**BC/NW 2019 № 1 (34):8.1**

**АЛГОРИТМИЗАЦИЯ ПОСТРОЕНИЯ ТОПОЛОГИЧЕСКОЙ МАТРИЦЫ КОРПОРАТИВНОЙ ВЫЧИСЛИТЕЛЬНОЙ СЕТИ**

Волкова А.А., Абросимов Л.И.

Разработка и модернизация эффективно функционирующих региональных и корпоративных вычислительных сетей (КВС) является сложным и трудоемким процессом, так как разработчику необходимо учитывать много критериев эффективности, большое количество ограничений, широкий спектр разнообразных устройств, каждое из которых обладает индивидуальным набором технических характеристик, зависящих от местоположения и взаимосвязей с другими устройствами.

При разработке КВС для качественной оценки ее производительности необходимы модели, которые: учитывают реальную размерность ВС; базируются на аналитических соотношениях, позволяющих оперативно и детально оценивать предельную производительность ВС; позволяют однозначно, с необходимой детализацией соотносить результаты, определенные для элементов моделирования, с параметрами устройств ВС.

***Проблемы, с которыми сталкивается разработчик при проектировании вычислительных сетей***

Разработчику ВС и администратору приходится постоянно сталкиваться со следующими до сих пор не решенными проблемами:

• Необходимость упорядочить и постоянно обновлять исходные данные, которые могут понадобиться при разработке и модернизации сетей ЭВМ;

• При разработке и модернизации необходимо согласовать производительность ВС с постоянно возрастающими требованиями пользователей для обеспечения качественного обслуживания пользователей;

• Необходимость определять производительность и выявлять узкие места для различных типов топологий и архитектур сетей ЭВМ, чтобы принимать правильные решения по их устранению;

•Необходимость использовать и согласовать между собой адаптивные разнообразные методы и модели, обеспечивающие количественное обоснование проектных решений;

Для успешного решения указанных проблем необходима компьютеризация не только процессов хранения и поиска данных, но и процессов построения моделей.

При разработке и модернизации ВС используются концептуальные положения метода контуров [1], содержащие 8 этапов:

1) Описание структуры вычислительной сети

2) Описание функциональной схемы вычислительной сети

3) Представление графовой модели вычислительной сети

4) Составление линейных уравнений баланса

5) Составление нелинейных уравнений баланса

6) Решение линейных уравнений баланса

7) Решение нелинейных уравнений баланса

8) Определение функциональных характеристик вычислительных сетей

Структура S вычислительной сети определяет такие основные характеристики, как количество *N* узлов сети и наличие связей между узлами, задаваемых матрицей связей. Кроме того, с элементами структуры связываются параметры ресурсов устройств и параметры трафика нагрузки сети.

Поэтому формализованное представление структуры КВС, пригодное для автоматизированного ввода в ЭВМ, которая выполняет этапы моделирования КВС реальной размерности, является чрезвычайно важной задачей.

**Матричная запись моделей**

На сегодняшний день разработаны [1] следующие матричные модели, позволяющие выполнять все операции по оценке производительности КВС:

1) Топологическая матрица (Т) - предоставляет информацию об устройствах (тип, кол-во разъемов) и о соединениях между ними (тип линии связи, тип канала связи) в пределах одной комнаты.

где – тип устройства **У**, *А* и *В* – идентификатор, определяющий местоположение **У,** *R* – количество разъемов в устройстве **У, -** тип линии связи Л (физическая реализация), - тип канала связи Л (дуплексный, полудуплексный, симплексный).

2) Логическая матрица (L) - предназначена для отображения узлов, соответствующих тем устройствам, представленным в матрице Т, которым в ВС присвоены логические имена, адреса и номера интерфейсов.

После построения логической матрицы, появляется возможность сформировать контура *q*  и фазы поступающих на обслуживание потоков сообщений. Далее, при наличии информации об интенсивности  обслуживания сообщений в узлах, осуществляется переход к следующему этапу *«Описание функциональной структуры вычислительной сети».*

3) Функциональная матрица (F) - является основой математической модели ВС и предназначена для отображения функциональных элементов, соответствующих устройствам и линиям, которые задерживают транзакции при обработке и оказывают существенное влияние на производительность ВС.

где *i* и *j* – соответственно, строки и столбцы в матрице F, а r и k – разъемы устройств А и В.

После построения функциональной матрицы автоматически составляются и решаются системы *линейных* и *нелинейных уравнений* и рассчитываются *функциональные характеристики.*

**Постановка задачи**

Настоящая работа посвящена первому этапу метода контуров.

Необходимо в соответствии с ограничениями, которые рассмотрены в следующем разделе разработать алгоритм автоматизированного построения:

* топологической матрицы КВС для записи в моделирующую ЭВМ
* матрицу изображения КВС для разработчика и администратора.

Целью разработки является повышение качества принятия решений сетевого администратора при эксплуатации и модернизации вычислительной сети.

Для достижения поставленной цели поставлены и решены следующие задачи:

1. расширить данную математическую модель до уровня этажа, корпуса, здания и конкретной территории (в связи с тем, что существующая топологическая структура описывает соединения в рамках одной комнаты);
2. отобразить структуру заданной КВС с возможностью редактирования;
3. предусмотреть возможность создания новой топологической матрицы КВС (для любого ее уровня) или отображения/ изменения/ удаления существующей.

***Исходные данные и ограничения для реализуемого алгоритма***

Для разработки алгоритма построения топологической матрицы вычислительной сети необходимы следующие две группы данных:

Таблица 1.1

Входные данные

|  |  |
| --- | --- |
| **Местоположение** | **Устройства** |
| Карта зданий корпорации  Поэтажные планы зданий  Количество зданий  Количество корпусов  Количество этажей  Количество комнат  Таблица соответствия «Комната – Тип»  Номер комнаты – номер и тип устрйства | Типы устройств {У}, каналов{К} и линий связи{Л}  Спецификация {У}, {К}, {Л}  Принципиальная схема структуры КВС  Схема межкомнатных связей  Типовые и специализированные схемы комнат (информация о количестве устройств в комнате и соединениями между ними) |

Ограничения, установленные в представленной работе:

1. Рассматривается модель вычислительной сети, которая состоит из трёх зданий, в каждом из которых располагается по три корпуса, в корпусах по три этажа, а на каждом этаже по семь комнат определенных типов, представленных в приложении;
2. Три основных типа устройств: рабочая станция, сервер и коммутатор.
3. Типы линий связи: витая пара и оптическое волокно без указания категории;
4. Типы каналов связей: дуплексный и полудуплексный.

**Разработка алгоритма построения топологической матрицы КВС**

В настоящей работе предложен и реализован *принцип мозаики*, предусматривающий модульность построения изображения и выборочное изменение отдельных модулей изображения. Каждый модуль изображения имеет номер (адрес), что упрощает его поиск при внесении изменений.

Опишем этапы построения структуры КВС:

1. По полученной из входных данных карте зданий анализируется информация об их количестве и расположении на территории;
2. Анализируются полученные из входных данных поэтажные схемы каждого здания. Данные схемы отображают местоположения комнат в здании;
3. Рассчитывается количество комнат в каждом здании по следующим формулам:

*Rобщ* =

где *Rобщ* – количество комнат на территории, *Rb* - количество комнат в здании *b*, B – количество зданий на территории

*Rb* =

здесь:

*Rj* – количество комнат на этаже j

*F* – количество этажей в корпусе k

*C* – количество корпусов в здании b

1. По полученным из входных данных: принципиальной схемы КВС и информации о местоположении устройств; по рассчитанному количеству комнат в каждом здании, строится *матрица изображения местоположения комнат корпорации (МИ)*

Каждая комната имеет адрес своего местоположения. Администратору необходимо знать, в какой комнате совершено изменение и ее местоположение, поэтому в работе предложен *принцип мозаики:* каждому изображению комнаты присвоен свой номер (номер ячейки в матрице), что существенно упрощает процедуру внесения изменений в структуре ВС.

