

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ
Федеральное государственное бюджетное образовательное учреждение высшего образования
«ДАГЕСТАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
Факультет математики и компьютерных наук

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ
Современные операционные системы

Кафедра дискретной математики и информатики факультета математики
и компьютерных наук

Образовательная программа
02.04.02 Фундаментальная информатика и информационные технологии

Профиль подготовки
Информационные технологии

Уровень высшего образования
магистратура

Форма обучения
очная

Статус дисциплины: базовый

Махачкала, 2017

Рабочая программа дисциплины составлена в 2017 году в соответствии с требованиями ФГОС ВО по направлению подготовки 02.04.02 – Фундаментальная информатика и информационные технологии (уровень магистратуры) от 17.08.2015г. №.830

Разработчик(и): доцент кафедры дискретной математики и информатики, к.ф.-м.н. Ражабова Наима Шамильевна

Рабочая программа дисциплины одобрена:
на заседании кафедры дискретной математики и информатики от «13» января 2017 г., протокол № 5.

Зав. кафедрой AMB Магомедов А.М.
(подпись)

на заседании Методической комиссии _____ факультета от

« 17 » января 2017 г., протокол № 5.

Председатель BM Меджидов З.Г.
(подпись)

Рабочая программа дисциплины согласована с учебно-методическим

управлением « ____ » _____ 20 ____ г. AMC
(подпись)

1. Цели освоения дисциплины

Целями освоения дисциплины «Современные операционные системы» являются:

- получение фундаментальных знаний по организации операционных систем, включая изучение таких аспектов, как организацию файловых систем, создание и управление процессами, межпроцессные взаимодействия;
- формирование представлений об общих принципах построения сетевых служб.

Задачи курса:

- ознакомить с вопросами организации ядра операционной системы: управление процессами и ветвями, синхронизация процессов;
- ознакомиться с основами организации распределенных файловых систем;
- дать представление о базовых механизмах сетевого взаимодействия: потоки (Streams), связывание со стекком протоколов TCP/IP, программные гнезда (Sockets).

2. Место дисциплины в структуре ООП магистратуры

Дисциплина «Современные операционные системы» относится к базовой части образовательной программы магистратуры по направлению 02.04.02 - Фундаментальная информатика и информационные технологии и преподается на 1 курсе во 2 семестре (3 зачетные единицы). Изучение предмета завершается письменным экзаменом в конце семестра.

Дисциплина «Современные операционные системы» логически и содержательно взаимосвязана с такими дисциплинами, как «Объектно-ориентированные языки и системы программирования», «Сетевые технологии».

3. Компетенции обучающегося, формируемые в результате освоения дисциплины (перечень планируемых результатов обучения).

Компетенции	Формулировка компетенции из ФГОС ВО	Планируемые результаты обучения (показатели достижения заданного уровня освоения компетенций)
ОК-1	Способность к абстрактному мышлению, анализу, синтезу.	Знать: принципы самостоятельного поиска необходимой литературы для подготовки реферата и сдачи экзамена. Уметь: самостоятельно выполнять домашние задания, готовить реферат; анализировать принципы организации и идеологию UNIX-подобных систем. Владеть: навыками анализа архитектуры UNIX-подобных систем.
ОПК-3	Способность использовать и применять углубленные теоретические и практи-	Знать: основные команды и утилиты для работы пользователя в

	ческие знания в области фундаментальной информатики и информационных технологий.	операционной системе Ubuntu Linux. Уметь: решать задачи в режиме командной строки и создавать командные файлы для повторяющихся действий. Владеть: навыками использования команд для работы с файловой системой Ubuntu Linux.
ПК-4	Способность к применению методов математического и алгоритмического моделирования при решении теоретических и прикладных задач.	Знать: языки программирования и основы архитектуры современных операционных систем. Уметь: разрабатывать системные приложения на языке C. Владеть: навыками работы с современными инструментальными средствами (VMware Workstation Pro) для создания и отладки системных приложений.
ПК-10	Способность к преподаванию математических дисциплин и информатики в общеобразовательных организациях, профессиональных образовательных организациях и образовательных организациях высшего образования.	Знать: методы и организационные формы обучения информатике и информационным технологиям, современные образовательные стандарты. Уметь: использовать методическую и научно-техническую литературу и электронные источники информации для поиска и подбора учебных материалов в соответствии с дидактическими требованиями к курсу информатики и современным состоянием предметной области. Владеть: навыками планирования содержания и структуры занятия в соответствии с его целями, особенностями учебного материала и цикличностью на различных этапах изучения информатики.

