

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ

Ярославский государственный университет им. П.Г. Демидова

Кафедра вычислительных и программных систем

УТВЕРЖДАЮ

Декан факультета ИВТ

 Д.Ю. Чалый

«18» мая 2020 г.

Рабочая программа дисциплины

«Операционные системы семейства UNIX и их администрирование»

Направление подготовки

01.03.02 Прикладная математика и информатика

Направленность (профиль)

«Прикладная математика и информатика»

Квалификация выпускника

Бакалавр

Форма обучения

очная

Программа рассмотрена
на заседании кафедры
от 23 апреля 2020 г., протокол № 8

Программа одобрена НМК
факультета ИВТ
протокол № 7 от 17 мая 2020 г.

Ярославль

1. Цели освоения дисциплины

Целями дисциплины «Операционные системы семейства UNIX и их администрирование» являются изучение принципов устройства POSIX-совместимых операционных систем, приёмов и методики их администрирования. Поскольку операционные системы семейства UNIX в существенной степени разрабатывались в университетской среде и в чистом виде воплотили в себе многие основополагающие концепции построения компонентов операционных систем, то данный курс способствует фундаментализации образования.

2. Место дисциплины в структуре ОП бакалавриата

Дисциплина «Операционные системы семейства UNIX и их администрирование» относится к вариативной части (дисциплина по выбору) ОП бакалавриата.

Она базируется на знаниях и навыках, полученных студентами при изучении общепрофессиональных дисциплин компьютерного цикла, в наибольшей степени дисциплины «Основы программирования», «Операционные системы».

Помимо расширения общепрофессиональной составляющей образования студентов дисциплина направлена на их подготовку к профессиональной деятельности в области системного администрирования и системной интеграции.

3. Планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения ОП бакалавриата

Процесс изучения дисциплины направлен на формирование следующих элементов компетенций в соответствии с ФГОС ВО, ОП ВО и приобретения следующих знаний, умений, навыков и (или) опыта деятельности:

Формируемая компетенция (код и формулировка)	Индикатор достижения компетенции (код и формулировка)	Перечень планируемых результатов обучения
Профессиональные компетенции		
ПК – 3 Способен к разработке и проектированию программного обеспечения, к использованию современных технологий программирования	ПК – 3.2 Умеет работать с современными средствами разработки программного обеспечения (ПО), имеет навыки разработки ПО с использованием современных инструментальных средств.	Знать • принципы организации и операционных систем семейства UNIX; • устройство основных компонентов операционных систем семейства UNIX; • основные команды операционных систем семейства UNIX. Уметь • выполнять основные операции с элементами файловой системы; • выполнять типовые операции по обслуживанию операционной системы UNIX; • диагностировать и устранять неполадки операционной системы UNIX; Владеть • навыками разработки средств автоматизации (скриптов), предназначенных для решения

		задач администрирования UNIX; навыками планирования конфигурации и профилей настройки операционной системы UNIX.
--	--	---

4. Объем, структура и содержание дисциплины

Общая трудоемкость дисциплины составляет 5 зач. ед., 180 акад. час.

№ п/п	Темы (разделы) дисциплины, их содержание	Семестр	Виды учебных занятий, включая самостоятельную работу студентов, и их трудоемкость (в академических часах)						Формы текущего контроля успеваемости Форма промежуточной аттестации (по семестрам)	
			Контактная работа							
			лекции	практические	лабораторные	консультации	аттестационные испытания	самостоятельная работа		
1.	История, базовые понятия и механизмы ОС UNIX	5	2		2			2		
2.	Базовые команды UNIX	5	2		2			2	лабораторная работа	
3.	Устройство файловой системы и команды для работы с файлами	5	2		2	1		4		
4.	Процессы в UNIX. Управление процессами	5	2		2			4		
5.	Права доступа процессов к файлам и каталогам	5	2		2	1		4		
6.	Командный интерпретатор bash. Разработка shell- скриптов	5	2		2			4	лабораторная работа	
7.	Текстовые редакторы vim и emacs	5	2		2	1		4		
8.	Обработка текстовых данных в UNIX	5	2		2	1		4		

9.	Процесс начальной загрузки системы (на примере ОС GNU/Linux)	5	2		2			3,7	Тест
	Всего за 5 семестр		18		18	4		31,7	Зачет
1.	Служба запуска заданий по расписанию cron	6			4	1		3	
2.	Типы файловых систем в UNIX. Разметка жёсткого диска	6			4			3	
3.	Установка приложений из пакетов (в ОС GNU/Linux) и исходных текстов	6			4	1		3	лабораторная работа
4.	Настройка и диагностика сетевых подключений	6			4			3	
5.	Удалённый доступ к командной оболочке через SSH	6			4	1		3	лабораторная работа
6.	Web-сервер и его конфигурирование	6			4			3	лабораторная работа
7.	Сетевые файловые системы NFS и CIFS и их конфигурирование.	6			4	1		3	
8.	Служба аутентификации PAM и конфигурирование её модулей	6			4	1		3	
9.	Оконная система X Window System	6			4			7	
	Всего за 6 семестр				36	5		31	Экзамен
	Всего		18		54	9		62,7	

Содержание разделов дисциплины:

1. История, базовые понятия и механизмы ОС UNIX.
2. Базовые команды UNIX
3. Устройство файловой системы и команды для работы с файлами. Разметка жёсткого диска, таблица разделов, расширенные и дополнительные разделы. Понятие виртуальной файловой системы и точек монтирования. Виды файловых систем (общего назначения: ext4, btrfs, zfs, reiserfs). Создание файловых систем и инструменты их создания. Дефрагментация файловых систем.
4. Процессы в UNIX. Управление процессами
5. Права доступа процессов к файлам и каталогам
6. Командный интерпретатор bash. Разработка shell-скриптов
7. Текстовые редакторы vim и emacs
8. Обработка текстовых данных в UNIX
9. Процесс начальной загрузки системы (на примере ОС GNU/Linux)
10. Служба запуска заданий по расписанию cron
11. Типы файловых систем в UNIX. Разметка жёсткого диска
12. Установка приложений из пакетов (в ОС GNU/Linux) и исходных текстов. Понятие пакета и его зависимостей, скриптов подготовки, настройки, удаления. Низкоуровневые средства установки и манипулирования пакетам: и dpkg и rpm. Понятие репозитория пакетов. Структура репозитория Debian. Менеджеры пакетов apt, urpm, yum, aptitude. Установка приложений из исходных кодов. Системы сборки make, cmake. Установка пакетов в языках python.
13. Настройка и диагностика сетевых подключений. Структура сети интернет: адреса, сети, маршрутизация пакетов. Отображение конфигурации сетевых интерфейсов.

Команды `ipconfig`, `ip`, `tracert`, `ping`. Маршрутизация пакетов в IPv4 (v6?) сетях. Настройка маршрутизации. Команды `ip`, `route`. Понятие прокси-сервера. Настройка доступа через прокси-сервер. Клиентские приложения. `wget`, `curl`. Веб браузеры `lynx`, `w3m`.