Произведем стилизацию зданий: каждое здание на территории делится по корпусам, в которых, в свою очередь, располагаются этажи, на каждом этаже находятся комнаты. Введем такое понятие, как *уровень отображения* территории, здания, корпуса, этажа или комнаты.

На данном этапе вводится принцип *многоуровневой адресации*, который позволит определить местоположение каждой комнаты сети.

На каждом уровне вводится такое понятие, как *Таблица метаданных комнаты*, в которой в каждому набору *(B, C, F, R)* соответствует свой идентификатор *id.*

Таблица 2.1

Таблица метаданных комнаты

|  |  |  |  |
| --- | --- | --- | --- |
| B.C.F.R | 1.1.1.0 | 1.1.1.1 | 1.1.1.2 |
| id | 1 | 2 | 3 |

Таким образом, здесь используется многоуровневая адресация. Где *B, C, F* соответственно номер здания, корпуса и этажа, которые указывают местоположение комнаты с номером *R.*

Таблица метаданных на одном уровне отображения несколько отличается от таблицы другого уровня. Например, в таблице метаданных *для корпуса* в наборе *(B, C, F, R)* номер здания *(B)* и номер корпуса *(C)* являются фиксированными величинами, так как на уровне корпуса отображаются комнаты, располагающиеся на этажах корпуса *C*, местоположение которого строго определено (здание *B*).

Таблица 2.2

Таблица метаданных комнаты для корпуса

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **F.R** | 1.0 | 1.2 | 1.3 | 1.4 |
| **Id** | 1 | 2 | 3 | 4 |

где *F-*номер этажа,

*R –* номер комнаты

1. Типизация комнат. Из входных данных получаем основные типы комнат, используемых в каждом здании. Также, задана таблица соответствия *Комната - Тип*, указывающая какой тип соответствует каждой комнате здания.
2. Построенная ранее в пункте 3 матрица заполняется полученными изображениями комнат каждого типа и в результате формируется конечная *матрица изображения структуры КВС (МИ КВС).*

МИ КВС связана с топологической матрицей. При изменении топологической матрицы, необходимо также вносить изменения в структуру вычислительной сети. МИ КВС позволяет оперативно найти ту комнату, в которой произведено изменение и выполнить с ней необходимые действия.

В каждой комнате располагаются устройства, разных типов и с разным количеством разъемов. Устройства и их разъемы имеют порядковые номера, поэтому необходимо расширить многоуровневую адресацию для комнат (рассмотренную в пункте 4), добавив в нее номер устройства в комнате и номер его разъема.

Тогда, таблица метаданных, построенная в пункте 4, преобразуется в таблицу метаданных для устройств, в которой каждому набору *(B, C, F, R, D, P)* соответствует свой идентификатор *id.*

Таблица 2.3

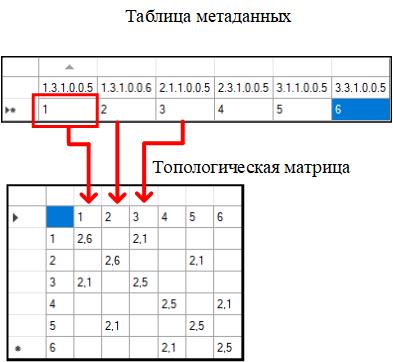
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| B.C.F.R.D.P | 1.1.1.1.0.0 | 1.1.1.1.1.1 | 1.1.1.1.2.1 | 1.1.1.1.3.1 |
| id | 1 | 2 | 3 | 4 |

Таблица метаданных устройств

где *B, C, F, R* (элементы из таблицы метаданных для комнаты) соответственно номер здания, корпуса, этажа и комнаты, которые указывают местоположение устройства с номером *D* и разъемом *P.*

**Алгоритм построения таблицы метаданных устройств**

Таблица метаданных устройств необходима для дальнейшего построения топологической матрицы вычислительной сети (Рис. 2), в которой первые столбец и строка заполняются значениями из строки *id* таблицы метаданных.



1. Связь таблицы метаданных устройств с топологической матрицей

Топологическая матрица может быть построена для каждого из пяти уровней соединений:

* Уровень территории
* Уровень здания
* Уровень корпуса
* Уровень этажа
* Уровень комнаты

Таблица метаданных устройств, также, как и таблица метаданных для комнаты на одном уровне несколько отличается от таблицы другого уровня. Например, в таблице метаданных *для этажа* в наборе *(B, C, F, R, D)* номер здания *(B)*, номер корпуса *(C)* и номер этажа *(F)* являются фиксированными величинами, так как на уровне этажа отображаются соединения между комнатами на этаже *F*, местоположение которого строго определено (здание *B,* корпус *C*).

Таблица 2.4

Таблица метаданных устройств для этажа

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **R, D, P** | 0.0.1 | 0.0.2 | 1.2.4 | 2.2.4 |
| **Id** | 1 | 2 | 3 | 4 |

где *R –* номер комнаты, *D -* номер устройства, *P-* номер разъема

Следовательно, для начала выбирается уровень соединения и устанавливаются размеры таблицы метаданных.

Затем, в зависимости от выбранного уровня соединения выполняются следующие действия:

*уровень комнаты:* поочередно для всех устройств в комнате, для каждого разъема устройства проверяется наличие у этого устройства какого-либо из соединений в комнате. Если устройство имеет соединение, то первая строка метаданных заполняется следующими данными: *D, P* – соответственно номер устройства и его разъема, полученного на данном шаге

*уровень этажа:* поочередно для всех комнат на этаже повторяются действия, аналогичные *уровню комнаты*, за исключением того, что устройство проверяется на наличие какого-либо соединения на этаже. Если устройство имеет соединение, то первая строка метаданных заполняется следующими данными: *R, D, P* – соответственно номер комнаты, номер устройства и его разъема, полученного на данном шаге

*уровень корпуса:* поочередно для всех этажей в корпусе повторяются действия, аналогичные *уровню этажа*, за исключением того, что устройство проверяется на наличие какого-либо соединения в корпусе. Если устройство имеет соединение, то первая строка метаданных заполняется следующими данными: *F, R, D, P* – соответственно номер этажа, номер комнаты, номер устройства и его разъема, полученного на данном шаге

*уровень здания:* поочередно для всех корпусов в здании повторяются действия, аналогичные *уровню корпуса*, за исключением того, что устройство проверяется на наличие какого-либо соединения в здании. Если устройство имеет соединение, то первая строка метаданных заполняется следующими данными: *С, F, R, D, P* – соответственно номер корпуса, номер этажа, номер комнаты, номер устройства и его разъема, полученного на данном шаге

*уровень территории:* поочередно для всех зданий на территории повторяются действия, аналогичные *уровню здания*, за исключением того, что устройство проверяется на наличие какого-либо соединения на территории. Если устройство имеет соединение, то первая строка метаданных заполняется следующими данными: *B, С, F, R, D, P* – соответственно номер корпуса, номер этажа, номер комнаты, номер устройства и его разъема, полученного на данном шаге

Количество повторений описанных действий зависят от количества зданий, корпусов, этажей, комнат и устройств, а также от количества соединений каждого уровня.

В связи с тем, что на каждом уровне соединений применяется свой алгоритм так называемого «пути» прохождения по нижним уровням, на схеме алгоритма построения таблицы метаданных для компактного представления обозначим эти алгоритмы как: «алгоритм пути для уровня комнаты», «алгоритм пути для уровня этажа» и т.д.

