).
8. Класс Win32_Processor. [Электронный ресурс] URL: <https://learn.microsoft.com/ru-ru/windows/win32/cimwin32prov/win32-processor> (дата обращения: 25.04.2024 г.).
9. Системные требования Windows Server 2022. [Электронный ресурс] URL: <https://learn.microsoft.com/en-us/windows-server/get-started/hardware-requirements?tabs=cpu> (дата обращения: 25.04.2024 г.).
10. Системные требования Windows 10. [Электронный ресурс] URL: <https://support.microsoft.com/en-us/windows/windows-10-system-requirements-6d4e9a79-66bf-7950-467c-795cf0386715> (дата обращения: 25.04.2024 г.).

11. Системные требования Windows 11. [Электронный ресурс] URL: <https://www.microsoft.com/en-us/windows/windows-11-specifications> (дата обращения: 25.04.2024 г.).

12. Лучшие практики обеспечения безопасности парольной защиты для организаций. [Электронный ресурс] URL: <https://www.netwrix.com/password-policy-best-practices.html> (дата обращения: 25.04.2024 г.).

13. Подробное руководство по развертыванию AD DS. [Электронный ресурс] URL: <https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/deploy/install-active-directory-domain-services--level-100-> (дата обращения: 25.04.2024 г.).

14. Лучшие практики наименования виртуальных машин. [Электронный ресурс] URL: <https://www.techtarget.com/searchitoperations/answer/Best-practices-for-naming-VMware-virtual-machines> (дата обращения: 25.04.2024 г.).

15. Обзор AD DS. [Электронный ресурс] URL: <https://learn.microsoft.com/ru-ru/windows-server/identity/ad-ds/get-started/virtual-dc/active-directory-domain-services-overview> (дата обращения: 25.04.2024 г.).

16. Обзор Azure. [Электронный ресурс] URL: <https://partner.microsoft.com/ru-kz/solutions/azure> (дата обращения: 25.04.2024 г.).

17. Нотация CIDR. [Электронный ресурс] URL: <http://subnets.ru/wrapper.php?p=35> (дата обращения: 25.04.2024 г.).

18. Информация о CLI. [Электронный ресурс] URL: <https://itglobal.com/ru-ru/company/glossary/cli/> (дата обращения: 25.04.2024 г.).

19. Информация о CLR. [Электронный ресурс] URL: <https://learn.microsoft.com/ru-ru/dotnet/standard clr> (дата обращения: 25.04.2024 г.).

20. Информация о CN. [Электронный ресурс] URL: https://docs.gandi.net/en/ssl/create/common_name.html (дата обращения: 25.04.2024 г.).
21. Информация о CPU. [Электронный ресурс] URL: https://www.reg.ru/blog/chto-takoe-protssessor-cpu/amp/?utm_source=google.com&utm_medium=organic&utm_campaign=google.com&utm_referrer=google.com (дата обращения: 25.04.2024 г.).Ц
22. Информация о DC. [Электронный ресурс] URL: <https://encyclopedia.kaspersky.ru/glossary/domain-controller/> (дата обращения: 25.04.2024 г.).
23. Информация о DHCP. [Электронный ресурс] URL: <https://selectel.ru/blog/dhcp-protocol/> (дата обращения: 25.04.2024 г.).
24. Информация о DIT. [Электронный ресурс] URL: <https://pro-ldap.ru/tr/zytrax/ch2/> (дата обращения: 25.04.2024 г.).
25. Информация о DN. [Электронный ресурс] URL: <https://learn.microsoft.com/en-us/previous-versions/windows/desktop/ldap/distinguished-names> (дата обращения: 25.04.2024 г.).
26. Информация о DNS. [Электронный ресурс] URL: <https://www.cloudflare.com/learning/dns/what-is-dns/> (дата обращения: 25.04.2024 г.).
27. Информация об Ethernet. [Электронный ресурс] URL: <https://neerc.ifmo.ru/wiki/index.php?title=Ethernet> (дата обращения: 25.04.2024 г.).
28. Обзор MS Exchange Server. [Электронный ресурс] URL: <https://swsu.ru/kva/kompyuternye-programmy-i-igry/chto-takoe-microsoft-exchange-server.php> (дата обращения: 25.04.2024 г.).
29. Информация о FQDN. [Электронный ресурс] URL: <https://whois.ru/fully-qualified-domain-name.html> (дата обращения: 25.04.2024 г.).

