ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ

**БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ НАЦИОНАЛЬНЫЙ**

**ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ**

(НИУ «БелГУ»)

ИНСТИТУТ ИНЖЕНЕРНЫХ И ЦИФРОВЫХ ТЕХНОЛОГИЙ

КАФЕДРА МАТЕМАТИЧЕСКОГО

И ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ИНФОРМАЦИОННЫХ СИСТЕМ

**Отчет по лабораторной работе 4**

**по дисциплине: «Компьютерные сети»**

студента очного отделения

4 курса 12001801 группы

Капустина Виктора Сергеевича

Проверил(а):

Маматов Евгений Михайлович

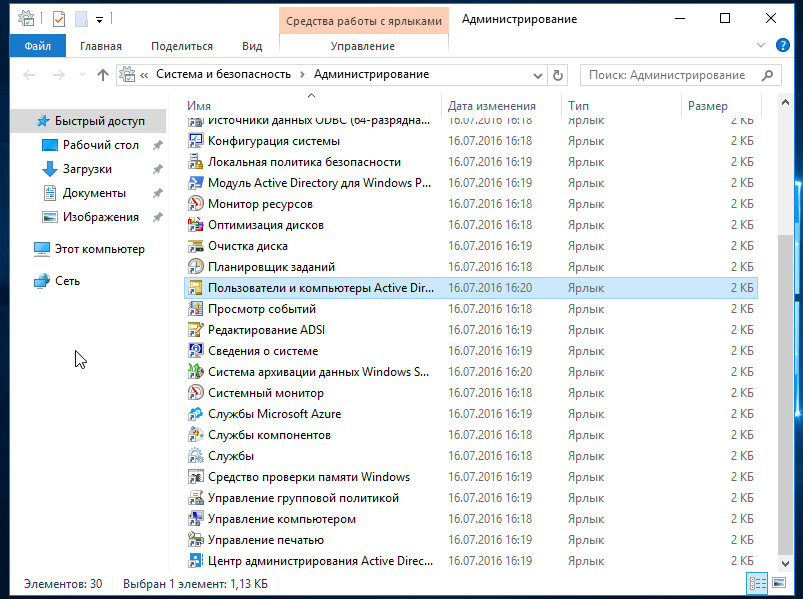
Белгород 2022

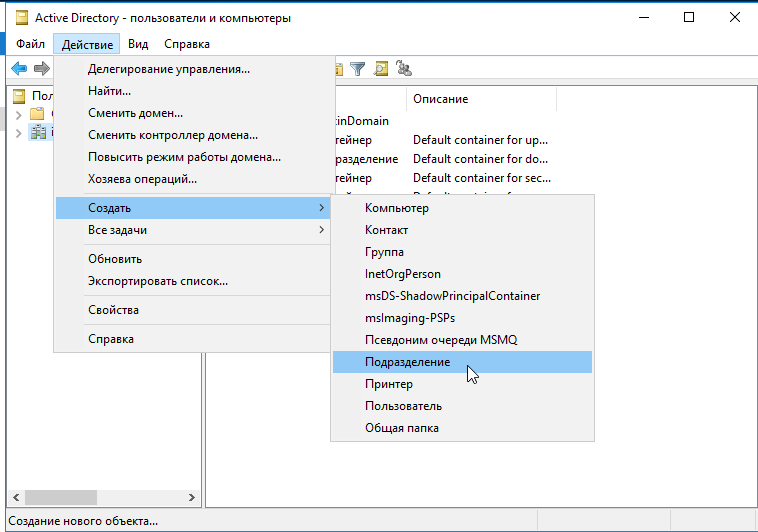
**Лабораторная работа 4**

**Цель работы:** создание объектов в Active Directory Windows 2016 Server.

**Ход работы**

**Задание 1**

  
Рис. 1 – Вход в оснастку Пользователи и компьютеры Active Directory

  
Рис. 2 Создание подразделения

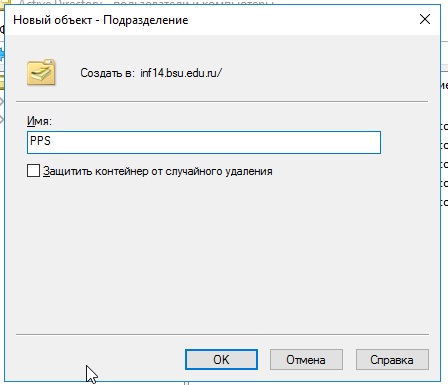
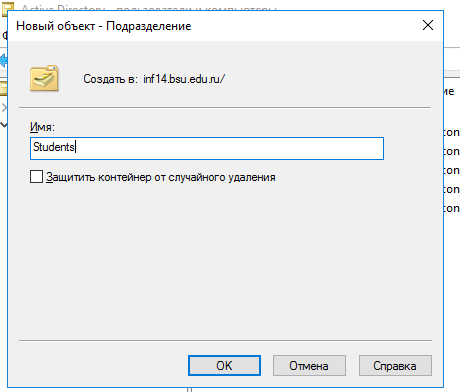
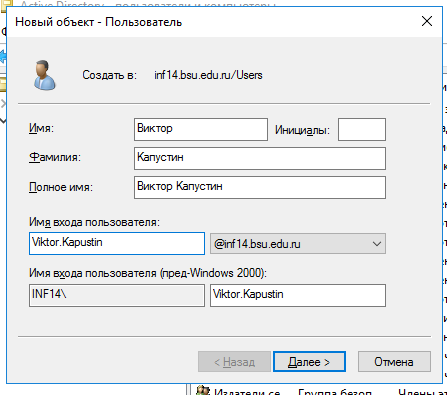
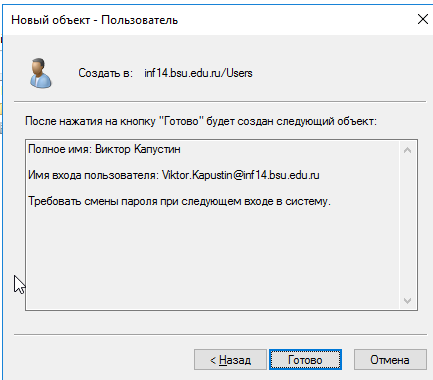
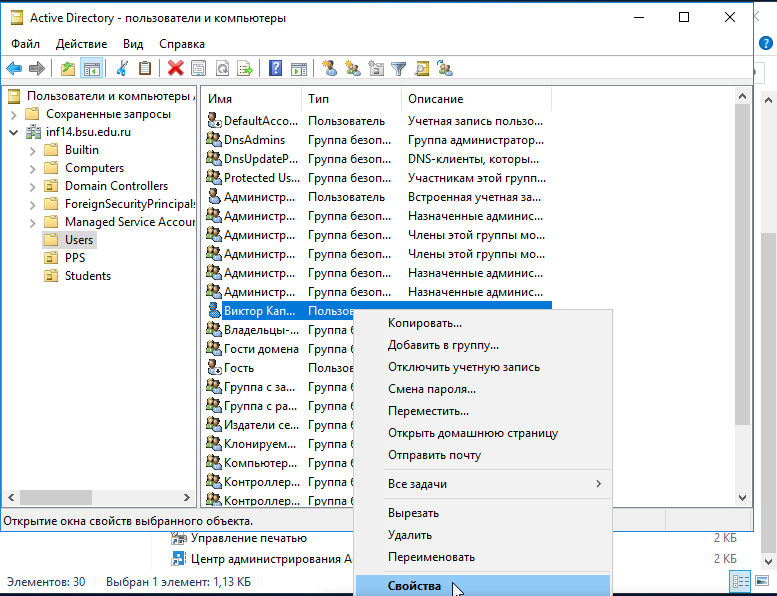