Схема описанного алгоритма представлена на Рис. 3

|  |
| --- |
|  |
|  |

Рис. 1. Алгоритм построения таблицы метаданных

**Алгоритм построения топологической матрицы вычислительной сети**

Когда таблица метаданных для выбранного уровня построена начинается переход к следующему этапу – построению топологической матрицы вычислительной сети.

Для начала, устанавливаются размеры матрицы, затем первый столбец и строка топологической матрицы заполняются значениями из строки *id* таблицы метаданных.

Далее, проходя по всем строкам *(i)* и столбцам *(j)* ячейки матрицы заполняются по следующему алгоритму:

1. Если *i = j*, то в соответствующей ячейке матрицы записывается следующая пара значений: (*DeviceType, PortsCount*), где *DeviceType –* числовое значение, соответствующее определенному типу устройства, *PortsCount –* количество разъемов устройства, местоположение которого определяется из ячейки таблицы метаданных в первой строке, а номер столбца *id* равен  *i = j*.
2. Если *i ≠ j*, то все соединения выбранного уровня проверяются на принадлежность разъемам устройств, местоположения которых определяются из ячеек таблицы метаданных: для первого устройства: *MetaTable[1, i]*, для второго устройства: *MetaTable[1, j].*

Если разъемы данных двух устройств имеют соединение между собой, то на пересечении *i-ой* строки и *j-го* столбца записывается следующая пара значений: (*LineType, ChannelType*), где *LineType* – числовое значение, соответствующее определенному типу линии связи, *ChannelType -* числовое значение, соответствующее определенному типу канала связи.

Если разъемы данных двух устройств не имеют соединение между собой, то осуществляется переход к следующему столбцу матрицы.

После прохода по последней строке топологической матрицы, осуществляется ее отображение для пользователя.

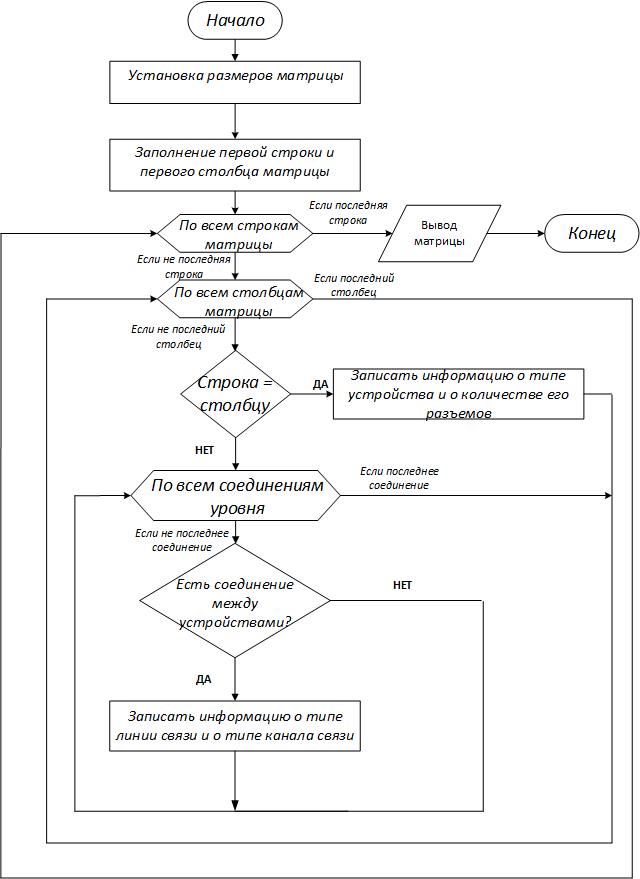


Рис. 4 Схема алгоритма построения топологической матрицы

**Особенности реализации алгоритмов**

Для решения этих задач было создано приложение для Windows на языке C#, с помощью интерфейса программирования Windows Forms, в среде Visual Studio 2015.

В настоящей работе было принято решение выбрать объектно-ориентированный подход программирования, так как в программе необходимо создать модель вычислительной сети, а объектно-ориентированный подход наиболее адекватно отражает реальный мир, представляющий собой совокупность взаимодействующих объектов.

В настоящей работе разработано приложение для персонального компьютера с операционной системой *Windows*, представляющее собой интерфейс для работы с топологической матрицей вычислительной сети. Разработка основной части приложения произведена на языке *С#* с применением среды разработки *Visual Studio*. Некоторая необходимая информация для работы с приложением хранится в таблицах *Excel.*

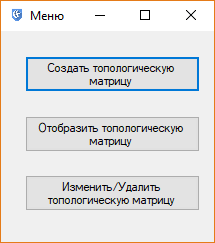
Пользователем приложения является системный администратор. Предполагается, он знает, как работать с приложением.

Для того, чтобы отобразить топологическую матрицу, необходимо иметь информацию об устройствах вычислительной сети и о соединениях между ними.

Главная форма (Меню) содержит 4 элемента класса *Button*:

* *Создать топологическую матрицу (CreateMatrix)* – при нажатии на данную кнопку открывается форма создания топологической матрицы *(CreateMatrix.Form);*
* *Отобразить топологическую матрицу (ShowMatrix)* – при нажатии на данную кнопку вызывается функция *ShowM*, которая формирует матрицу и создает необходимые элементы для ее отображения;
* *Изменить топологическую матрицу/Удалить топологическую матрицу (ChangeMatrix)* – при нажатии на данную кнопку открывается форма изменения и удаления топологической матрицы *(ChangeMatrix.Form);*

При запуске приложения перед пользователем появляется меню. Предполагается, пользователь одновременно работает с приложением и с матрицей изображения КВС (Рис.5).



**Тестирование**

***тестовое построение матрицы изображений КВС***

*Входные данные для теста*:

1) Карта зданий корпорации

2) Поэтажная схема каждого здания

3) Принципиальная схема КВС территории

4) Таблица местоположения устройств

5) Каталог изображений основных типов комнат

6) Таблица соответствия «Комната – Тип»

Рассмотрим этапы построения структуры КВС, на примере заданного теста.

Задана следующая карта зданий корпорации:

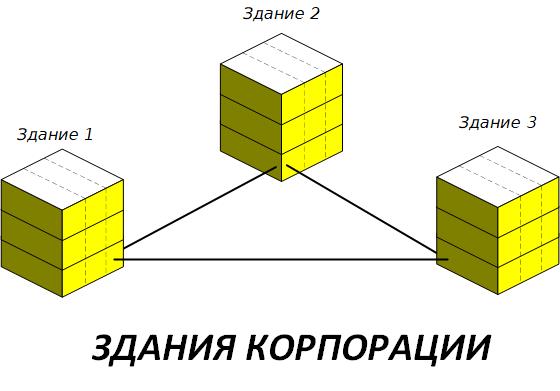


Рис.6 Карта зданий корпорации

Дано 3 здания. Известно, что в каждом здании имеется 3 корпуса, в каждом корпусе по 3 этажа, а на каждом этаже 7 комнат.