30. Информация о GUI. [Электронный ресурс] URL: <https://itglobal.com/ru-ru/company/glossary/gui/> (дата обращения: 25.04.2024 г.).
31. Информация о GUID. [Электронный ресурс] URL: <https://www.techtarget.com/searchwindowsserver/definition/GUID-global-unique-identifier>. (дата обращения: 25.04.2024 г.).
32. Информация о протоколе ICMP. [Электронный ресурс] URL: <https://www.ibm.com/docs/ru/aix/7.2?topic=protocols-internet-control-message-protocol> (дата обращения: 25.04.2024 г.).
33. Информация об IP. [Электронный ресурс] URL: <http://www.iana.org/numbers> (дата обращения: 25.04.2024 г.).
34. Информация об IP-адресах. [Электронный ресурс] URL: <https://habr.com/ru/articles/703854/> (дата обращения: 25.04.2024 г.).
35. Информация о протоколе LDAP. [Электронный ресурс] URL: <https://habr.com/ru/articles/538662/> (дата обращения: 25.04.2024 г.).
36. Информация о платформе NET. [Электронный ресурс] URL: <https://learn.microsoft.com/ru-ru/dotnet/welcome> (дата обращения: 25.04.2024 г.).
37. Информация о NIC. [Электронный ресурс] URL: <https://www.sciencedirect.com/topics/computer-science/network-interface-controller> (дата обращения: 25.04.2024 г.).
38. Информация о NOS. [Электронный ресурс] URL: <https://www.geeksforgeeks.org/what-is-a-network-operating-system/> (дата обращения: 25.04.2024 г.).
39. Информация об OS. [Электронный ресурс] URL: https://www.techtarget-com.translate.goog/whatis/definition/operating-system-OS?_x_tr_sl=en&_x_tr_tl=ru&_x_tr_hl=ru&_x_tr_pto=sc (дата обращения: 25.04.2024 г.).

40. Информация о RDN. [Электронный ресурс] URL: <https://docs.oracle.com/cd/E19182-01/820-6573/ghusi/index.html> (дата обращения: 25.04.2024 г.).
41. Информация о shell. [Электронный ресурс] URL: <https://habr.com/ru/articles/548078/> (дата обращения: 25.04.2024 г.).
42. Обзор MS SQL Server. [Электронный ресурс] URL: <https://learn.microsoft.com/ru-ru/sql/tools/overview-sql-tools?view=sql-server-ver16> (дата обращения: 25.04.2024 г.).
43. Информация о Windows. [Электронный ресурс] URL: https://www.tadviser.ru/index.php/%D0%9F%D1%80%D0%BE%D0%B4%D1%83%D0%BA%D1%82:Microsoft_Windows (дата обращения: 25.04.2024 г.).
44. Информация о XML. [Электронный ресурс] URL: https://www.nic.ru/help/что-такое-xml-format-i-gde-on-ispolbzuetsya_11113.html (дата обращения: 25.04.2024 г.).
45. Информация об АРМ. [Электронный ресурс] URL: <https://alexrovich.ru/info/articles/что-значит-arm-автоматизированное-рабочее-место/> (дата обращения: 25.04.2024 г.).
46. Информация о виртуальных машинах. [Электронный ресурс] URL: https://www.nic.ru/help/что-такое-virtualnaya-mashina_11115.html (дата обращения: 25.04.2024 г.).
47. Информация о деревьях домена. [Электронный ресурс] URL: <https://learn.microsoft.com/ru-ru/windows/win32/ad/domain-trees> (дата обращения: 25.04.2024 г.).
48. Информация о домене. [Электронный ресурс] URL: <https://learn.microsoft.com/ru-ru/windows-server/identity/ad-ds/plan/understanding-the-active-directory-logical-model> (дата обращения: 25.04.2024 г.).
49. Информация о командлетах Powershell. [Электронный ресурс] URL: <https://learn.microsoft.com/ru-ru/powershell/scripting/developer/cmdlet/>

cmdlet-overview?view=powershell-7.3 (дата обращения: 25.04.2024 г.).