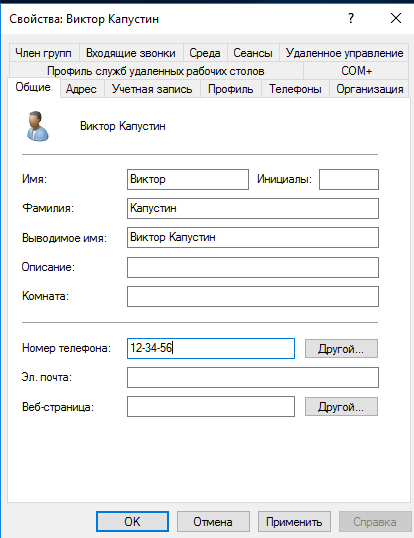
Рис. 3 создание подразделения PPS

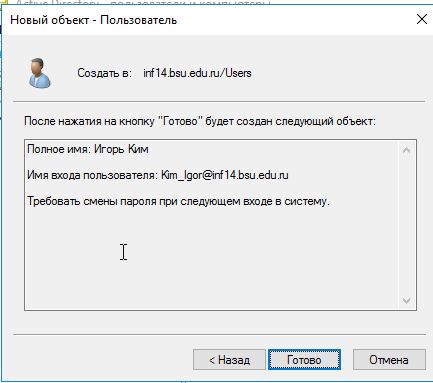
  
Рис. 4 Создание подразделения Students

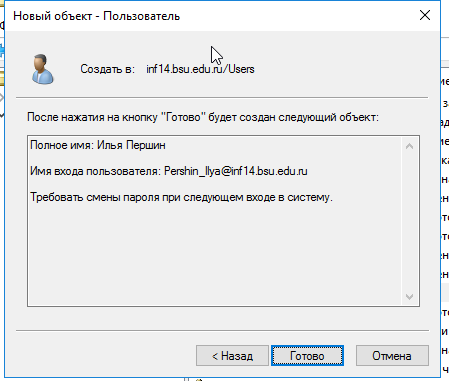
  
Рис. 5 Создание пользователя

  
Рис. 6 Подтверждение создания нового пользователя

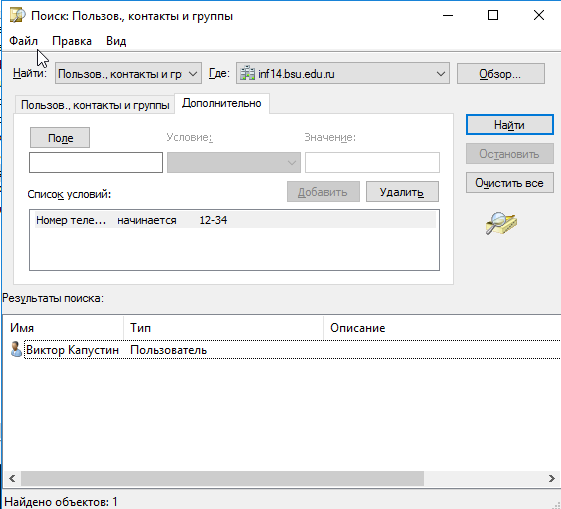
  
Рис. 7 Выбор нового пользователя

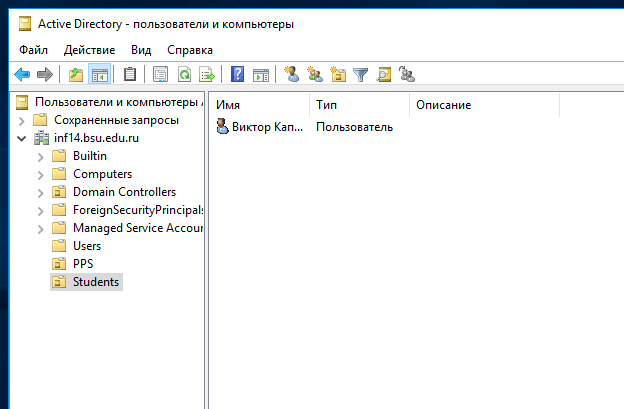
  
Рис. 8 Настройка свойств нового пользователя

  
Рис. 9 Создание первого нового пользователя

  
Рис. 10 Создание второго нового пользователя

**Задание 2**

****Рис. 11 Найден ранее созданный пользователь

  
Рис. 12 Результат перемещения пользователя в ОП Students

**Вывод:** на данной лабораторной научился создавать объекты в Active Directory Windows 2016 Server.

**Ответы на контрольные вопросы:**

**1.** **Объекты Active Directory, их описание.**

| Объект | Описание |
| --- | --- |
| Компьютер | Представляет компьютер в сети. Для компьютеров Windows NT Workstation и Windows NT Server это будет учетная запись компьютера. Объект содержит сведения о компьютере, входящем в домен. |
| Контакт | Учетная запись без ограничений по доступу. Объекты Contact обычно применяются для представления внешних пользователей в электронной почте. |
| Группа | Включает пользователей, компьютеры и другие группы, упрощает обслуживание большого числа объектов. |
| Принтер | Сетевой принтер, опубликованный в каталоге. В действительности это ссылка на подключенный к компьютеру принтер. |
| Пользователь | Главный элемент системы безопасности в каталоге. Данные, содержащиеся в этом объекте, позволяют пользователю входить в систему Windows 2003. Содержит множество дополнительных полей, таких как имя, фамилия, псевдоним и адрес электронной почты. |
| Общая папка | Общий ресурс, опубликованный в каталоге. На самом деле это ссылка на общую папку - содержит адрес данных, а не сами данные. |

**2.** **Чем обусловлена необходимость создания ОП?**

ОП создаются для упрощения администрирования сети.