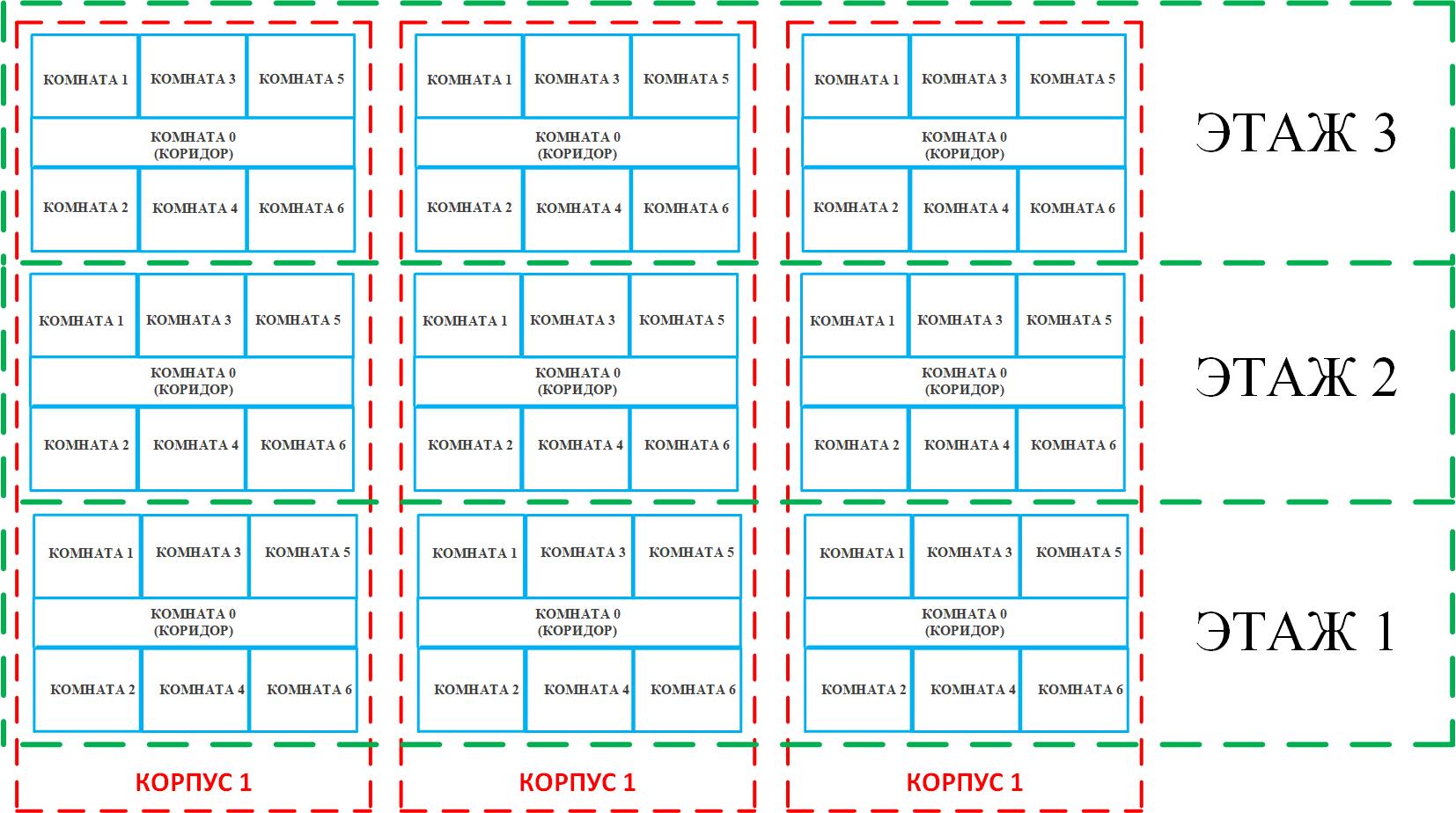
 Задана схема типового здания:

Рис. 7 Поэтажный план типового здания

На Рис. 7 отображена детальная схема здания с точностью до местоположения комнат. В данном тесте все три здания на территории одинаковые, поэтому схема справедлива для всех зданий.

Расчёт количества комнат

По формулам (1) и (2) рассчитаем количество комнат в каждом здании.

Для данного теста в одном здании: R1 = = (7 \* 3)\*3 = 63

В первом здании 63 комнаты. В работе все три здания являются идентичными по структуре, следовательно, учитывая количество зданий, имеем: *Rобщ* = = = 63\*3 = 189.

Всего: 189 комнат.

Построение матрицы изображения местоположения комнат корпорац**ии**

Обозначения, принятые в следующих схемах:

Corp - корпус здания;

G1 – группа, состоящая из двух рабочих станций;

G2 – группа, состоящая из трёх рабочих станций;

S (Switch) – коммутатор;

SR (Server) – сервер;

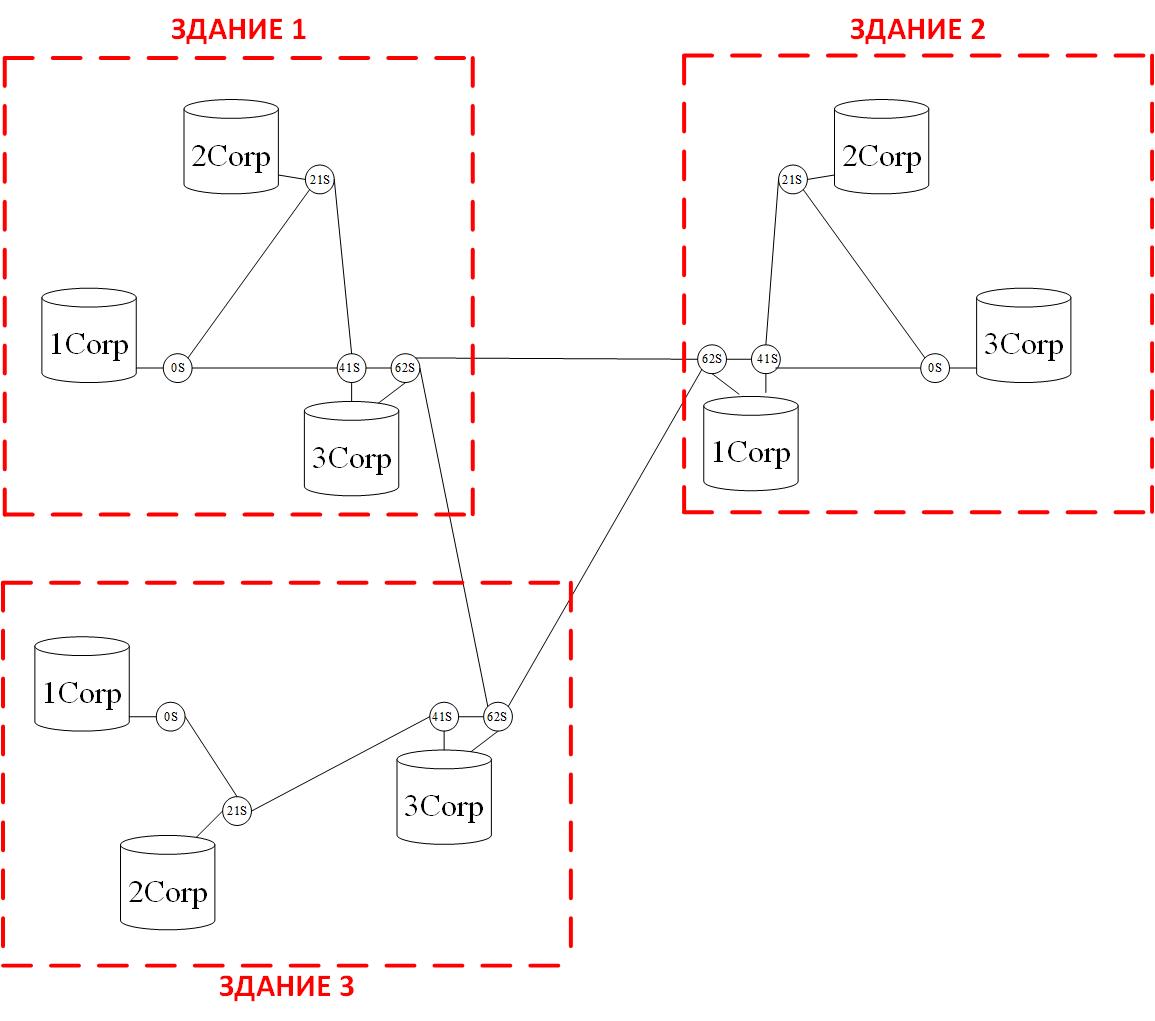
Задана принципиальная схема КВС территории**:**