50. Информация о коммутаторах. [Электронный ресурс] URL: <https://servergate.ru/articles/chto-takoe-kommutator-/> (дата обращения: 25.04.2024 г.).

51. Информация о лесе доменов. [Электронный ресурс] URL: <https://learn.microsoft.com/ru-ru/powershell/scripting/developer/cmdlet/cmdlet-overview?view=powershell-7.3> (дата обращения: 25.04.2024 г.).

52. Информация о маршрутизаторах. [Электронный ресурс] URL: <https://rootstore.ru/news/marshrutizatory-chto-eto-takoe-i-zachem-oni-nuzhny/> (дата обращения: 25.04.2024 г.).

53. Информация о репозиториях. [Электронный ресурс] URL: <https://blog.skillfactory.ru/glossary/repozitorij/> (дата обращения: 25.04.2024 г.).

54. Информация о службах каталогов. [Электронный ресурс] URL: https://interface31.ru/tech_it/2011/01/sluzhby-katalogov-chast-1-obshhie-ponyatiya.html (дата обращения: 25.04.2024 г.).

55. Информация о хосте. [Электронный ресурс] URL: <https://www.calltouch.ru/blog/glossary/host/> (дата обращения: 25.04.2024 г.).

ПРИЛОЖЕНИЕ. Код разработанной утилиты

Листинг 1 – Код разработанной утилиты

```
function func_Directory
{
    $global:Directory = "C:\\";

    if (Test-Path $global:Directory)
    {
        func_AD_Computers_Names_Collection;
    }
    else
    {
        Write-Host "The directory you specified does not exist. The
utility start was interrupt.";
    }
}
function func_AD_Computers_Names_Collection
{
    $global:AD_Computers_Names = Get-ADComputer -Filter * -Properties
Name | Select-Object -ExpandProperty Name;
    $global:AD_Computers_Names = $global:AD_Computers_Names | Sort-Ob-
ject;
    $global:AD_Computers_All_Online = $true;
    $global:AD_Computers_Online_List = @();
    $global:AD_Computers_Offline_List = @();

    func_AD_Computers_Availability;
}
function func_AD_Computers_Availability
{
    for ($i = 1; $i -le $global:AD_Computers_Names.Length; $i++)
    {
        if (Test-Connection -BufferSize 32 -Count 1 -ComputerName
$global:AD_Computers_Names[$i-1] -Quiet)
        {
            $global:AD_Computers_Online_List += $global:AD_Comput-
ers_Names[$i-1];
        }
        else
        {
            $global:AD_Computers_All_Online = $false;
            $global:AD_Computers_Offline_List += $global:AD_Comput-
ers_Names[$i-1];
        }
    }

    $global:AD_Computers_Online_List = $global:AD_Computers_Online_List |
Sort-Object;
    $global:AD_Computers_Offline_List = $global:AD_Computers_Offline_List
| Sort-Object;

    if ($AD_Computers_All_Online -eq "true")
    {
        func_AD_Computers_CPU_Info_Collection;
        func_AD_Computers_OS_Info_Collection;
    }
    else
    {
        Write-Host "Not all domain computers are available.`nList of
unavailable computers:";
        Write-Host -Separator "`n" $global:AD_Computers_Offline_List;
    }
}
```