4. Объем, структура и содержание дисциплины.

4.1. Объем дисциплины составляет 3 зачетные единицы, 108 академических часов.

4.2. Структура дисциплины.

№ п/п	Разделы и темы дисциплины	Семестр	Неделя семестра	Виды учебной работы, включая самостоятельную работу студентов и трудоемкость (в часах)	Самостоятель-	Формы текущего контроля успеваемости (по неделям семестра) Форма промежуточ-
-------	---------------------------	---------	-----------------	--	---------------	---

				Всего	Лекции	Практические занятия	Лабораторные занятия		ной аттестации (по семестрам)
Модуль 1. Принципы создания ОС									
1	Введение в архитектуру UNIX	2	1-2	6		2	2	2	Прием лабораторных работ (ЛР) и реферата (Р)
2	Ядро ОС: управление процессами и ветвями	2	3-4	10		2	2	4	ЛР, Р
3	Ядро ОС: взаимодействие между процессами	2	5-6	10		2	2	6	ЛР, Р
4	Файловая система и средства ввода/вывода.	2	7-8	10		2	2	6	ЛР, Р
	Подготовка к КСР	3						4	КСР
	Итого:			36		8	8	20	Модуль 1
Модуль 2. Базовые механизмы сетевых взаимодействий									
5	Программный интерфейс сокетов	3	9-11	8		2	4	8	ЛР, Р
6	Удаленный вызов процедур	3	12-14	16		2	2	10	ЛР, Р
	Подготовка к КСР	3						2	КСР
	Итого:			36		4	6	20	Модуль 2
	Подготовка к экзамену	3		36				36	Экзамен
	ИТОГО:	3		108		12	14	82	

4.3. Содержание дисциплины, структурированное по темам (разделам).

Лекции по курсу не предусмотрены.

4.3.1. Темы практических занятий

Модуль 1. Принципы создания ОС

Тема 1. Введение в архитектуру UNIX

1. Внутренняя структура ядра
2. Краткая характеристика основных подсистем ядра
3. Основы работы в ОС UNIX.

Тема 2. Ядро ОС: управление процессами и ветвями

1. Типы процессов
2. Жизненный путь процесса
3. Создание процессов и управление ими в UNIX.

Тема 3. Ядро ОС: взаимодействие между процессами

1. Сигналы.
2. Каналы.
3. Сообщения.
4. Семафоры

Тема 4. Файловая система и средства ввода/вывода

1. Краткий обзор файловых систем UNIX
2. Базовая архитектура драйверов
3. Потоки Streams.

Модуль 2. Базовые механизмы сетевых взаимодействий

Тема 5. Программный интерфейс сокетов

1. Протоколы TCP/IP
2. Сокеты в клиент-серверном взаимодействии

Тема 6. Удаленный вызов процедур

1. Описание механизма RPC
2. Связывание
3. Представление данных

4.3.2 Темы лабораторных занятий

Лабораторные работы предусмотрены по всем темам модуля. Целью каждой лабораторной работы является построение несложного распределенного приложения с использованием соответствующей технологии.

Лабораторная работа 1. Создание процессов и управление ими.

Лабораторная работа 2. Взаимодействие между процессами.

Лабораторная работа 3. Файловая система и средства ввода/вывода.

Лабораторная работа 4. Связывание со стеком протоколов TCP/IP.

Лабораторная работа 5. Программный интерфейс сокетов.

Лабораторная работа 6. Удаленный вызов процедур.

Примерная лабораторная работа на тему «Создание процессов и управление ими».

Цель работы: ознакомиться с системным вызовом для создания (распараллеливания) процессов.

В *Unix*-системах, *fork()* — системный вызов, создающий новый процесс (потомок), который является практически полной копией процесса-родителя, выполняющего этот вызов.

Между процессом-потомком и процессом-родителем существуют различия:

PID процесса-потомка отличен от PID процесса-родителя;

значению PPID процесса-потомка присваивается значение PID процесса-родителя;

Процесс-потомок получает собственную таблицу файловых дескрипторов, являющуюся копией таблицы процесса-родителя на момент вызова *fork()*. Это означает, что открытые файлы наследуются, но если процесс-потомок, например, закроет какой-либо файл, то это не повлияет на таблицу дескрипторов процесса-родителя.