Рис. 8 Принципиальная схема КВС территории

Задана принципиальная схема КВС корпуса**:**

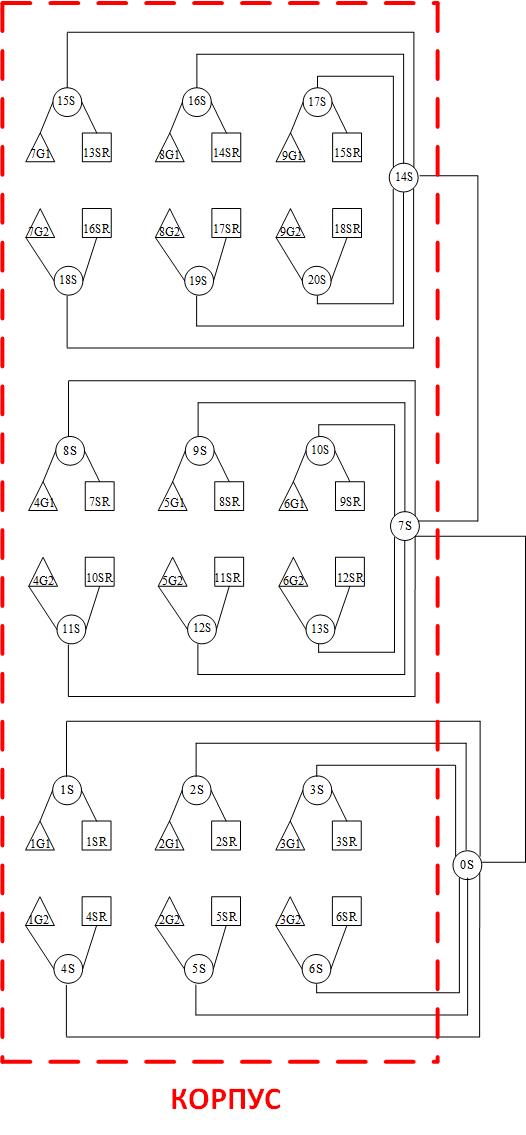


Рис. 9 Принципиальная схема КВС корпуса

В данном тесте все корпуса зданий одинаковые, поэтому данная схема справедлива для всех корпусов (Corp).

Задана информация о местоположении устройств и групп в здании:

Таблица 4.1

Местоположение устройств и групп в здании

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование | Корпус | Этаж | Комната |
| 1G1 | 1 | 1 | 1 |
| 1SR | 1 | 1 | 1 |
| 1S | 1 | 1 | 1 |
| 2G1 | 1 | 1 | 2 |
| 2SR | 1 | 1 | 2 |
| 0S | 1 | 1 | 0 |
| … | … | … | … |
| 62S | 3 | 1 | 0 |

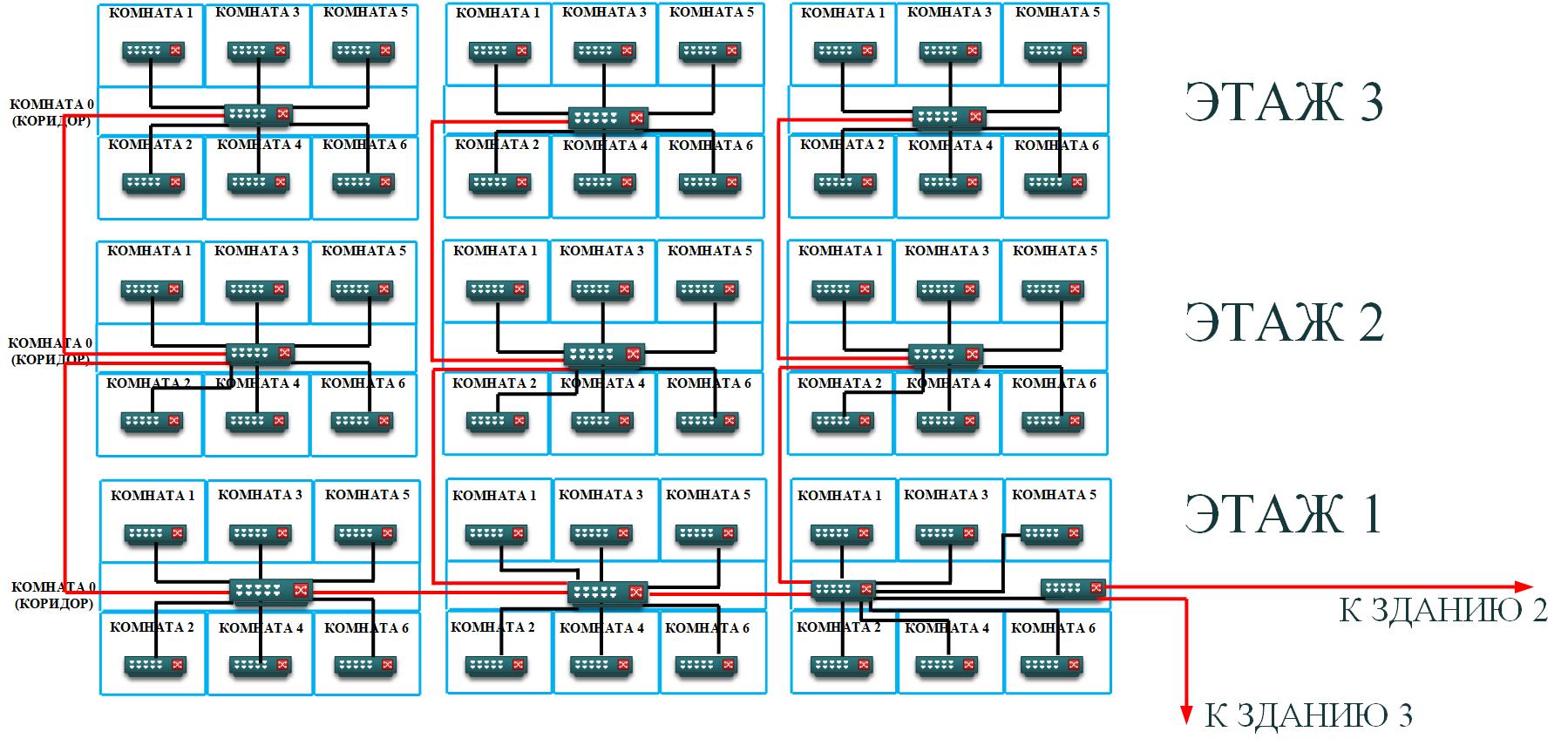
Объединяя заданные данные получаем следующую схему для здания:

Рис. 9 Детальная схема поэтажного плана зданий

На данной схеме отображены соединения между уровнями (межкомнатные, межэтажные, межкорпусные) без подробного описания каждой комнаты.

Матрица изображения местоположения комнат корпорации

Матрица состоит из изображений комнат каждого здания. Элементы матрицы – комнаты. Каждой ячейке соответствует комната (Рис. 32).

