МИНОБРНАУКИ РОССИИ
Ярославский государственный университет им. П.Г. Демидова

Кафедра вычислительных и программных систем

УТВЕРЖДАЮ
Декан факультета ИВТ
_____Д.Ю. Чалый
«_23_»____мая_____2023 г.

Рабочая программа дисциплины
«Операционные системы»

Направление подготовки
09.03.03 Прикладная информатика

Направленность (профиль)
«Информационные технологии в цифровой экономике»

Квалификация выпускника
Бакалавр

Форма обучения
очная

Программа рассмотрена
на заседании кафедры
от 21 апреля 2023 г.,
протокол № 8

Программа одобрена НМК
факультета ИВТ
протокол № 6 от
28 апреля 2023 г.

Ярославль

1. Цели освоения дисциплины

Целями дисциплины «Операционные системы» являются изучение принципов устройства операционных систем GNU/Linux, приёмов и методик их использования и администрирования с использованием командного интерфейса. Данные операционные системы широко распространён в индустрии, они применяются как во встраиваемых, так и в высоконагруженных системах. Грамотное использование командного интерфейса позволит студентам эффективно решать задачи в рамках любых современных операционных систем и обеспечит доступ к специализированным технологиям разработки и доставки программного обеспечения.

2. Место дисциплины в структуре ОП бакалавриата

Дисциплина «Операционные системы» относится к базовой части ОП бакалавриата.

Она базируется на знаниях и навыках, полученных студентами при изучении общепрофессиональных дисциплин компьютерного цикла, в наибольшей степени дисциплин «Основы программирования», «Архитектура ЭВМ и язык Ассемлера».

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**
Общепрофессиональные компетенции		
ОПК-5 Способен инсталлировать и сопровождать программное обеспечение информационных систем и баз данных, в том числе отечественного происхождения, с учетом информационной безопасности	ОПК-5.2 Знает принципы устройства операционных систем, приёмов и методики их Администрирования ОПК-5.3 Умеет использовать методы и алгоритмы обработки данных в сетевых компьютерных системах с учетом информационной безопасности	Знать • принципы обработки команд в командном интерпретаторе; • структуру файловой системы ОС GNU/Linux. Уметь • использовать инструменты командного интерфейса ОС GNU/Linux; • формировать скрипты на языке программирования Bash для автоматизации собственной деятельности. Владеть • навыками работы с инструментами командной строки для выполнения типичных задач (core utils); • навыками работы с системами получения информации (man, info) и документацией встроенной в

		приложения.

4. Объем, структура и содержание дисциплины

Общая трудоемкость дисциплины составляет 3 зач. ед., 108 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 ОС GNU/Linux и UNIX-подобных ОС	3	1					5	
2.	Базовые принципы использования командного интерфейса	3	2		3			5	
3.	Устройство файловой системы и основные команды для работы с файлами	3	2		4			4	
4.	Получение информации о командах из стандартных руководств	3	2		4			4	Лабораторная работа
5.	Использование текстовых редакторов Vim и GNU Emacs.	3			4			4	
6.	Перенаправление потоков ввода-вывода, построение конвейеров	3	1		3			4	
7.	Права доступа к файлам и каталогам	3	2		3			5	
8.	Написание скриптов на	3	2		4			5	Лабораторная работа

	языке Bash								
9.	Управление процессами в GNU/Linux	3	1		2			4	
10.	Управление пакетами с помощью инструментов dpkg и apt	3	2		3			5	
11.	Установка приложений из исходных кодов для языков C++ и Python	3	2		3			5	
12.	Система инициализации и управления службами SystemD	3	1		3			3,7	Лабораторная работа
	Всего		**18**		**30**		**0, 3**	**53,7**	**Зачёт**

Содержание разделов дисциплины:

1. История ОС GNU/Linux и UNIX-подобных ОС. В рамках лекции студенты знакомятся с контекстом создания, развития ОС GNU/Linux и других UNIX-подобных ОС. Студенты также знакомятся с типичными сферами применения данных ОС.
2. Базовые принципы использования командного интерфейса. Целью данных занятий является объяснение принципов передачи команд, их обработки и выполнения в рамках командного интерфейса. Студенты знакомятся с понятием текущего рабочего каталога, абсолютными и относительными путями.
3. Устройство файловой системы и основные команды для работы с файлами. Рассматривается вопрос организации файловой системы с точки зрения стандартов и их реализации. Производится знакомство с командами по манипулированию файловыми каталогами.
4. Получение информации о командах из стандартных руководств. Рассматриваются различные виды команд, которые пользователь может отдавать через командный интерфейс. Производится знакомство с системой руководств man, системой помощи по встроенным командам Bash, со встроенными руководствами в приложения.
5. Использование текстовых редакторов Vim и GNU Emacs. Целью данных занятий является получение практических навыков редактирования тестовых файлов с помощью указанных текстовых редакторов. Ввиду мощности и распространенности данные инструменты используются в профессиональной деятельности многими разработчиками ПО.
6. Перенаправление потоков ввода-вывода, построение конвейеров. Рассматриваются техники объединения нескольких приложений в конвейер для обработки текстовых данных. Рассматриваются приложения sort, uniq, head, tail, grep и т.п.
7. Права доступа к файлам и каталогам. Рассматриваются вопросы организации системы прав доступа: уникальные идентификаторы пользователей, групп, принадлежность пользователей к группам, организация прав доступа к файлам и каталогам.
8. Написание скриптов на языке Bash. Рассматривается возможность формирования файлов с заранее сформированными командами для автоматизации рутинных операций. Делается упор на более глубокое понимание процесса обработки и расширения строк в процессе выполнения.