Вы должны создавать ОП в следующих случаях:

• Чтобы предоставить административные полномочия другим пользователям или адми­нистраторам;

• Для группирования объектов, над которыми выполняются сходные административные операции; это облегчает поиск сходных сетевых ресурсов и их обслуживание – так, можно объединить в одном ОП все объекты User для временных служащих;

• Для ограничения видимости сетевых ресурсов в хранилище Active Directory – пользо­ватели увидят только те объекты, к которым имеют доступ; разрешения для ОП можно легко изменить, ограничив доступ к конфиденциальной информации.

**3.** **Разрешения Active Directory.**

Разрешения Active Directory обеспечивают защиту ресурсов, позволяя управлять доступом к экземплярам объектов или атрибутам объектов и определять вид предоставляемого доступа.

Разрешения зависят от типа объекта — например, разрешение Reset Password допустимо для объектов User, но не для объектов Computer.

Пользователь может быть членом нескольких групп с разными разрешениями для каж­дой их них, обеспечивающих разные уровни доступа к объектам.

Вы можете предоставлять или аннулировать разрешения. Аннулированные разреше­ния для пользователей и групп приоритетнее любых выданных разрешений. Если Вы за­претили пользователю обращаться к объекту, то он не получит доступ к нему даже как член полномочной группы. Аннулировать разрешения следует, только если требуется зап­ретить какое-то действие для конкретного пользователя, входящего в полномочную группу.

Настроить разрешения объектов и их атрибутов позволяет оснастка Active Directory – пользователи и компьютеры. Назначить разрешения также можно на вкладке Безопасность (Security) диалогового окна свойств объекта.

Для выполнения большинства задач администрирования достаточно стандартных раз­решений. Впрочем, Вам могут потребоваться и специальные разрешения. Чтобы их на­строить, щелкните на вкладке Безопасность кнопку Дополнительно. На вкладке Разрешения (Permissions) щелкните интересующее Вас разрешение, а затем — кнопку Показать/Изменить (View/ Edit). Просмотреть разрешения конкретных атрибутов можно на вкладке свойств диалогового окна Элемент разрешения (Permission Entry).

Наследование разрешений в Active Directory упрощает настройку доступа к объектам– *Вы* можете применить разрешения к дочерним объектам текущего объекта. Для отмены наследования разрешений сбросьте флажок Переносить наследуемые от родительского объекта разрешений на этот объект, тогда объект получит лишь те разрешения, которые Вы явно ему назначили. Предотвратить наследование можно на вкладке Безопасность диалогового окна свойств объекта.

При запрещении наследования можно:

·скопировать ранее унаследованные разрешения — набор новых явных разрешений объекта будет копией разрешений, которые он ранее наследовал от родительского объекта; потом разрешения можно скорректировать явно;

·удалить унаследованные разрешения – Windows 2016 удаляет все унаследованные раз­решения, т. е. для объекта вообще не будут определены разрешения; потом для него можно явно указать любые разрешения.

**4.** **Управление памятью. Типы адресов.**

**Управление памятью**

Память является важнейшим ресурсом, требующим тщательного управления со стороны ОС. Распределению подлежит вся оперативная память, не занятая операционной системой. Обычно ОС располагается в самых младших адресах, однако может занимать и самые старшие адреса. Функциями ОС по управлению памятью являются:

· отслеживание свободной и занятой памяти,

· выделение памяти процессам и освобождение памяти при завершении процессов,

· вытеснение процессов из оперативной памяти на диск, когда размеры основной памяти не достаточны для размещения в ней всех процессов, и возвращение их в оперативную память, когда в ней освобождается место,

· настройка адресов программы на конкретную область физической памяти.

**Типы адресов**

Для идентификации переменных и команд используются символьные имена (метки), виртуальные адреса и физические адреса.

Символьные имена присваивает пользователь при написании программы на алгоритмическом языке или ассемблере.

Виртуальные адреса вырабатывает транслятор, переводящий программу на машинный язык. Совокупность виртуальных адресов процесса называется *виртуальным адресным пространством*. Каждый процесс имеет собственное виртуальное адресное пространство. Максимальный размер виртуального адресного пространства ограничивается разрядностью адреса, присущей данной архитектуре компьютера, и, как правило, не совпадает с объемом физической памяти, имеющимся в компьютере.