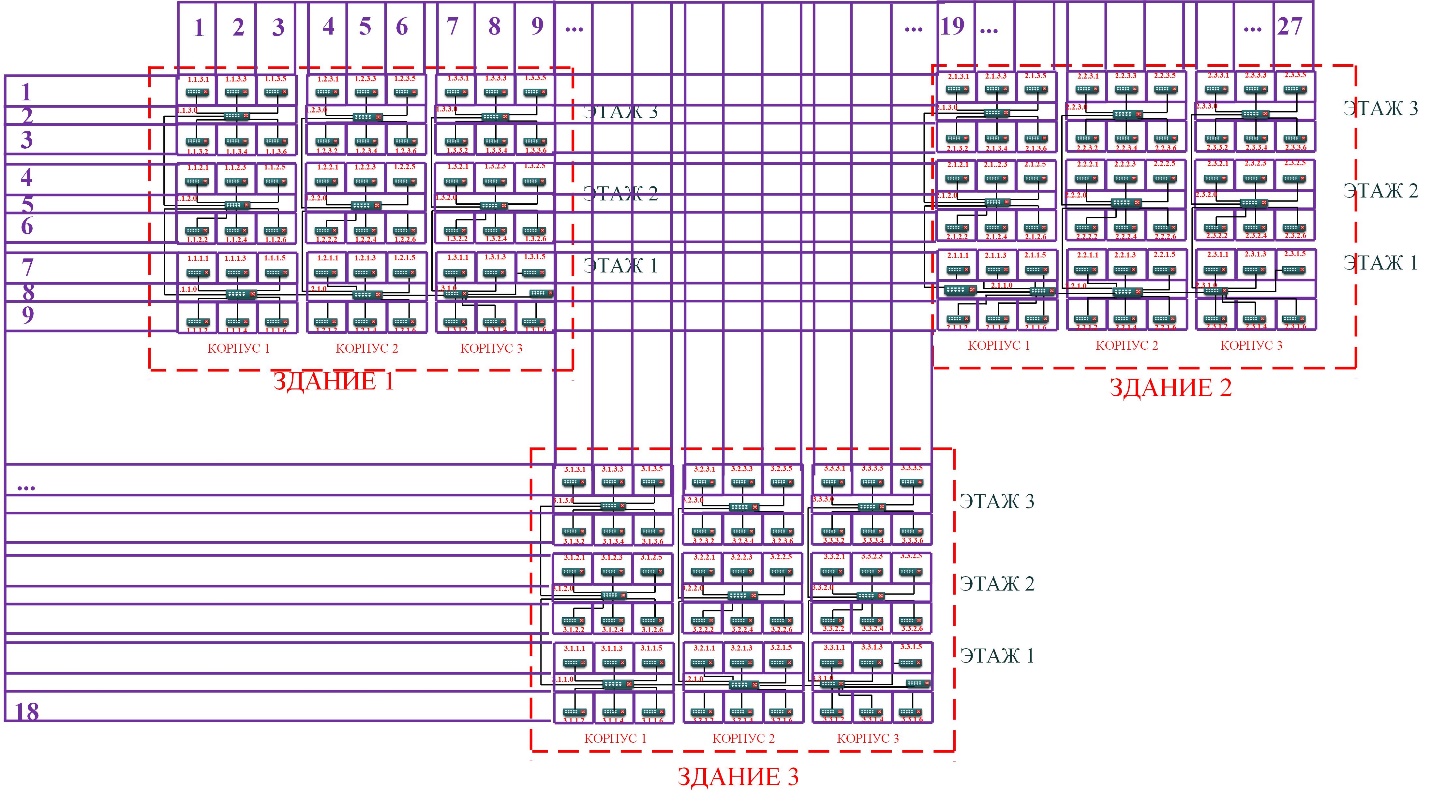
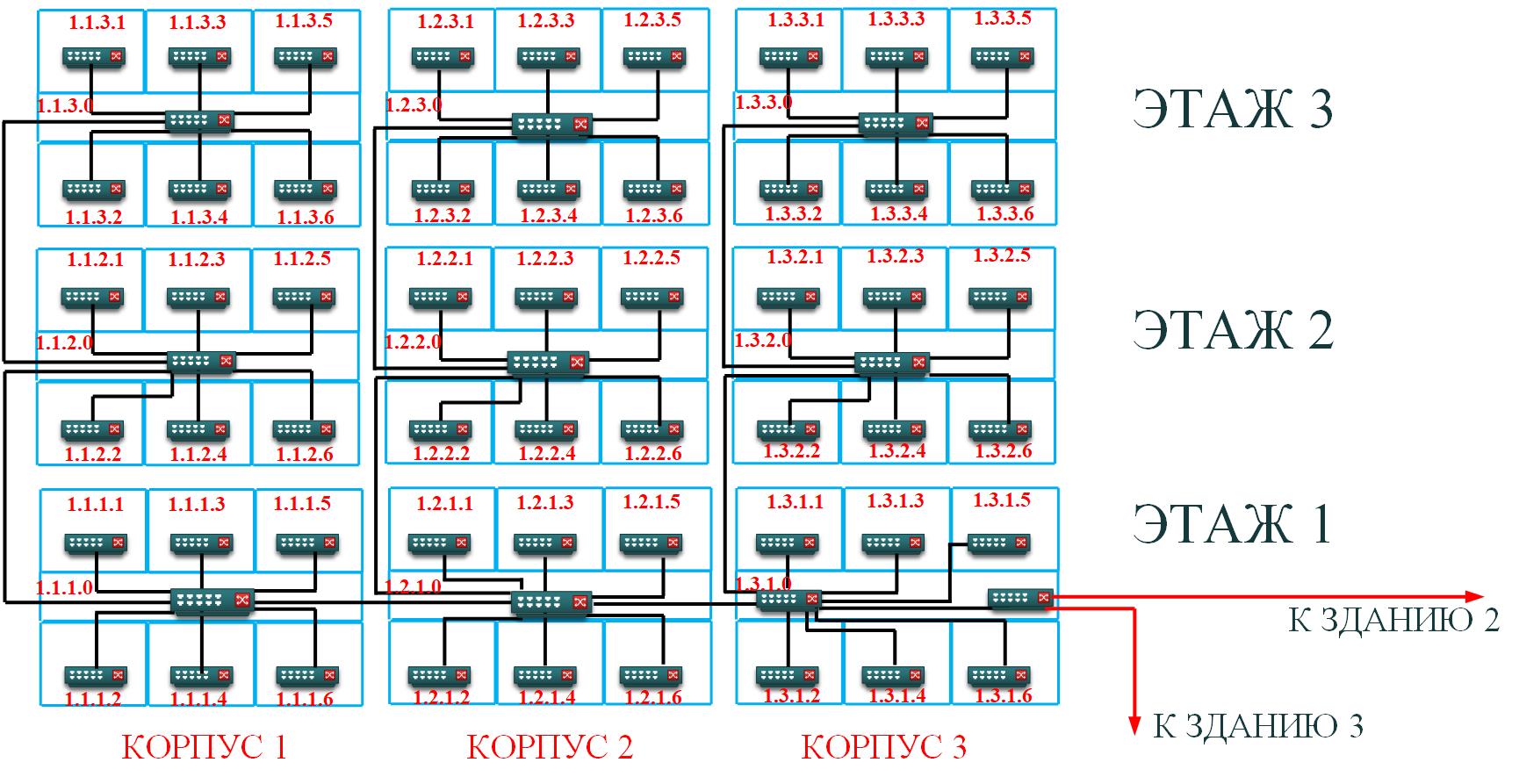


Рис.10 Матрица изображений местоположения комнат

Матрица имеет размерность 18\*27. Некоторым комнатам, а именно коридорам соответствует 3 ячейки. Например элементы {2,1},{2,2},{2,3} соответствуют одной и той же комнате – Коридор в зданиии1, корпусе1, этаж3, комната0.