Продолжение листинга 1 приложения

```
Write-Host "Continue collecting information? (yes/no)";

while($true)
{
    $UserInput = Read-Host;
    if ($UserInput -eq "yes")
    {
        func_AD_Computers_CPU_Info_Collection;
        func_AD_Computers_OS_Info_Collection;
    }
    elseif ($UserInput -eq "no")
    {
        Write-Host "The utility start was interrupt.";
    }
    else
    {
        Clear-Host;
        Write-Host "Please try again and enter the correct
answer (yes/no).";
        continue;
    }
    break;
}
}

function func_AD_Computers_CPU_Info_Collection
{
    $AD_Computers_CPU_info_file_name = "AD_Computers_CPU_info.txt";
    $AD_Computers_CPU_info_file_path = $global:Directory + $AD_Comput-
ers_CPU_info_file_name;

    $AD_Computer_CPU_parameters = @(
        "Name",
        "NumberOfCores",
        "NumberOfLogicalProcessors",
        "NumberOfEnabledCore",
        "CurrentClockSpeed",
        "MaxClockSpeed"
    );

    [string]$AD_Computer_CPU_parameters_string = "CN;";

    for ($i = 0; $i -lt $AD_Computer_CPU_parameters.Length; $i++)
    {
        $AD_Computer_CPU_parameters_string += $AD_Computer_CPU_parame-
ters[$i];
        $AD_Computer_CPU_parameters_string += ";";
    }

    Add-Content -Path $AD_Computers_CPU_info_file_path -Value $AD_Com-
puter_CPU_parameters_string;

    [String]$AD_Computer_CPU_info;

    for ($i = 0; $i -lt $global:AD_Computers_Online_List.Length; $i++)
    {
        $AD_Computer_CPU_info = "";
        $AD_Computer_CPU_info += $global:AD_Computers_Online_List[$i] +
";";

        for ($j = 0; $j -lt $AD_Computer_CPU_parameters.Length; $j++)
```


Окончание листинга 1 приложения

```
{
    $AD_Computer_CPU_info += Get-WmiObject -Class Win32_Pro-
cessor -ComputerName $global:AD_Computers_Online_List[$i] | Select-Object -
ExpandProperty $AD_Computer_CPU_parameters[$j];

    $AD_Computer_CPU_info += ";";
}

Add-Content -Path $AD_Computers_CPU_info_file_path -Value
$AD_Computer_CPU_info;
}
}
function func_AD_Computers_OS_Info_Collection
{
    $AD_Computers_OS_info_file_name = "AD_Computers_OS_info.txt";
    $AD_Computers_OS_info_file_path = $global:Directory + $AD_Comput-
ers_OS_info_file_name;

    $AD_Computer_OS_parameters = @(
        "Name",
        "Version",
        "OSArchitecture",
        "OSLanguage",
        "CurrentTimeZone",
        "InstallDate",
        "SystemDirectory"
    );

    [string]$AD_Computer_OS_parameters_string = "CN;";

    for ($i = 0; $i -lt $AD_Computer_OS_parameters.Length; $i++)
    {
        $AD_Computer_OS_parameters_string += $AD_Computer_OS_parame-
ters[$i];
        $AD_Computer_OS_parameters_string += ";";
    }

    Add-Content -Path $AD_Computers_OS_info_file_path -Value $AD_Com-
puter_OS_parameters_string;

    [String]$AD_Computer_OS_info;

    for ($i = 0; $i -lt $global:AD_Computers_Online_List.Length; $i++)
    {
        $AD_Computer_OS_info = "";
        $AD_Computer_OS_info += $global:AD_Computers_Online_List[$i] +
";";

        for ($j = 0; $j -lt $AD_Computer_OS_parameters.Length; $j++)
        {
            $AD_Computer_OS_info += Get-WmiObject -Class Win32_Oper-
atingSystem -ComputerName $global:AD_Computers_Online_List[$i] | Select-Ob-
ject -ExpandProperty $AD_Computer_OS_parameters[$j];

            $AD_Computer_OS_info += ";";
        }

        Add-Content -Path $AD_Computers_OS_info_file_path -Value
$AD_Computer_OS_info;
    }
}
func_Directory
```