- для процесса-потомка очищаются все ожидающие доставки сигналы;
- временная статистика выполнения процесса-потомка в таблицах ОС обнуляется;
- блокировки памяти и записи, установленные в процессе-родителе, не наследуются.

После вызова *fork()* алгоритм обычно разветвляется (в случае успешного выполнения функции *fork()*, она возвращает PID процесса-потомка родительскому процессу и нуль —

процессу-потомку. Если порождение процесса-потомка закончилось неудачей, функция `fork()` возвращает значение -1).

После `fork()` процесс-потомок чаще всего выполняет системный вызов `exec()`, загружающий в пространство процесса новую программу (именно так, и только так, в Unix-системе выполняется запуск программы в отдельном процессе). Так, первый (нулевой) процесс Unix (ядро системы) создаёт свою копию, чтобы запустить `init` (процесс с `PID = 1`), который в свою очередь создаёт дочерние процессы для запуска инициализации системы и терминалов.

Некоторые программы создают дочерние процессы не для запуска другой программы, а для выполнения параллельной задачи. Так, например, поступают простые сетевые серверы — при подсоединении клиента, сервер создаёт свою копию (дочерний процесс), которая обслуживает клиентское соединение и завершается по его закрытию. Родительский же процесс продолжает ожидать новых соединений.

Замечание. Вызов `fork()` выполняется довольно долго, так как требует копирования большого количества данных. Для того чтобы это обойти, некоторые сетевые серверы (например, веб-серверы *Apache* и *Lighttpd*), создают дочерние процессы заранее, чтобы уменьшить время отклика сервера. Также существуют «облегченные» реализации `fork()`. Например в ядре *Linux*, `fork()` отображает в новый процесс страницы памяти родительского процесса, вместо того чтобы их копировать. Новая страница создаётся только при изменении её содержимого одним из процессов, что существенно снижает время создания нового процесса (т. н. *copy-on-write*).

Задание 1. Выясните результат выполнения следующей программы.

```
#include <sys/types.h>
#include <unistd.h>
#include <stdio.h>
int fork(void);
void exit(int status);
main()
{ int p=fork();
  if(p==-1)
    {perror("Error fork");
     exit(1);
  }
  if(p==0)
    {
      printf("I am child!\n");
      printf("My process ID is %d\n",getpid());
      printf("My parent process ID is %d\n",getppid());
    }
  else
    { printf("I am parent!\n");
      printf("My process ID is %d\n",getpid());
      printf("My parent process ID is %d\n",getppid());
    }
}
```

Задание 2. Выясните результат выполнения команд `ps`, `ps -f`, `ps -l`, `pstree` (`pstree -up | less`)

Задание 3. Объясните работу следующей программы.

```
#include <sys/types.h>
#include <unistd.h>
#include <stdio.h>
int fork(void);
void exit(int status);
main()
{ int a=0;
  int p=fork();
  if(p==-1)
    {perror("Error fork");
     exit(1);
    }
  if(p==0)
    {
      printf("I am child!\n");
      printf("a= %d\n", ++a);
    }
  else
    { printf("I am parent!\n");
      printf("a=%d\n", a);
    }
  printf("End!\n");
}
```

Задание 4. Создать иерархию из 5 процессов.

Задание 5. Используя системный вызов `wait()` обеспечить выполнение процесса-потомка до процесса-родителя.

```
#include <sys/types.h>
#include <unistd.h>
#include <sys/wait.h>
int fork(void);
void exit(int status);
int wait(int *status);
int main()
{
  int status;
  int p=fork();
  if(p==-1)
    { printf("Error!\n");
      exit(1);
    }
  If (p==0)
    {
      printf("Hello from child!\n");
      printf("ABCDEFGHIJKLMN\n");
    }
  else
    {
      wait(&status);
    }
}
```

```

if (WIFEXITED(status)==0)
    printf("Not done!\n");
else
    printf("Done!\n");
}
return 0;
}

```

5. Образовательные технологии

Процесс изложения учебного материала сопровождается презентациями и демонстрацией решения задач в интерактивном режиме с использованием мультимедийного проектора. Предусмотрено регулярное общение с лектором и представителями российских и зарубежных компаний по электронной почте и по скайпу.