Многоуровневая адресация**.**

Рассмотрим рисунок для одного здания (Рис. 10 ) И присвоим каждой комнате следующий номер: *B.C.F.R,* здесь *B* - номер здания, в котором располагается комната, *C* - номер корпуса, *F* - номер этажа, *R –* номер самой комнаты(Рис. 33).

Рис. 10 Многоуровневая адресация в здании

Как видно из рисунка, такая многомерная адресация не является удобной из-за большой длины.

Таблица метаданных комнаты (для корпуса)

Рассмотрим построение таблицы метаданных для корпуса 1, здания 1.

Как было рассчитано ранее, корпус 1 имеет 21 комнату, построим таблицу метаданных:

Таблица 4.2

Таблица метаданных комнаты для корпуса

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **F.R** | 1.0 | 1.1 | 1.2 | 1.3 | 1.4 | 1.5 | 1.6 | 2.0 | 2.1 | 2.2 | 2.3 | … | 3.6 |
| **Id** | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | … | 21 |

Соответственно схема корпуса 1, здания 1 значительно упрощается (Рис. 34):

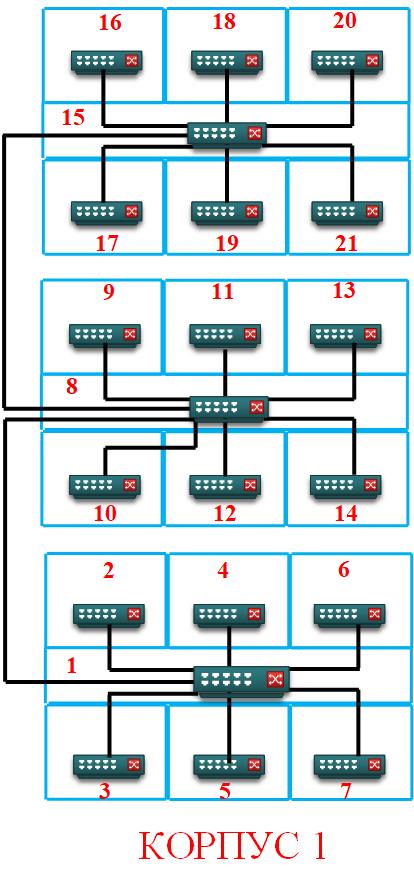


Рис. 11 Применение таблицы метаданных комнаты для корпуса

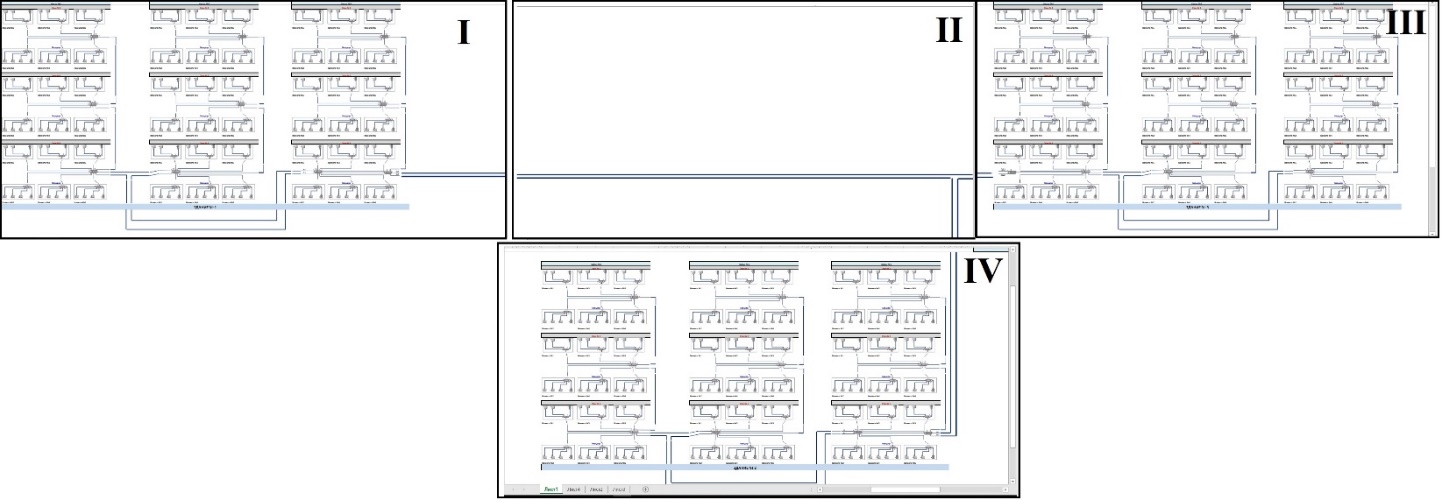
Таблица метаданных упрощает адресацию в пределах одного уровня

Построение *матрицы изображения структуры КВС*

Заполняем построенную в 3 этапе матрицу изображениями комнат соответствующих типов при помощи таблицы соответствия «Комната – Тип»

Для реализации *принципа мозаики* используется *Excel*, в котором каждому изображению комнаты соответствует ячейка с определенным номером.

Построенную матрица изображения КВС на экране пользователя можно представить изображением, состоящим из четырех модулей (Рис. 35)

1. Матрица изображения КВС в общем виде

Рассмотрим подробно каждый модуль изображения:

1) Первый модуль (I) отображает принципиальную схему КВС первого здания;