14. Удалённый доступ к командной оболочке через SSH. Протокол взаимодействия `ssh`. Настройка клиента и сервера Open SSH. Способы аутентификации, поддерживаемые Open SSH, генерация и использование пары из открытого и секретного ключа.
15. Web-сервер и его конфигурирование. Конфигурационный файл (или файлы в Debian подобных дистрибутивах). Управление доступом. Модули Apache и их настройка. Журналы сервера. Динамическое содержимое: CGI-bin, FastCGI, специализированные модули. Виртуальные хосты, их настройка и условия использования.
16. Сетевые файловые системы NFS и CIFS и их конфигурирование.
17. Служба аутентификации PAM и конфигурирование её модулей. Расположение и назначением модулей PAM. Клиенты PAM и настройка сценариев аутентификации для них.
18. Оконная система X Window System. Понятие X-сервера и X-клиента. Стартовые скрипты X-сервера `xinit` и `startx`. Конфигурационный файл `xorg.conf`. Переменная окружения `DISPLAY` и запуск X-клиентов. Эмуляторы терминала, менеджеры окон и среды рабочего стола. Виртуальный X-сервер.

5. Образовательные технологии, используемые при осуществлении образовательного процесса по дисциплине

В процессе обучения используются следующие образовательные технологии:

Вводная лекция – дает первое целостное представление о дисциплине и ориентирует студента в системе изучения данной дисциплины. Студенты знакомятся с назначением и задачами курса, его ролью и местом в системе учебных дисциплин и в системе подготовки в целом. Дается краткий обзор курса, история развития науки и практики, достижения в этой сфере, имена известных ученых, излагаются перспективные направления исследований. На этой лекции высказываются методические и организационные особенности работы в рамках данной дисциплины, а также дается анализ рекомендуемой учебно-методической литературы.

Академическая лекция (или лекция общего курса) – последовательное изложение материала, осуществляемое преимущественно в виде монолога преподавателя.

Требования к академической лекции: современный научный уровень и насыщенная информативность, убедительная аргументация, доступная и понятная речь, четкая структура и логика, наличие ярких примеров, научных доказательств, обоснований, фактов.

Лекция-беседа или «диалог с аудиторией», является наиболее распространенной и сравнительно простой формой активного вовлечения студентов в учебный процесс. Эта лекция предполагает непосредственный контакт преподавателя с аудиторией.

Преимущество лекции-беседы состоит в том, что она позволяет привлекать внимание студентов к наиболее важным вопросам темы, определять содержание и темп изложения учебного материала с учетом особенностей студентов.

Мастер-класс – это особая форма учебного занятия, когда преподаватель-мастер передает свой опыт путем прямого и комментированного показа последовательности действий, методов, приемов и форм педагогической деятельности. Целью проведения мастер-класса является профессиональное, интеллектуальное и эстетическое воспитание студентов, и прежде всего, развитие в ходе мастер-класса способности студента самостоятельно и нестандартно мыслить.

Лабораторная работа – организация учебной работы с реальными материальными и информационными объектами, экспериментальная работа с аналоговыми моделями

реальных объектов.

6. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень лицензионного программного обеспечения и информационных справочных систем (при необходимости)

В процессе осуществления образовательного процесса используются: для разработки документов, презентаций, для работы с электронными таблицами

OfficeStd 2013 RUS OLP NL Acdmc 021-10232

LibreOffice (свободное)

издательская система LaTeX;

– OS Linux (свободная)

– для поиска учебной литературы библиотеки ЯрГУ – Автоматизированная библиотечная информационная система "БУКИ-NEXT" (АБИС "Буки-Next").

7. Перечень основной и дополнительной учебной литературы, ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

а) основная:

1. Робаческий, А. М., Операционная система UNIX / А. М. Робачевский, С. А. Немнюгин, О. Л. Стесник. - 2-е изд., перераб. и доп., СПб., БХВ-Петербург, 2014, 635с

б) дополнительная:

1. Костромин, В. А., Самоучитель Linux для пользователя, СПб., БХВ-Петербург, 2004, 672с

Курячий Г. В. Операционная система Linux: курс лекций : учеб. пособие для вузов. / Г. В. Курячий, К. А. Маслинский - М.: Интернет-Ун-т Информационных Технологий, 2005. - 387 с.

в) ресурсы сети «Интернет»

- Учебные материалы по UNIX <http://www.ee.surrey.ac.uk/Teaching/Unix/unixintro.html>

Доступ свободный

Электронно-библиотечная система «Юрайт»(<https://urait.ru/>).

Электронно-библиотечная система «Лань»(<https://e.lanbook.com/>).

8. Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине

Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине включает в свой состав специальные помещения:

-учебные аудитории для проведения занятий лекционного типа и практических занятий (семинаров);

-учебные аудитории для проведения лабораторных занятий;

- учебные аудитории для проведения групповых и индивидуальных консультаций,

- учебные аудитории для проведения текущего контроля и промежуточной аттестации;

-помещения для самостоятельной работы;

-помещения для хранения и профилактического обслуживания технических средств обучения.

Специальные помещения укомплектованы средствами обучения, служащими для представления учебной информации большой аудитории.

Для проведения занятий лекционного типа предлагаются наборы демонстрационного оборудования и учебно-наглядных пособий, хранящиеся на электронных носителях и обеспечивающие тематические иллюстрации, соответствующие рабочим программам дисциплин.

Помещения для лабораторных занятий и самостоятельной работы обучающихся оснащены компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду организации.

Число посадочных мест в лекционной аудитории больше либо равно списочному составу потока, а в аудитории для практических занятий (семинаров) – списочному составу группы обучающихся.

Автор(ы) :

Старший преподаватель кафедры ВПС А.М.Васильев

**Приложение №1 к рабочей программе дисциплины
«Операционные системы семейства UNIX и их администрирование»
Фонд оценочных средств
для проведения текущей и промежуточной аттестации студентов
по дисциплине**

**1. Типовые контрольные задания или иные материалы, необходимые для оценки
знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы
формирования компетенций**

**1.1. Контрольные задания и иные материалы, используемые в процессе текущей
аттестации**

Примеры заданий для лабораторных работ

Лабораторная работа по теме «Базовые команды UNIX»

Целью данного задания является ознакомление с системными вызовами UNIX. В качестве практики предлагается реализовать простую командную оболочку. Результат своей работы (исходный код командной оболочки) будет необходимо загрузить в качестве ответа.

Скачайте скелет командной оболочки, прикрепленный к этому практическому заданию, и просмотрите его код. Исходный код можно разделить на две части: обработку строки команды и её выполнение. Обработкой занимается часть приложения, называемая парсером.

Парсер поддерживает разбор лишь для базовых команд. Пример набора команд приведён ниже. Скопируйте эти команды в файл с названием test.sh.:

```
ls > y
cat < y | sort | uniq | wc > y1
cat y1
rm y1
ls | sort | uniq | wc
rm y
```

Вы можете скомпилировать скелет командной оболочки с помощью компилятора gcc.:

```
$ gcc sh.c
```

В результате будет создан исполняемый файл с названием a.out, который следует запустить следующим образом.:

```
$ ./a.out < test.sh
```

Во время выполнения вы увидите множество сообщений. Эти сообщения говорят о невозможности выполнить действия из скрипта.