6. Учебно-методическое обеспечение самостоятельной работы студентов.

Пакет заданий для самостоятельной работы выдается в начале семестра, определяются предельные сроки их выполнения и сдачи. Самостоятельная работа студентов складывается из проработки учебного материала (настоятельно рекомендуется самостоятельное практическое решение всех разобранных упражнений), видео-лекций и соответствующих форумов интернет, решения всех заданий из индивидуальных заданий, решения рекомендуемых задач, подготовки к сдаче промежуточных отчетов по лабораторным работам и реферата, а также экзамена.

№	Вид самостоятельной работы	Вид контроля	Учебно-методическое обеспечение
1.	Изучение рекомендованной литературы и материалов соответствующих форумов интернет.	Контрольный фронтальный опрос, прием и презентация рефератов.	См. разделы 7.3, 8, 9 данного документа
3.	Подготовка к отчетам по лабораторным работам.	Проверка выполнения работ, опрос по теме работы.	См. разделы 7.3, 8, 9 данного документа
4.	Подготовка к сдаче промежуточных форм контроля.	Контрольные работы по каждому модулю и прием рефератов.	См. разделы 7.3, 8, 9 данного документа

Пакет заданий для самостоятельной работы выдается в начале семестра, определяются предельные сроки их выполнения и сдачи.

7. Фонд оценочных средств для проведения текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины.

7.1. Перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы.

Перечень компетенций с указанием этапов их формирования приведен в описании образовательной программы.

Компетенция	Знания, умения, навыки	Процедура освоения
-------------	------------------------	--------------------

ОК-1	<p>Знать: принципы самостоятельного поиска необходимой литературы для подготовки реферата и сдачи экзамена.</p> <p>Уметь: самостоятельно выполнять домашние задания, готовить реферат; анализировать принципы организации и идеологию UNIX-подобных систем.</p> <p>Владеть: навыками анализа архитектуры UNIX-подобных систем.</p>	Устный опрос, письменный опрос, подготовка реферата.
ОПК-3	<p>Знать: основные команды и утилиты для работы пользователя в операционной системе Ubuntu Linux.</p> <p>Уметь: решать задачи в режиме командной строки и создавать командные файлы для повторяющихся действий.</p> <p>Владеть: навыками использования команд для работы с файловой системой Ubuntu Linux.</p>	Письменный опрос, выполнение лабораторных заданий.
ПК-4	<p>Знать: языки программирования и основы архитектуры современных операционных систем.</p> <p>Уметь: разрабатывать системные приложения на языке C.</p> <p>Владеть: навыками работы с современными инструментальными средствами (<i>VMware Workstation Pro</i>) для создания и отладки системных приложений.</p>	Устный опрос, выполнение лабораторных заданий, подготовка реферата.
ПК-10	<p>Знать: методы и организационные формы обучения информатике и информационным технологиям, современные образовательные стандарты.</p> <p>Уметь: использовать методическую и научно-техническую литературу и электронные источники информации для поиска и подбора учебных материалов в соответствии с дидактическими требованиями к курсу информатики и современным состоянием предметной области.</p> <p>Владеть: навыками планирования содержания и структуры занятия в соответствии с его целями, особенностями учебного материала и цикличностью на различных этапах изучения информатики.</p>	Устный опрос, выполнение лабораторных заданий, подготовка реферата. Подготовка к контрольной работе.

7.2. Описание показателей и критериев оценивания компетенций, описание шкал оценивания.

ОК-1

Схема оценки уровня формирования компетенции «Способность к абстрактному мышлению, анализу, синтезу».

Уровень	Показатели (что обучающийся должен продемонстрировать)	Оценочная шкала		
		Удовлетворительно	Хорошо	Отлично
Базовый	Способен выявить	Допускает ошибки	В большинстве	Свободно и уве-