9. Управление процессами в GNU/Linux. Рассматривается понятие процесса, а также рассматриваются инструменты (ps, top) для получения информации о процессах, включая вычислительные ресурсы и эффективные права доступа. Рассматривается вопрос о передаче сигналов процессам для управления их поведением.
10. Управление пакетами с помощью инструментов dpkg и apt. Рассматривается понятие пакета, его зависимостей и деталей процесса его установки, настройки и удаления. Происходит знакомство с низкоуровневой системой управления пакетами dpkg и её возможностями. Происходит знакомство с понятием репозитория, с высокоуровневыми инструментами подсистемы apt.
11. Установка приложений из исходных кодов для языков C++ и Python. Рассматривается принципиальная схема компиляции C++-приложения, рассматривается система пакетной обработки файлов Make, систем сборки Automake и CMake. Для Python-приложений рассматривается система установки pip-пакетов и управления ими.
12. Система инициализации и управления службами SystemD. Рассматривается процесс запуска ОС GNU/Linux, роль системы инициализации. Разбираются детали настройки SystemD, включая управление зарегистрированными службами и создание собственных описаний служб и включение их в процесс запуска ОС.

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 создания документов, презентаций: текстовый редактор Emacs (свободное ПО), конвертер документов Pandoc (свободное ПО), генератор статических сайтов Jekyll (свободное ПО), фреймворк для построения HTML-презентаций React.JS (свободное ПО);
- для проведения занятий: дистрибутив Debian GNU/Linux (свободное ПО) и другое свободное ПО, поставляемое в рамках дистрибутива;
- для координации занятий: свободная образовательная платформа Moodle;
- 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. Синицын, С. В., Операционные системы : учебник для вузов / С. В. Синицын, А. В. Батаев, Н. Ю. Налютин, М., Академия, 2010, 297с

2. Лукьянов, А. В., Современные операционные системы : метод. указания / А. В. Лукьянов; Яросл. гос. ун-т, Ярославль, ЯрГУ, 2012, 43с

3. Лукьянов, А. В., Современные операционные системы [Электронный ресурс] : метод. указания / А. В. Лукьянов; Яросл. гос. ун-т, Ярославль, ЯрГУ, 2012, 43с http://www.lib.uniyar.ac.ru/edocs/iuni/20120406.pdf

б) дополнительная:

1. Олифер, В. Г., Сетевые операционные системы : учеб. пособие для вузов / В. Г. Олифер, Н. А. Олифер. - 2-е изд., СПб., Питер, 2009, 668с

в) ресурсы сети «Интернет»

1. Введение в ОС Linux http://uneex.ru/Books/LinuxIntro

2. Викиучебник Linux от А до Я https://ru.wikibooks.org/wiki/Linux:_%D0%BE%D1%82_%D0%90_%D0%B4%D0%BE_%D0%AF

3. База знаний дистрибутива ArchLinux на русском языке https://wiki.archlinux.org/index.php/Main_page_(%D0%A0%D1%83%D1%81%D1%81%D0%BA%D0%B8%D0%B9)

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) :
Доцент 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, прикреплённый 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 execcmd, которая передаётся функции runcmd.

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

```
$ man 3 exec
```

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 конвееров

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

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

Парсе распознаёт команды по созданию конвеера. В результате разбора создаётся структура pipecmd. 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 ознакомится 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(), находящихся 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 сисемного 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 *.[chS]
```

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.
- Объем 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 репозиториев 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 задачиу
- 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-avaiable 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-avaiable/my-cool-site-config в каталоге /etc/nginx/sites-enabled: cd /etc/nginx/sites-enabled sudo ln -sf /etc/nginx/sites-avaiable/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

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`.

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, brasero.
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, brasero.
- 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 репозиториев 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 иходными 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.

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.

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.