Вашей задачей является модификация скелета командной оболочки таким образом, чтобы приложение могло успешно выполнять тестовый скрипт.

Выполнение простых команд

Начните реализацию данного приложения с поддержки выполнения простых команд, таких как.:

```
$ ls
```

То есть ваша командная оболочка должна научиться запускать простые приложения.

Парсер уже может разбирать данную ситуацию. В результате разбора создаётся структура типа `exescmd`, которая передаётся функции `runcmd`.

Для решения этой задачи достаточно написать код обработчика ситуации ' ' внутри оператора case. Вам потребуется использовать системный вызов `exes`, поэтому прочитайте официальную документацию, доступную в системе руководств.:

```
$ man 3 exes
```

Для проверки реализации в тестовом файле достаточно написать одну команду `ls`.

Перенаправление ввода-вывода

Далее реализуйте перенаправление вывода в файл и перенаправление ввода из файла. Данная операция реализуется в командной оболочке с помощью операторов `>` и `<`. После реализации этого функционала ваше приложение сможет выполнять следующий код::

```
echo "6.828 is cool" > x.txt
```

```
cat < x.txt
```

Парсер умеет разбирать команды перенаправления вывода в файл (`>`) и ввода из файла (`<`). Если команда содержала перенаправление, то создаётся структура типа `redircmd`. Как и в предыдущем случае достаточно реализовать код в соответствующей ветке оператора `case`. Для реализации этого функционала вам потребуются системные вызовы `open` и `close`. Возможно, придётся посмотреть данную информацию в сети Интернет.

Для проверки используйте код примера, приведённого выше.

Реализация конвейеров

Далее реализуйте перенаправление ввода-вывода с использованием конвейеров (`pipe`), чтобы можно было выполнить следующий код::

```
$ ls | sort | uniq | wc
```

Парсер распознаёт команды по созданию конвейера. В результате разбора создаётся структура `pipescmd`. Для поддержки достаточно реализовать код в последней ветке оператора `case`: `|`. Для реализации этой функции вам могут потребоваться системные вызовы `pipe`, `fork`, `close`, `dup`.

Для проверки используйте код примера, приведённого выше.

Последняя проверка

Убедитесь, что ваша командная оболочка способна выполнять код примера, приведённого в начале задания.

После этого можете загружать решение задачи на сайт.

Лабораторная работа по теме «Командный интерпретатор `bash`. Разработка `shell-скриптов`»

Ваша задача состоит в том, чтобы выводить строку с описанием системного вызова каждый раз, когда ядро обрабатывает системный вызов. Достаточно вывести лишь имя системного вызова и возвращаемое значение. Не надо выводить значения аргументов системного вызова.

В рамках данной задачи вам предлагается ознакомиться с обработкой системных вызовов внутри ядра операционной системы.

После выполнения этой задачи, при загрузке операционной системы вы увидите строки, наподобие следующих:

```
fork -> 2
```

```
exec -> 0
```

```
open -> 3
```

```
close -> 0
```

```
$write -> 1
```

```
write -> 1
```

Данная последовательность системных вызовов описывает создание нового процесса системой инициализации `init`, и последующего запуска приложения `sh`. Данный процесс открывает два файловых дескриптора, а после `sh` записывает символы `$` и пробел.

Для решения этой задачи вам потребуется изменить код функции `syscall()`, находящийся в файле `syscall.c`.

Дополнительное задание: разберитесь с системой получения аргументов системных вызовов и выведите их вместе с именем системного вызова.

Реализация нового системного вызова

В рамках данного задания предлагается реализовать системный вызов `halt`, позволяющий завершить работу операционной системы. Цель данного задания - изучить различные особенности организации системных вызовов.

Реализация системного вызова будет завершать работу виртуальной машины QEMU. Ниже приведён кусок исходного кода, который позволяет это сделать::

```
char *p = "Shutdown";
for( ; *p; p++)
    outb(0x8900, *p);
```

Данный код должен выполняться на уровне ядра операционной системы, а не в рамках прикладного программного обеспечения.

Помимо реализации системного вызова, вам потребуется реализовать программу прикладного уровня, которая будет делать системный вызов. Добавьте следующий код в файл `halt.c` ::

```
#include "types.h"
#include "stat.h"
#include "user.h"
```

```
int main(int argc, char *argv[])
{
    halt();
    return 0;
}
```

Для того, чтобы это приложение можно было вызывать из операционной системы xv6, добавьте `_halt` к списку `UPROGS`, определённого в файле `Makefile`.

Предлагаемый подход

В качестве основы для создания нового системного вызова вам предлагается изучить и скопировать структуру существующего системного вызова. Наиболее подходящим является системный вызов, который не принимает никаких аргументов, например `uptime`. Сначала вам необходимо выяснить: в каких файлах находится реализация поддержки системных вызовов. Для этого используйте приложение `grep` ::

```
$ grep -n uptime *.c
```

После того, как закончите реализацию, вызов `halt` внутри операционной системы xv6 приведёт к завершению эмулятора.

Дополнительное задание: Реализуйте системный вызов `dup2()`.

Лабораторная работа по теме «Установка приложений из пакетов (в ОС GNU/Linux) и исходных текстов»

Используя систему управления пакетами `apt` и приложения `apt-get`, `apt-update`, `apt` выполните следующие задачи:

- Установите текстовый редактор `emacs`.
- Установите пакет, содержащий приложение `docker`.
- Начните установку пакета `texlive`, но не устанавливайте его. Оцените следующие параметры при установке:
 - Объём скачиваемых данных.
 - Количество устанавливаемых пакетов.
 - Объём данных после установки.

Используя систему поиска пакетов `apt-cache` и информационный сайт о пакетах GNU/Debian выполните следующие задачи.

- Найдите имена как минимум 4 консольных веб-браузеров.
- Установите один из них, который ещё не установлен в систему.
- Зайдите на информационный сайт и выполните поиск браузеров на нём.

- Посмотрите статистику по локальным пакетам.
Давайте поставим самую последнюю версию приложения для скачивания видео-контента из сети интернет путём установки пакета из репозитория. Приложение называется youtube-dl.
Установите приложение для скачивания видео-контента из сети интернет путём установки пакета из репозитория.
- Посмотрите информацию о пакете youtube-dl. => 17.05.18.1.4
- Откройте ресурс <http://deb-multimedia.org>
- Добавьте репозиторий deb-multimedia в список репозитория для вашей системы.
- Обновите список доступных вам пакетов.
 - Для установки ключей подписей пакетов обычно используются 2 подхода: 1. apt-key 2. установка пакетов, содержащих необходимые подписи.
 - В deb-multimedia используется deb-multimedia-keyring для предоставления ключей.
- Поставьте пакет deb-multimedia-keyring.
- Обновите список доступных вам пакетов. Ошибок с ключом быть не должно.
- Посмотрите информацию о пакете youtube-dl.

Лабораторная работа по теме «Удалённый доступ к командной оболочке через SSH»
Настройте доступ к удалённому компьютеру с использованием авторизации по приватному ключу.