Способен самостоятельно выявить надежные источники информации по указанной проблеме, собрать информацию, необходимую и достаточную для решения проблемы, обработать ее, избрать метод решения проблемы в стандартных условиях и решить ее.	<p>достоверные источники информации</p> <p>Способен обработать, проанализировать и синтезировать информацию.</p> <p>Способен выбрать метод решения проблемы в стандартных условиях.</p> <p>Способен решить проблему, используя выбранный метод.</p>	в определении достоверности источников информации, способен правильно решать только типичные, наиболее часто встречающиеся проблемы в конкретной области (обрабатывать информацию, выбирать метод решения проблемы и решать ее).	случаев способен выявить достоверные источники информации, обработать, анализировать и синтезировать предложенную информацию, выбрать метод решения проблемы и решить ее. Допускает единичные серьезные ошибки в решении проблем, испытывает сложности в редко встречающихся или сложных случаях решения проблем.	ренно находит достоверные источники информации, оперирует предоставленной информацией, отлично владеет навыками анализа и синтеза информации, знает все основные методы решения проблем, предусмотренные учебной программой, знает типичные ошибки и возможные сложности при решении той или иной проблемы и способен выбрать и эффективно применить адекватный метод решения конкретной проблемы.
--	---	--	---	--

ОПК-3

Схема оценки уровня формирования компетенции «Способностью использовать и применять углубленные теоретические и практические знания в области фундаментальной информатики и информационных технологий».

Уровень	Показатели (что обучающийся должен продемонстрировать)	Оценочная шкала		
		Удовлетворительно	Хорошо	Отлично
Базовый Способен создавать различные типы системных приложений для UNIX-подобных систем.	Способен использовать команды и утилиты для работы с ОС Ubuntu. Способен создавать клиент-серверные системы на основе транспортных протоколов (UDP и TCP).	Допускает ошибки при решении практических задач администрирования ОС Ubuntu, способен создавать простые программные системы на основе изученных технологий.	В большинстве случаев способен решать задачи администрирования Linux Ubuntu.	Свободно и уверенно владеет навыками создания системных приложений.

ПК-4

Схема оценки уровня формирования компетенции «Способность к применению методов математического алгоритмического моделирования при решении теоретических и прикладных задач».

Уровень	Показатели (что обучающийся должен продемонстрировать)	Оценочная шкала		
		Удовлетвори-	Хоро-	Отлич-

		тельно	шо	но
Базовый	Процент выполненных заданий, связанных с применением методов математического алгоритмического моделирования	50%	65%	80%

ПК-10

Схема оценки уровня формирования компетенции «Способность к преподаванию математических дисциплин и информатики в общеобразовательных организациях, профессиональных образовательных организациях и образовательных организациях высшего образования».

Уровень	Показатели (что обучающийся должен продемонстрировать)	Оценочная шкала		
		Удовлетворительно	Хорошо	Отлично
Базовый	Процент выполненных заданий, связанных с разработкой учебных планов занятий по информатике	50%	65%	80%

Если хотя бы одна из компетенций не сформирована, то положительная оценки по дисциплине быть не может.

7.3. Типовые контрольные задания

7.3.1 Вопросы для самостоятельной работы

1. Классификация операционных систем. Различные схемы классификации.
2. Требования к характеристикам операционных систем. Зависимость требований от типа операционных систем.
4. Архитектура UNIX. Двухуровневая модель системы.
5. Работа в операционной системе UNIX. Типы файлов.
6. Структура файловой системы UNIX. Краткое описание основных каталогов.
7. Создание и навигация по файлам и каталогам: *touch, ls, mv, rm, cp, ln, pwd, cd, mkdir, rmdir, find*.
8. Выполнение программ в среде UNIX(создание, компиляция, запуск).
9. Основные системные функции для работы с файлами.
10. Понятие процесса. Типы процессов. Атрибуты процесса.
11. Жизненный путь процесса.
12. Сигналы.
13. Системные вызовы для создания процессов. Элементы создания процессов.

7.3.2 Контрольные вопросы

1. Назначение и функции операционных систем (ОС).
2. Ядро ОС: управление процессами и ветвями, синхронизация процессов, обработка прерываний, управление памятью, распределение времени процессора, приоритетное планирование, управление доступом.
3. Стандарты интерфейсов с прикладными программами (POSIX).
4. Файловая система и средства ввода/вывода.

5. Управление вычислительным процессом.
6. Базовые механизмы сетевых взаимодействий.
7. Потоки (Streams).
8. Связывание потоков со стеком протоколов TCP/IP.
9. Программные гнезда (Sockets).
10. Вызовы удаленных процедур.
11. Распределенные файловые системы.
12. Сетевая файловая система.
13. Организация распределенной обработки информации.