Физические адреса соответствуют номерам ячеек оперативной памяти, где в действительности расположены или будут расположены переменные и команды. Переход от виртуальных адресов к физическим может осуществляться двумя способами. В первом случае замену виртуальных адресов на физические делает специальная системная программа - перемещающий загрузчик. Перемещающий загрузчик на основании имеющихся у него исходных данных о начальном адресе физической памяти, в которую предстоит загружать программу, выполняет загрузку программы, совмещая ее с заменой виртуальных адресов физическими.

Второй способ заключается в том, что программа загружается в память в неизмененном виде в виртуальных адресах, при этом операционная система фиксирует смещение действительного расположения программного кода относительно виртуального адресного пространства. Во время выполнения программы при каждом обращении к оперативной памяти выполняется преобразование виртуального адреса в физический. Второй способ является более гибким, он допускает перемещение программы во время ее выполнения, в то время как перемещающий загрузчик жестко привязывает программу к первоначально выделенному ей участку памяти. Вместе с тем использование перемещающего загрузчика уменьшает накладные расходы, так как преобразование каждого виртуального адреса происходит только один раз во время загрузки, а во втором случае - каждый раз при обращении по данному адресу.

**5.** **Методы распределения памяти**

**Распределение памяти фиксированными разделами**

Самым простым способом управления оперативной памятью является разделение ее на несколько разделов фиксированной величины. Это может быть выполнено вручную оператором во время старта системы или во время ее генерации. Очередная задача, поступившая на выполнение, помещается либо в общую очередь, либо в очередь к некоторому разделу.

При очевидном преимуществе - простоте реализации - данный метод имеет существенный недостаток - жесткость. Даже если программа имеет небольшой объем, она будет занимать весь раздел, что приводит к неэффективному использованию памяти. С другой стороны, даже если объем оперативной памяти машины позволяет выполнить некоторую программу, разбиение памяти на разделы не позволяет сделать этого.

##### **Распределение памяти разделами переменной величины**

В этом случае память машины не делится заранее на разделы. Сначала вся память свободна. Каждой вновь поступающей задаче выделяется необходимая ей память. Если достаточный объем памяти отсутствует, то задача не принимается на выполнение и стоит в очереди. После завершения задачи память освобождается, и на это место может быть загружена другая задача.

Задачами операционной системы при реализации данного метода управления памятью является:

* ведение таблиц свободных и занятых областей, в которых указываются начальные адреса и размеры участков памяти,
* при поступлении новой задачи - анализ запроса, просмотр таблицы свободных областей и выбор раздела, размер которого достаточен для размещения поступившей задачи,
* загрузка задачи в выделенный ей раздел и корректировка таблиц свободных и занятых областей,
* после завершения задачи корректировка таблиц свободных и занятых областей.

Программный код не перемещается во время выполнения, то есть может быть проведена единовременная настройка адресов посредством использования перемещающего загрузчика.

Выбор раздела для вновь поступившей задачи может осуществляться по разным правилам, таким, например, как "первый попавшийся раздел достаточного размера", или "раздел, имеющий наименьший достаточный размер", или "раздел, имеющий наибольший достаточный размер". Все эти правила имеют свои преимущества и недостатки.

По сравнению с методом распределения памяти фиксированными разделами данный метод обладает гораздо большей гибкостью, но ему присущ очень серьезный недостаток - *фрагментация памяти*. Фрагментация — это наличие большого числа несмежных участков свободной памяти очень маленького размера (фрагментов). Настолько маленького, что ни одна из вновь поступающих программ не может поместиться ни в одном из участков, хотя суммарный объем фрагментов может составить значительную величину, намного превышающую требуемый объем памяти.

##### **Страничное распределение**

Виртуальное адресное пространство каждого процесса делится на части одинакового, фиксированного для данной системы размера, называемые виртуальными страницами.

Вся оперативная память машины также делится на части такого же размера, называемые физическими страницами.

Размер страницы обычно выбирается равным степени двойки для упрощения механизма преобразования адресов.

При загрузке процесса часть его виртуальных страниц помещается в оперативную память, а остальные - на диск. При загрузке операционная система создает для каждого процесса информационную структуру - таблицу страниц, в которой устанавливается соответствие между номерами виртуальных и физических страниц для страниц, загруженных в оперативную память, или делается отметка о том, что виртуальная страница выгружена на диск.