- Создайте приватный ключ пользователя с использованием утилиты ssh-keygen.
- Перенесите публичную часть ключа пользователя на удалённый компьютер.
Публичную часть ключа необходимо поместить в файл ~/.ssh/authorized_keys. В данном файле содержится список ключей, которые могут быть использованы для внешнего подключения в данную учётную запись.
- Используем приложение ssh-copy-id. Данное приложение предназначено для решения поставленной задачи
- Используем приложение scp. Данное приложение позволяет копировать данные между текущей машиной и удалённым компьютером. Логика его работы в целом совпадает с логикой работы cp.

Лабораторная работа по теме «Web-сервер и его конфигурирование»

Создайте свою собственную конфигурацию веб-сервера Nginx, которая способна показывать информацию из отдельного каталога. Используйте следующую инструкцию для достижения результата.

- Создать свой собственный конфигурационный файл сайта на основании default. 1.1. Перейти в каталог /etc/nginx/sites-available 1.2. Скопировать файл default в my-cool-site-config. Очевидно, что файл конфигурации my-cool-site-config должен располагаться в конфигурационном каталоге nginx: /etc/nginx/sites-available 1.3. Открыть файл my-cool-site-config в текстовом редакторе и изменить в нём путь к статическим файлам приложения ruby-app 1.4. Т.е. нужно в root указать путь к каталогу public.
- Создать ссылку на файл конфигурации /etc/nginx/sites-available/my-cool-site-config в каталоге /etc/nginx/sites-enabled: `cd /etc/nginx/sites-enabled sudo ln -sf /etc/nginx/sites-available/my-cool-site-config`
- Убрать ссылку на конфигурацию по умолчанию из /etc/nginx/sites-enabled: `cd /etc/nginx/sites-enabled sudo unlink default`
- Перезагрузить конфигурацию nginx: `sudo systemctl reload nginx`
- Проверить, что nginx выдаёт статические файлы ruby-app, открыв в браузере <http://localhost>

Критерии оценивания лабораторных работ

Оценка	Критерии
Отлично Уровень формирования компетенций: высокий	<p>ОПК-3: Задание выполнено полностью, включая дополнительные пункты, если они сформулированы. Студент знает подходящие для решения средства и инструменты. Понимает использованные команды, свободно дает пояснения к выполненным действиям.</p> <p>ПК-7: Задание выполнено полностью, включая дополнительные пункты, если они сформулированы. Студент применяет подходящие для решения средства и инструменты. Свободно может изменить конфигурацию команд или выбрать и применить средства при изменении деталей задания.</p> <p>ПК-5: При решении задач студент свободно использует техническую литературу, осуществляет поиск необходимой информации в сети Интернет, анализирует найденные решения.</p>
Хорошо Уровень формирования компетенций: продвинутый	<p>ОПК-3: Задание выполнено полностью, но без дополнительных пунктов, если они сформулированы или с небольшими неточностями в одном – двух пунктах задания. Студент знает подходящие для решения средства и инструменты. Понимает использованные команды, дает пояснения к большинству выполненных действий.</p> <p>ПК-7: Задание выполнено полностью, но без дополнительных пунктов, если они сформулированы или с небольшими неточностями в одном – двух пунктах задания. Студент применяет подходящие для решения средства и инструменты. Может изменить конфигурацию команд или выбрать и применить средства при изменении деталей задания, но не во всех случаях.</p> <p>ПК-5: При решении задач студент использует знакомую техническую литературу, осуществляет поиск необходимой информации в сети Интернет.</p>
Удовлетворительно Уровень формирования компетенций: пороговый	<p>ОПК-3: Задание выполнено не полностью, или с ошибками, но не более чем в половине задания. Студент знает основные подходящие для решения средства и инструменты, но не для всех сформулированных пунктов задания. Понимает использованные команды, но с трудом дает пояснения к большинству выполненных действий.</p> <p>ПК-7: Задание выполнено не полностью, или с ошибками, но не более чем в половине задания. Студент применяет подходящие для решения средства и инструменты, но не для всех сформулированных пунктов задания. Не может изменить конфигурацию команд или выбрать и применить средства при изменении деталей задания, или выполняет изменения только в самых простых случаях.</p> <p>ПК-5: При решении задач студент затрудняется в поиске решений с помощью технической литературы и сети Интернет кроме простейших случаев.</p>
Неудовлетворительно	ОПК-3: Задание не выполнено, или выполнено с грубыми ошибками, в большей части задания. Студент не знает подходящие для решения средства и инструменты. Не понимает

	<p>использованные команды, не дает пояснения к большинству выполненных действий.</p> <p>ПК-7: Задание не выполнено, или выполнено с грубыми ошибками, в большей части задания. Студент не может применить подходящие для решения средства и инструменты. Не может изменить конфигурацию команд или выбрать и применить средства при изменении деталей задания.</p> <p>ПК-5: Студент не использует техническую литературу, не может сформулировать запрос при поиске необходимой информации в сети Интернет</p>
--	--

Примеры вопросов для тестового задания

Вопрос №1. Находясь в каталоге /var/log/messages Василий выполнил команду `ls -l el` и увидел следующий вывод:

```
lrwxrwxrwx 1 ad ad 20 дек 21 17:45 el -> ../lib/vagrant
```

Затем выполнил команду `ls -ld ../lib/vagrant` и увидел следующий вывод:

```
drwxr-xr-x 2 ad ad 4096 дек 3 15:38 ../vagrant
```

- 1.1. Чем является файл `vagrant`?
- 1.2. Как вы это определили?
- 1.3. Укажите полный путь к данному файлу.
- 1.4. Укажите полный путь к дополнительному файлу.

Ответ на вопрос №1.

- 1.1. Файл `vagrant` является директорией.
- 1.2. В выводе второй команды первая буква `d`, что обозначает, что файл является директорией.
- 1.3. Полный путь к каталогу `vagrant`: `/var/log/lib/vagrant`.
- 1.4. Полный путь к символической ссылке `el`: `/var/log/messages/el`.

Вопрос №2. Вы находитесь в каталоге `/tmp`. В нём существует каталог `a`, в котором есть пустой файл `b`.

- 2.1. Создайте каталог `/tmp/c`, в котором будет пустой каталог `d`.
- 2.2. Сделайте задание пункта 2.1 другим способом
- 2.3. Создайте символическую ссылку `/tmp/d`, ссылающуюся на пустой файл `b`.

Ответ на вопрос №2.

- 2.1.
\$ `mkdir c`
\$ `mkdir c/d`
- 2.2.
\$ `mkdir -p c/d`
- 2.3.
ln -s `/tmp/d b`

Вопрос №3. Василий выполнил команду `ls -ld /usr/local/lib` и увидел следующий вывод:

```
drwxrwsr-x 5 root staff 57 сен 19 14:29 /usr/local/lib
```

- 3.1. Что означает буква `s` в разрешениях для группы?
- 3.2. Напишите все факты о данном файле, которые можете извлечь из данного вывода.

Ответ на вопрос №3.

- 3.1. Буква `s` в разрешениях файла говорит о том, что при создании файлов или каталогов внутри данного каталога они будут принадлежать группе `staff` вне зависимости от того какими правами обладает пользователь, создающий данные каталоги.
- 3.2.

- Данный файл является директорией, т.к. первая буква в разрешениях - `d`.