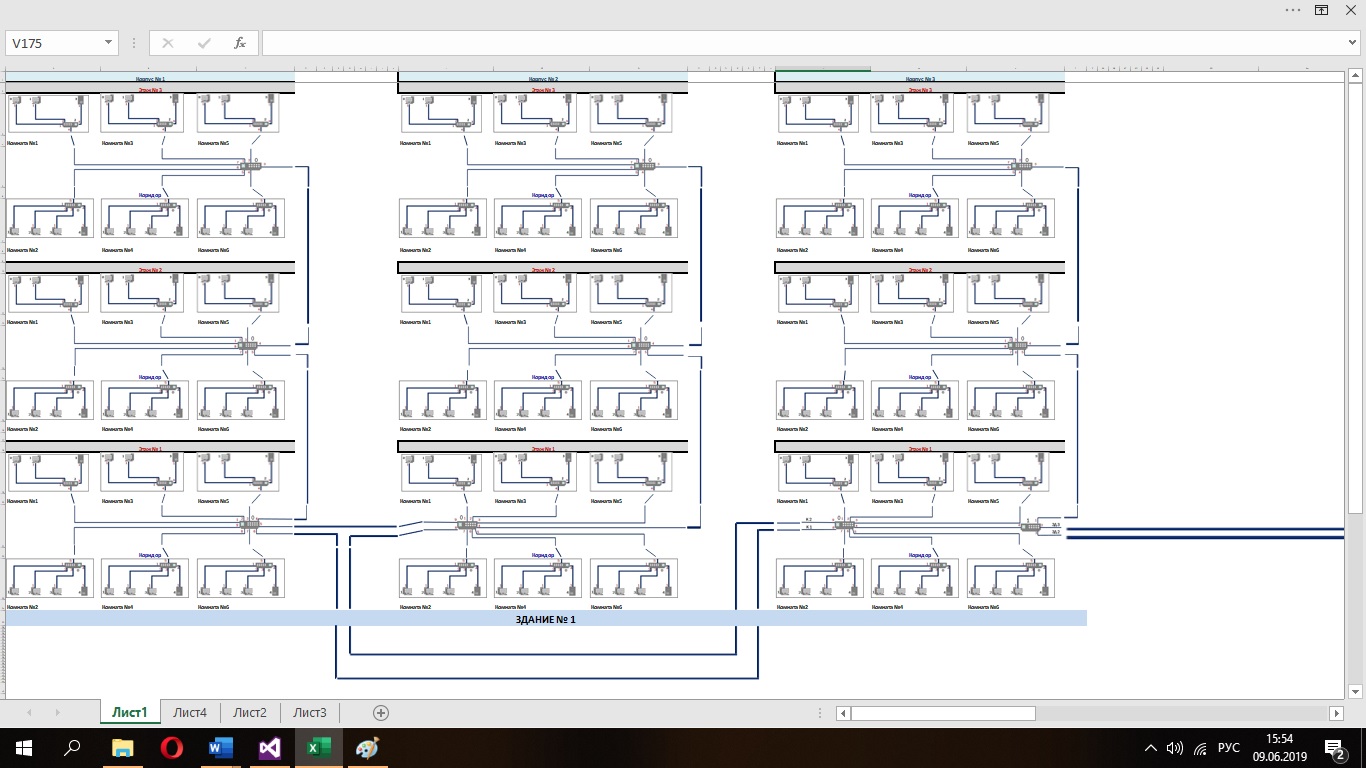
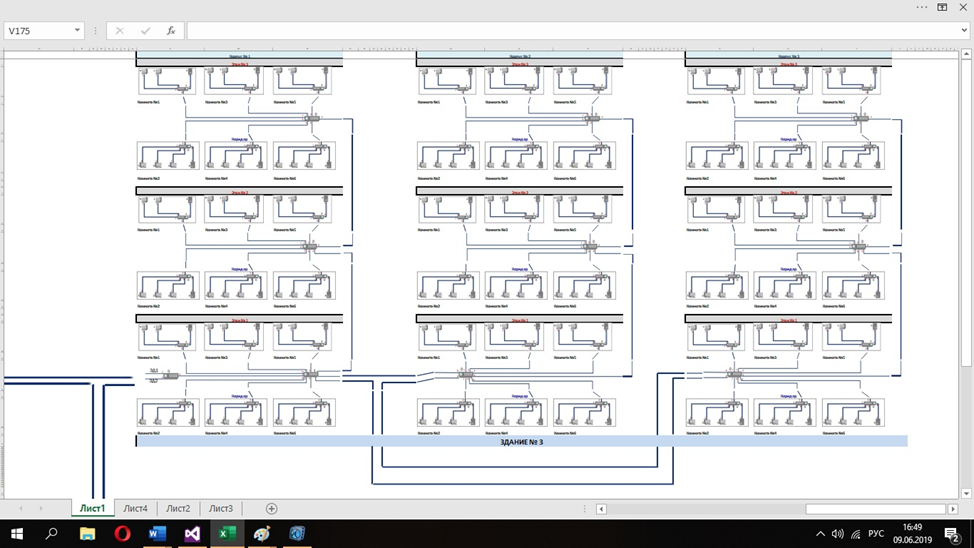


Рис. 12 Принципиальная схема КВС первого здания

2) Второй модуль (II) отображает некоторые линии соединения между зданиями;

3) Третий модуль (III) отображает принципиальную схему КВС третьего здания;



13 Принципиальная схема КВС третьего здания

4) Четвертый модуль (IV) отображает принципиальную схему КВС второго здания;

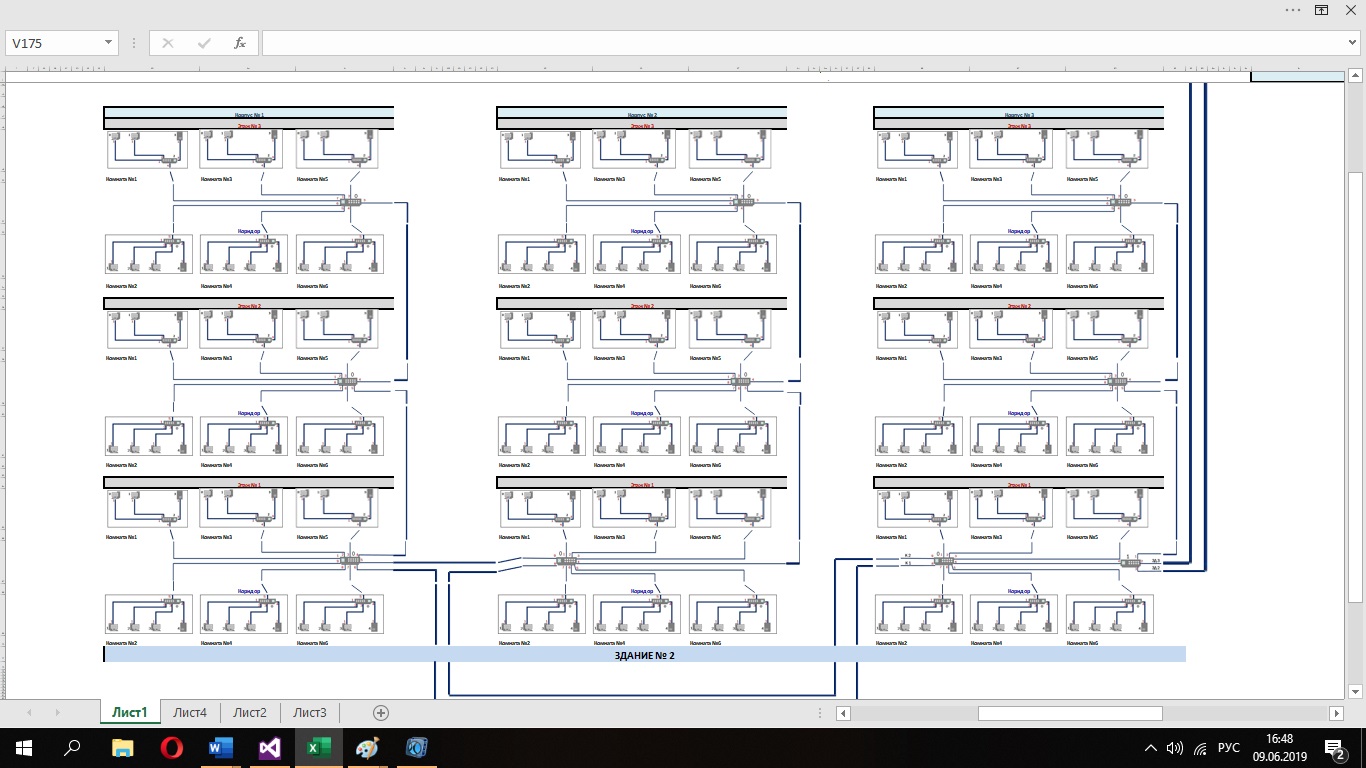


Рис. 14 Принципиальная схема КВС второго здания

Таблицы *Excel* позволяют: применять масштабирование; переходить к ячейке с определенным номером; оперативно производить вставку изображения в ячейку.

**Литература**

1. Абросимов Л.И. Базисные методы проектирования и анализа сетей ЭВМ: учебное пособие / Л.И. Абросимов. – М.: Университетская книга, 2015.- с. 227-235.

2. Руководство по программированию на языке C# [Электронный ресурс] // Официальный сайт компании Microsoft. URL: https://docs.microsoft.com/ru-ru/dotnet/csharp/programming-guide/index

3. Шилдт Герберт, Руководство для начинающих. - Osborne/McGraw Hill, 2001.

4. Джеффри Рихтер, CLR via C#. Программирование на платформе Microsoft.NET Framework 4.5 на языке C#. – Изд-во Питер, 2017. – 896 с.

5. Шилдт Герберт, C# 4.0: Полное руководство. - ООО "И.Д. Вильямс", 2011. – 1056 с.