При каждом обращении к памяти происходит чтение из таблицы страниц информации о виртуальной странице, к которой произошло обращение. Если данная виртуальная страница находится в оперативной памяти, то выполняется преобразование виртуального адреса в физический. Если же нужная виртуальная страница в данный момент выгружена на диск, то происходит так называемое страничное прерывание. Выполняющийся процесс переводится в состояние ожидания, и активизируется другой процесс из очереди готовых. Параллельно программа обработки страничного прерывания находит на диске требуемую виртуальную страницу и пытается загрузить ее в оперативную память. Если в памяти имеется свободная физическая страница, то загрузка выполняется немедленно, если же свободных страниц нет, то решается вопрос, какую страницу следует выгрузить из оперативной памяти.

В данной ситуации может быть использовано много разных критериев выбора, наиболее популярные из них следующие:

* дольше всего не использовавшаяся страница,
* первая попавшаяся страница,
* страница, к которой в последнее время было меньше всего обращений.

В некоторых системах используется понятие рабочего множества страниц. Рабочее множество определяется для каждого процесса и представляет собой перечень наиболее часто используемых страниц, которые должны постоянно находиться в оперативной памяти и поэтому не подлежат выгрузке.

После того, как выбрана страница, которая должна покинуть оперативную память, анализируется ее признак модификации (из таблицы страниц). Если выталкиваемая страница с момента загрузки была модифицирована, то ее новая версия должна быть переписана на диск. Если нет, то она может быть просто уничтожена, то есть соответствующая физическая страница объявляется свободной.

На производительность системы со страничной организацией памяти влияют временные затраты, связанные с обработкой страничных прерываний и преобразованием виртуального адреса в физический. При часто возникающих страничных прерываниях система может тратить большую часть времени впустую, на свопинг страниц. Чтобы уменьшить частоту страничных прерываний, следовало бы увеличивать размер страницы. Кроме того, увеличение размера страницы уменьшает размер таблицы страниц, а значит уменьшает затраты памяти.

Страничное распределение памяти может быть реализовано в упрощенном варианте, без выгрузки страниц на диск. В этом случае все виртуальные страницы всех процессов постоянно находятся в оперативной памяти. В этом случае фрагментация почти не происходит.

##### **Сегментное распределение**

Виртуальное адресное пространство процесса делится на сегменты, размер которых определяется программистом с учетом смыслового значения содержащейся в них информации. Отдельный сегмент может представлять собой подпрограмму, массив данных и т.п. Иногда сегментация программы выполняется по умолчанию компилятором.

При загрузке процесса часть сегментов помещается в оперативную память, а часть сегментов размещается в дисковой памяти. Во время загрузки система создает таблицу сегментов процесса (аналогичную таблице страниц), в которой для каждого сегмента указывается начальный физический адрес сегмента в оперативной памяти, размер сегмента, правила доступа, признак модификации, признак обращения к данному сегменту за последний интервал времени и некоторая другая информация. Если виртуальные адресные пространства нескольких процессов включают один и тот же сегмент, то в таблицах сегментов этих процессов делаются ссылки на один и тот же участок оперативной памяти, в который данный сегмент загружается в единственном экземпляре.

Система с сегментной организацией функционирует аналогично системе со страничной организацией: время от времени происходят прерывания, связанные с отсутствием нужных сегментов в памяти, при необходимости освобождения памяти некоторые сегменты выгружаются, при каждом обращении к оперативной памяти выполняется преобразование виртуального адреса в физический. Кроме того, при обращении к памяти проверяется, разрешен ли доступ требуемого типа к данному сегменту.

Недостатком данного метода распределения памяти является фрагментация на уровне сегментов и более медленное по сравнению со страничной организацией преобразование адреса.

##### **Странично-сегментное распределение**

Виртуальное пространство процесса делится на сегменты, а каждый сегмент в свою очередь делится на виртуальные страницы, которые нумеруются в пределах сегмента. Оперативная память делится на физические страницы. Загрузка процесса выполняется операционной системой постранично, при этом часть страниц размещается в оперативной памяти, а часть на диске. Для каждого сегмента создается своя таблица страниц, структура которой полностью совпадает со структурой таблицы страниц, используемой при страничном распределении. Для каждого процесса создается таблица сегментов, в которой указываются адреса таблиц страниц для всех сегментов данного процесса. Адрес таблицы сегментов загружается в специальный регистр процессора, когда активизируется соответствующий процесс.