- Файл принадлежит пользователю root и группе staff.
- Владелец может выполнять все функции с данным каталогом: добавлять, удалять, переименовывать файлы и запускать в рамках каталога любые приложения.
- Такими же правами обладает и группа staff.
- Все остальные пользователи могут лишь просматривать содержимое данного каталога и запускать в нём приложения.
- Данный каталог был создан 15 сентября в 14:29.

Показатели и критерии, используемые при выставлении оценки:

Вопрос теста	Критерий формирования компетенции	Баллы
1 вопр. 1 пункт	ОПК-3: верный полный ответ	1
1 вопр. 2 пункт	ОПК-3: верный полный ответ	1
1 вопр. 3 пункт	ПК-7: верный полный ответ	1
1 вопр. 4 пункт	ПК-7: верный полный ответ	1
2 вопр. 1 пункт	ПК-7: верный полный ответ	1
2 вопр. 2 пункт	ПК-5: верный полный ответ	1
2 вопр. 3 пункт	ПК-7: верный полный ответ	1
3 вопр. 1 пункт	ПК-5: верный полный ответ	1
3 вопр. 2 пункт	ОПК-3: верный полный ответ	1
3 вопр. 2 пункт	ПК-7: верный полный ответ	1

Максимальное количество баллов по ОПК-3 – 3 балла

Максимальное количество баллов по ПК-5 – 2 балла

Максимальное количество баллов по ПК-7 – 5 баллов.

Набранное количество баллов соответствует оценке за выполнение экзаменационной работы:

- 0 баллов по ОПК-3 или менее 2 баллов по ПК-7 или 0 баллов по ПК-5 — оценка «неудовлетворительно»,

- не менее 1 балла по ОПК-3 и не менее 2 баллов по ПК-7 и не менее 1 балла по ПК-5, в общей сумме от 4 до 6 баллов — оценка «удовлетворительно», пороговый уровень формирования компетенции,

- не менее 3 баллов по ОПК-3 и не менее 3 баллов по ПК-7 и не менее 1 балла по ПК-5, в общей сумме от 7 до 8 баллов — оценка «хорошо», продвинутый уровень формирования компетенции,

- не менее 4 баллов по ОПК-3 и не менее 4 баллов по ПК-7 и не менее 2 баллов по ПК-5, в общей сумме от 9 до 10 баллов — оценка «отлично», высокий уровень формирования компетенции.

Список вопросов к зачету:

1. Файловая подсистема
2. Разметка жёсткого диска, таблица разделов, расширенные и дополнительные разделы.
3. Понятие виртуальной файловой системы и точек монтирования.
4. Проверка доступного и занятого дискового пространства, приложения du, di, df, ls, braserо.
5. Использование программы fdisk.
6. Виды файловых систем (общего назначения: ext4, brtfs, zfs, reiserfs).

7. Создание файловых систем и инструменты их создания.
8. Проверка файловых систем с помощью fsck и badblocks.
9. Дефрагментация файловых систем. Инструмент e4defrag, shake, defrag.
10. Процессы в UNIX. Управление процессами
11. Права доступа процессов к файлам и каталогам
12. Командный интерпретатор bash. Разработка shell-скриптов
13. Текстовые редакторы vim и emacs
14. Обработка текстовых данных в UNIX
15. Процесс начальной загрузки системы (на примере ОС GNU/Linux)

Методические указания по выставлению зачета

Зачет выставляется по результатам выполнения всех лабораторных работ и теста на оценку не ниже удовлетворительно. Работы выполняются и сдаются в течение семестра последовательно в процессе освоения материала или в исключительных случаях на зачете.

Список вопросов к экзамену:

1. Файловая подсистема
 - Разметка жёсткого диска, таблица разделов, расширенные и дополнительные разделы.
 - Понятие виртуальной файловой системы и точек монтирования.
 - Проверка доступного и занятого дискового пространства, приложения du, di, df, ls, braserо.
 - Использование программы fdisk.
 - Виды файловых систем (общего назначения: ext4, brtfs, zfs, reiserfs).
 - Создание файловых систем и инструменты их создания.
 - Проверка файловых систем с помощью fsck и badblocks.
 - Дефрагментация файловых систем. Инструмент e4defrag, shake, defrag.
2. Процессы в UNIX. Управление процессами
3. Права доступа процессов к файлам и каталогам
4. Командный интерпретатор bash. Разработка shell-скриптов
5. Текстовые редакторы vim и emacs
6. Обработка текстовых данных в UNIX
7. Процесс начальной загрузки системы (на примере ОС GNU/Linux)
8. Служба запуска заданий по расписанию cron
9. Типы файловых систем в UNIX. Разметка жёсткого диска
10. Установка приложений
 - Понятие пакета и его зависимостей, скриптов подготовки, настройки, удаления и так далее.
 - Низкоуровневые средства установки и манипулирования пакетам:и dpkg и rpm.
 - Понятие репозитория пакетов. Структура репозитория Debian.
 - Менеджеры пакетов apt, urpm, yum, aptitude.
 - Установка приложений из исходных кодов. Системы сборки make, cmake. Установка пакетов в языках python и т.д.
11. Настройка и диагностика сети
 - Структура сети интернет: адреса, сети, маршрутизация пакетов.
 - Отображение конфигурации сетевых интерфейсов. Команды ipconfig, ip, traceroute, ping.
 - Маршрутизация пакетов в IPv4 (v6?) сетях.
 - Настройка маршрутизации. Команды ip, route.
 - Понятие прокси-сервера. Настройка доступа через прокси-сервер.

- Клиентские приложения. wget, curl. Веб браузеры lynx, w3m.
- 12. Удалённый доступ к командной оболочке.
- Протокол взаимодействия ssh.
- Настройка клиента и сервера Open SSH.
- Способы аутентификации, поддерживаемые Open SSH, генерация и использование пары из открытого и секретного ключа.
- Интересные особенности использования протокола ssh.
 - Туннелирование TCP-соединений
 - Создание SOCKS-прокси
- 13. Web-сервер Apache/nginx
- Конфигурационный файл (или файлы в Debian подобных дистрибутивах).
- Управление доступом.
- Модули Apache и их настройка.
- Журналы сервера.
- Динамическое содержимое: CGI-bin, FastCGI, специализированные модули.
- Виртуальные хосты, их настройка и условия использования.
- 14. Взаимодействие с сетями Microsoft с помощью Samba
- FTP-подобный клиент smbclient.
- Подключение разделяемых файловых каталогов Linux с помощью команды mount.cifs.
- Настройка разделяемых каталогов Linux-машины.
- Аутентификация в сетях Microsoft через NTLM и Kerberos.
- 15. Служба аутентификации PAM и её модули.
- Расположение и назначением модулей PAM.
- Клиенты PAM и настройка сценариев аутентификации для них.
- 16. X Window System.
- Понятие X-сервера и X-клиента.
- Стартовые скрипты X-сервера xinit и startx.
- Конфигурационный файл xorg.conf.
- Переменная окружения DISPLAY и запуск X-клиентов.
- Эмуляторы терминала, менеджеры окон и среды рабочего стола.
- Виртуальный X-сервер.

Пример экзаменационной работы

1. Перечислите названия пакетов или приложений, которые потребуются для компиляции приложения, написанного на языках Си и C++, под GNU/Linux.

2. Оцените количество пакетов, объем скачиваемых данных и объём данных, необходимых для установки графического настольного окружения KDE, базовый пакет task-kde-desktop.

Какие шаги вы предприняли, чтобы получить эту информацию?

3. Определите к каким пакетам относятся следующие файлы:

```
/bin/bash -
/usr/lib/libsupp.a -
/usr/bin/filan -
```

Как вы это выяснили?

4. Установите приложение agedu. Архив с исходными кодами приложения можно скачать по адресу <https://www.chiark.greenend.org.uk/~sgtatham/agedu/agedu-20180522.5b12791.tar.gz>. Напишите все шаги, которые были выполнены для установки приложения.

Показатели и критерии, используемые при выставлении оценки:

Номер задачи	Критерии	Шкала оценивания
1	ПК-5: Уметь: <ul style="list-style-type: none"> использовать техническую документацию операционной системы UNIX. Владеть: <ul style="list-style-type: none"> навыками поиска информации об операционной системе UNIX. 	0 баллов – студент полностью не верно ответил на вопрос; 1 балл – студент частично ответил на вопрос без ошибок или дал почти полный ответ с небольшими неточностями 2 балла – студент полностью ответил на вопрос
2	ОПК-3: Знать <ul style="list-style-type: none"> принципы организации и операционных систем ; устройство основных компонентов операционных систем. Уметь <ul style="list-style-type: none"> выполнять основные операции с элементами файловой системы; 	0 баллов – студент не знает средства, позволяющие решить задачу; 1 балл – студент в решил задачу полностью, но не поясняет шаги решения; 2 балла – студент полностью решил задачу и описал решение
3	ОПК-3: Уметь <ul style="list-style-type: none"> выполнять основные операции с элементами файловой системы; конфигурировать основные серверные службы; выполнять типовые операции по обслуживанию операционной системы; Владеть <ul style="list-style-type: none"> навыками разработки средств автоматизации (скриптов), предназначенных для решения типовых задач администрирования; 	0 баллов – студент не знает средства, позволяющие решить задачу; 1 балл – студент в решил задачу полностью, но не поясняет шаги решения; 2 балла – студент полностью решил задачу и описал решение
4	ПК-7: Уметь <ul style="list-style-type: none"> выполнять основные операции с элементами файловой системы; выполнять типовые операции по обслуживанию операционной системы UNIX; 	0 баллов – студент не может решить задачу; 1 балл – студент в решил задачу полностью, но не поясняет шаги решения; 2 балла – студент полностью решил задачу и описал решение

Максимальное количество баллов по ОПК-3 – 4 балла

Максимальное количество баллов по ПК-5 – 2 балла

Максимальное количество баллов по ПК-7 – 2 балла.

Набранное количество баллов соответствует оценке за выполнение экзаменационной работы:

- 0 баллов по ОПК-3 или 0 баллов по ПК-5 или 0 баллов по ПК-7 — оценка «неудовлетворительно»,
- не менее 1 балла по ОПК-3 и не менее 1 балла по ПК-7, и не менее 1 балла по ПК-5 в общей сумме от 3 до 4 баллов — оценка «удовлетворительно», пороговый уровень формирования компетенции,
- не менее 2 баллов по ОПК-3 и не менее 1 балла по ПК-5 и не менее 1 балла по ПК-7, в общей сумме от 5 до 6 баллов — оценка «хорошо», продвинутый уровень формирования компетенции,
- не менее 3 баллов по ОПК-3 и не менее 1 балла по ПК-5 и не менее 2 баллов по ПК-7, в общей сумме от 7 до 8 баллов — оценка «отлично», высокий уровень формирования компетенции.

Методические указания по выставлению оценки за экзамен

Оценка выставляется по результатам выполнения экзаменационной работы на оценку не ниже удовлетворительно при условии выполнения лабораторных работ в течении семестра на оценку не ниже удовлетворительно. Работы выполняются и сдаются в течение семестра последовательно в процессе освоения материала.

2. Перечень компетенций, этапы их формирования, описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкалы оценивания

2.1. Шкала оценивания сформированности компетенций и ее описание

Оценивание уровня сформированности компетенций в процессе освоения дисциплины осуществляется по следующей трехуровневой шкале:

Пороговый уровень - предполагает отражение тех ожидаемых результатов, которые определяют минимальный набор знаний и (или) умений и (или) навыков, полученных студентом в результате освоения дисциплины. Пороговый уровень является обязательным уровнем для студента к моменту завершения им освоения данной дисциплины.

Продвинутый уровень - предполагает способность студента использовать знания, умения, навыки и (или) опыт деятельности, полученные при освоении дисциплины, для решения профессиональных задач. Продвинутый уровень превосходит пороговый уровень по нескольким существенным признакам.

Высокий уровень - предполагает способность студента использовать потенциал интегрированных знаний, умений, навыков и (или) опыта деятельности, полученных при освоении дисциплины, для творческого решения профессиональных задач и самостоятельного поиска новых подходов в их решении путем комбинирования и использования известных способов решения применительно к конкретным условиям. Высокий уровень превосходит пороговый уровень по всем существенным признакам.

2.2. Перечень компетенций, этапы их формирования, описание показателей и критериев оценивания компетенций на различных этапах их формирования

Код компетенции	Форма контроля	Этапы формирования (№ темы (раздела))	Показатели оценивания	Шкала и критерии оценивания компетенций на различных этапах их формирования		
				Пороговый уровень	Продвинутый уровень	Высокий уровень
Общепрофессиональные компетенции						
ОПК-3	Лабораторная работа. Тест Зачет. Экзамен	1-9	Знать <ul style="list-style-type: none">• принципы организации и операционных систем ;• устройство основных компонентов операционных систем. Уметь <ul style="list-style-type: none">• выполнять основные операции с элементами файловой системы;• конфигурировать основные серверные службы;• выполнять типовые операции по обслуживанию операционной системы; Владеть <ul style="list-style-type: none">• навыками разработки средств автоматизации (скриптов), предназначенных для решения типовых	Студент должен знать принципы организации и операционных систем; устройство основных компонентов операционных систем; основные команды операционных систем. Уметь выполнять основные операции с элементами файловой системы; выполнять типовые операции по обслуживанию операционной системы; диагностировать неполадки операционной системы; Владеть навыками разработки средств автоматизации (скриптов), предназначенных для	Студент должен знать принципы организации и операционных систем ; устройство основных компонентов операционных систем; основные команды операционных систем. Уметь выполнять основные операции с элементами файловой системы; конфигурировать основные серверные службы; выполнять типовые операции по обслуживанию операционной системы; диагностировать неполадки операционной системы; Владеть навыками разработки средств автоматизации (скриптов), предназначенных для решения типовых задач администрирования;	Студент должен знать принципы организации и операционных систем; устройство основных компонентов операционных систем; основные команды операционных систем. Уметь выполнять основные операции с элементами файловой системы; конфигурировать основные серверные службы; выполнять типовые операции по обслуживанию операционной системы; диагностировать и устранять неполадки операционной системы; Владеть навыками разработки средств автоматизации (скриптов), предназначенных для решения задач администрирования различной сложности; навыками планирования конфигурации и профилей настройки

			задач администрирования; • навыками планирования конфигурации и профилей настройки операционной системы в соответствии с заданными требованиями к её функционированию.	решения простых задач администрирования; навыками планирования конфигурации и профилей настройки операционной системы	навыками планирования конфигурации и профилей настройки операционной системы в соответствии с заданными требованиями к её функционированию. Студент должен уметь использовать основы системного подхода, критерии эффективной организации вычислительного процесса для постановки и решения задач организации оптимального функционирования вычислительных систем; выбирать, обосновывая свой выбор, оптимальные алгоритмы управления ресурсами;	операционной системы в соответствии с заданными требованиями к её функционированию. Студент должен уметь использовать основы системного подхода, критерии эффективной организации вычислительного процесса для постановки и решения задач организации оптимального функционирования вычислительных систем; выбирать, обосновывая свой выбор, оптимальные алгоритмы управления ресурсами; сравнивать и оценивать различные методы, лежащие в основе планирования процессов; разрабатывать прикладные многопоточные приложения; пользоваться функциями ОС при оценке качества функционирования алгоритмов управления ресурсами вычислительной системы. Студент должен владеть навыками разработки прикладных многопоточных приложений; разработки алгоритмов планирования процессов; использования алгоритмов и механизмов синхронизации
Профессиональные компетенции						

ПК-7	Лабораторная работа. Тест Зачет. Экзамен	1-9	<p>Знать</p> <ul style="list-style-type: none"> • принципы организации и операционных систем семейства UNIX; • устройство основных компонентов операционных систем семейства UNIX; • основные команды операционных систем семейства UNIX. <p>Уметь</p> <ul style="list-style-type: none"> • выполнять основные операции с элементами файловой системы; • выполнять типовые операции по обслуживанию операционной системы UNIX; • диагностировать и устранять неполадки операционной системы UNIX; <p>Владеть</p> <ul style="list-style-type: none"> • навыками разработки средств автоматизации (скриптов), предназначенных для решения задач администрирования UNIX; • навыками планирования конфигурации и 	<p>Студент должен знать</p> <p>принципы организации и операционных систем семейства UNIX;</p> <p>устройство основных компонентов операционных систем семейства UNIX;</p> <p>основные команды операционных систем семейства UNIX.</p> <p>Уметь выполнять</p> <p>основные операции с элементами файловой системы; выполнять типовые операции по обслуживанию операционной системы;</p> <p>диагностировать неполадки операционной системы;</p> <p>Владеть навыками разработки средств автоматизации (скриптов), предназначенных для решения простых задач администрирования;</p> <p>навыками планирования конфигурации и профилей настройки операционной системы</p>	<p>Студент должен знать</p> <p>принципы организации и операционных систем семейства UNIX;</p> <p>устройство основных компонентов операционных систем семейства UNIX;</p> <p>основные команды операционных систем семейства UNIX.</p> <p>Уметь выполнять</p> <p>основные операции с элементами файловой системы; конфигурировать основные серверные службы; выполнять типовые операции по обслуживанию операционной системы;</p> <p>диагностировать неполадки операционной системы;</p> <p>Владеть навыками разработки средств автоматизации (скриптов), предназначенных для решения типовых задач администрирования;</p> <p>навыками планирования конфигурации и профилей настройки операционной системы в соответствии с заданными требованиями к её функционированию.</p>	<p>Студент должен знать принципы организации и операционных систем семейства UNIX;</p> <p>устройство основных компонентов операционных систем семейства UNIX;</p> <p>основные команды операционных систем семейства UNIX.</p> <p>Уметь выполнять основные операции с элементами файловой системы;</p> <p>конфигурировать основные серверные службы; выполнять типовые операции по обслуживанию операционной системы; диагностировать и устранять неполадки операционной системы;</p> <p>Владеть навыками разработки средств автоматизации (скриптов), предназначенных для решения задач администрирования различной сложности; навыками планирования конфигурации и профилей настройки операционной системы в соответствии с заданными требованиями к её функционированию.</p> <p>Студент должен уметь использовать основы системного подхода, критерии эффективной организации вычислительного процесса для постановки и</p>
------	---	-----	--	---	---	---

			профилей настройки операционной системы UNIX.		Студент должен уметь использовать основы системного подхода, критерии эффективной организации вычислительного процесса для постановки и решения задач организации оптимального функционирования вычислительных систем; выбирать, обосновывая свой выбор, оптимальные алгоритмы управления ресурсами;	решения задач организации оптимального функционирования вычислительных систем; выбирать, обосновывая свой выбор, оптимальные алгоритмы управления ресурсами; сравнивать и оценивать различные методы, лежащие в основе планирования процессов; разрабатывать прикладные многопоточные приложения; пользоваться функциями ОС при оценке качества функционирования алгоритмов управления ресурсами вычислительной системы. Студент должен владеть навыками разработки прикладных многопоточных приложений; разработки алгоритмов планирования процессов; использования алгоритмов и механизмов синхронизации
ПК-5	Лабораторная работа. Тест Зачет. Экзамен	1-9	Знать: • структуру технической литературы и документации операционных систем в том числе в сети Интернет. Уметь: • использовать техническую	Студент должен знать структуру учебной литературы об операционных системах. Студент должен уметь формулировать поисковые запросы для решения технологических задач.	Студент должен знать структуру учебной литературы и документации операционных систем. Студент должен уметь использовать техническую литературу для получения информации об ОС UNIX, формулировать поисковые	Студент должен знать структуру технической литературы и документации операционных систем, анализировать качество источников информации о языке программирования. Студент должен уметь использовать техническую литературу для получения информации об ОС UNIX, формулировать поисковые

			<p>документацию операционной системы UNIX. Владеть:</p> <ul style="list-style-type: none"> • навыками поиска информации об операционной системе UNIX. 	<p>Студент должен владеть навыками поиска информации об ОС UNIX в учебной литературе и в сети Интернет</p>	<p>запросы для решения технологических задач. Студент должен владеть навыками поиска информации об ОС UNIX в учебной литературе и в сети Интернет</p>	<p>запросы для решения технологических задач и оценивать найденные решения. Студент должен владеть навыками поиска информации об ОС UNIX в технической и учебной литературе и в сети Интернет</p>
--	--	--	--	--	---	---

3. Методические рекомендации преподавателю по процедуре оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций

Целью процедуры оценивания является определение степени овладения студентом ожидаемыми результатами обучения (знаниями, умениями, навыками и (или) опытом деятельности).

Процедура оценивания степени овладения студентом ожидаемыми результатами обучения осуществляется с помощью методических материалов, представленных в разделе «Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций»

3.1 Критерии оценивания степени овладения знаниями, умениями, навыками и (или) опытом деятельности, определяющие уровни сформированности компетенций

Пороговый уровень (общие характеристики):

- владение основным объемом знаний по программе дисциплины;
- знание основной терминологии данной области знаний, стилистически грамотное, логически правильное изложение ответа на вопросы без существенных ошибок;
- владение инструментарием дисциплины, умение его использовать в решении стандартных (типовых) задач;
- способность самостоятельно применять типовые решения в рамках рабочей программы дисциплины;
- усвоение основной литературы, рекомендованной рабочей программой дисциплины;
- знание базовых теорий, концепций и направлений по изучаемой дисциплине;
- самостоятельная работа на практических и лабораторных занятиях, периодическое участие в групповых обсуждениях, достаточный уровень культуры исполнения заданий.

Продвинутый уровень (общие характеристики):

- достаточно полные и систематизированные знания в объеме программы дисциплины;
- использование основной терминологии данной области знаний, стилистически грамотное, логически правильное изложение ответа на вопросы, умение делать выводы;
- владение инструментарием дисциплины, умение его использовать в решении учебных и профессиональных задач;
- способность самостоятельно решать сложные задачи (проблемы) в рамках рабочей программы дисциплины;
- усвоение основной и дополнительной литературы, рекомендованной рабочей программой дисциплины;
- умение ориентироваться в базовых теориях, концепциях и направлениях по изучаемой дисциплине и давать им сравнительную оценку;
- самостоятельная работа на практических и лабораторных занятиях, участие в групповых обсуждениях, высокий уровень культуры исполнения заданий.

Высокий уровень (общие характеристики):

- систематизированные, глубокие и полные знания по всем разделам дисциплины;
- точное использование терминологии данной области знаний, стилистически грамотное, логически правильное изложение ответа на вопросы, умение делать обоснованные выводы;

- безупречное владение инструментарием дисциплины, умение его использовать в постановке и решении научных и профессиональных задач;
- способность самостоятельно и творчески решать сложные задачи (проблемы) в рамках рабочей программы дисциплины;
- полное и глубокое усвоение основной и дополнительной литературы, рекомендованной рабочей программой дисциплины;
- умение ориентироваться в основных теориях, концепциях и направлениях по изучаемой дисциплине и давать им критическую оценку;
- активная самостоятельная работа на практических и лабораторных занятиях, творческое участие в групповых обсуждениях, высокий уровень культуры исполнения заданий.

3.2 Описание процедуры выставления оценки

В зависимости от уровня сформированности каждой компетенции по окончании освоения дисциплины студенту выставляется оценка. Для дисциплин, изучаемых в течение нескольких семестров, оценка может выставляться не только по окончании ее освоения, но и в промежуточных семестрах. Вид оценки («отлично», «хорошо», «удовлетворительно», «неудовлетворительно», «зачтено», «незачтено») определяется рабочей программой дисциплины в соответствии с учебным планом.

Показатели и критерии, используемые при выставлении оценки подробно описаны в разделе «Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций».

Высокий уровень формирования компетенций соответствует оценке «отлично» за самостоятельные, контрольные работы и экзаменационную работу.

Продвинутый уровень формирования компетенций соответствует оценке «хорошо» за самостоятельные, контрольные работы и экзаменационную работу.

Пороговый уровень формирования компетенций соответствует оценке «удовлетворительно» за самостоятельные, контрольные работы и экзаменационную работу.

Оценка «отлично» выставляется студенту, у которого каждая компетенция (полностью или частично формируемая данной дисциплиной) сформирована на высоком уровне.

Оценка «хорошо» выставляется студенту, у которого каждая компетенция (полностью или частично формируемая данной дисциплиной) сформирована не ниже, чем на продвинутом уровне.

Оценка «удовлетворительно» выставляется студенту, у которого каждая компетенция (полностью или частично формируемая данной дисциплиной) сформирована не ниже, чем на пороговом уровне.

Оценка «неудовлетворительно» выставляется студенту, у которого хотя бы одна компетенция (полностью или частично формируемая данной дисциплиной) сформирована ниже, чем на пороговом уровне.

Оценка «зачет» выставляется студенту, у которого каждая компетенция (полностью или частично формируемая данной дисциплиной) сформирована не ниже, чем на пороговом уровне.

Оценка «незачтено» выставляется студенту, у которого хотя бы одна компетенция (полностью или частично формируемая данной дисциплиной) сформирована ниже, чем на пороговом уровне.

Приложение №2 к рабочей программе дисциплины «Операционные системы семейства UNIX и их администрирование»

Методические указания для студентов по освоению дисциплины

Занятия по данной дисциплине проводятся в различных формах. Все лекционные занятия проводятся в компьютерных классах с использованием мультимедиа-технологий, что позволяет выполнять немедленную демонстрацию концепций устройства UNIX-систем и принципов их администрирования, а также возможностей конкретных команд на практике, а также обеспечивает возможность изучения их студентом в интерактивном режиме. Практическое применение полученных знаний отрабатывается при выполнении лабораторных работ во время практических лабораторных занятий. Разбор типовых ошибок также осуществляется в ходе лабораторных занятий с привлечением метода мозгового штурма, активирующего креативные способности студентов.

Основной формой практической работы студентов по усвоению данного курса является выполнение ими самостоятельных лабораторных работ. По итогам каждой из лабораторных работ проводится промежуточная аттестация студента. Окончательная аттестация осуществляется в форме зачёта (в 5-м семестре) и экзамена (в 6-м семестре). Допуск к зачётам и экзаменам осуществляется в форме компьютерного тестирования, а также принимает во внимание средний балл по результатам промежуточных аттестаций.

Учебно-методическое обеспечение самостоятельной работы студентов по дисциплине

Для самостоятельной работы особенно рекомендуется использовать учебную литературу, указанную в разделе № 7 данной рабочей программы.

Также для подбора учебной литературы рекомендуется использовать широкий спектр интернет-ресурсов:

1. Электронно-библиотечная система «Университетская библиотека online» (www.biblioclub.ru) - электронная библиотека, обеспечивающая доступ к наиболее востребованным материалам-первоисточникам, учебной, научной и художественной литературе ведущих издательств (*регистрация в электронной библиотеке – только в сети университета. После регистрации работа с системой возможна с любой точки доступа в Internet.).

2. Для самостоятельного подбора литературы в библиотеке ЯрГУ рекомендуется использовать:

1. Личный кабинет (http://lib.uniyar.ac.ru/opac/bk_login.php) дает возможность получения on-line доступа к списку выданной в автоматизированном режиме литературы, просмотра и копирования электронных версий изданий сотрудников университета (учеб. и метод. пособия, тексты лекций и т.д.) Для работы в «Личном кабинете» необходимо зайти на сайт Научной библиотеки ЯрГУ с любой точки, имеющей доступ в Internet, в пункт меню «Электронный каталог»; пройти процедуру авторизации, выбрав вкладку «Авторизация», и заполнить представленные поля информации.

2. Электронная библиотека учебных материалов ЯрГУ (http://www.lib.uniyar.ac.ru/opac/bk_cat_find.php) содержит более 2500 полных текстов учебных и учебно-методических материалов по основным изучаемым дисциплинам, изданных в университете. Доступ в сети университета, либо по логину/паролю.

3. Электронная картотека «Книгообеспеченность» (http://www.lib.uniyar.ac.ru/opac/bk_bookreq_find.php) раскрывает учебный фонд научной библиотеки ЯрГУ, предоставляет оперативную информацию о состоянии книгообеспеченности дисциплин основной и дополнительной литературой, а также цикла дисциплин и специальностей. Электронная картотека «Книгообеспеченность» доступна в сети университета и через Личный кабинет.