Примерный контрольный тест по первому модулю

1. Даны следующие команды WSH:

- 1) `var W1 = new ActiveXObject("WScript.Shell");`
- 2) `var E1 = W1.Exec("calc");`
- 3) `var W2 = WScript.CreateObject("WScript.Shell")`
- 4) `W2.Run('Calc');`

Для запуска калькулятора Windows используются следующие команды:

- а) 1 и 2; б) 3 и 4; в) 1 и 4; г) 3 и 2

2. Замечательной особенностью WSH является возможность комбинировать все установленные в системе скриптовые языки в одном файле с расширением:

- а) `js`; б) `vbs`; в) `wsf`; г) `pls`

3. Текстовый редактор, входящий в практически каждый Unix дистрибутив

- а) `cd`
б) `ls`
в) `vi`
г) `notepad`

4. Сравнение файлов производится командой

- а) `man`
б) `ls`
в) `cmp`
г) `cp`

5. Для вывода имен файлов текущего каталога используется команда:

- а) `man`; б) `ls`; в) `mkdir`; г) `cp`

6. Имя учетной записи встроенного администратора любой Unix системы

- а) `user`
б) `admin`
в) `root`
г) `administrator`

7. Выполняя функцию `read`, ядро проверяет, правильно ли задан параметр "дескриптор файла", а также был ли файл предварительно открыт процессом для чтения. Функция `read(f,buf,20)` возвращает

- а) 20
б) 16
в) 1024

г) количество фактически прочитанных байт, которое может быть и не равно 20:

8. Выберите неверное утверждение:

- а) Процесс в ОС UNIX - это программа, выполняемая в собственном виртуальном адресном пространстве.
б) Когда пользователь входит в систему, автоматически создается процесс, в котором выполняется программа командного интерпретатора.

в) Если командному интерпретатору встречается команда, соответствующая выполняемому файлу, то он создает новый процесс и запускает в нем соответствующую программу, начиная с функции *main*.

г) Запущенная программа не может создать процесс и запустить в нем другую программу.

д) Дочерний процесс является точной копией родительского процесса и наследует атрибут родителя **переменные окружения**

9. Выберите неверное утверждение:

а) Все процессы в UNIX связаны отношениями процесс-родитель – процесс-ребенок.

б) Процесс *init* является родительским для всех остальных процессов в системе UNIX.

в) Демоны не связаны с пользовательским сеансом работы, но могут управляться пользователем.

г) Когда процесс – родитель завершает свою работу до завершения выполнения процесса – ребенка, идентификатор родительского процесса в данных ядра процесса – ребенка (PPID) изменяет свое значение на значение 1, соответствующее идентификатору процесса *init*.

д) Системный вызов *fork* служит для создания нового процесса в операционной системе UNIX.

д) при достижении конца функции *main()* неявно вызывается функция *exit* со значением параметра 0.

10. Сигнал аварийного завершения процесса — гарантированного, без освобождения ресурсов:

а) SIGILL (4)

б) SIGKILL (9)

в) SIGPIPE (13)

г) SIGTERM (15)

11. Выберите неверное утверждение:

а) Возврат из функции *fork* происходит как в родительский, так и в дочерний процесс.

б) Возврат значения -1 функцией *fork* свидетельствует об ошибке.

в) В случае ошибки при вызове *fork*, возврат происходит только в процесс, выполнивший системный вызов.

г) Системный вызов *execve* предназначен как для создания программ, так и для их запуска.

д) Функция *wait* позволяет процессу контролировать выполнение потомков.

12. Выберите неверное утверждение:

а) Системные процессы являются частью ядра и всегда расположены в оперативной памяти.

б) С помощью системного вызова *kill* процесс не может послать сигнал самому себе.

в) Время жизни прикладного процесса ограничено сеансом работы пользователя.

г) Число параметров, переданных функции *main* определяет аргумент *argc*.

д) Демоны не могут непосредственно управляться пользователем.

13. Что напечатает следующая программа?

```
int main()
{ int a=2;
  if ((p=fork())<0) exit(1);
  if (p==0){ printf(“%d\n”, ++a);}
  else { printf(“%d\n”, a); }
  printf(“end\n”);
  return 0;}
```

1. Структура файловой системы UNIX. Краткое описание основных каталогов. Доступ к файлам
2. Процессы. Типы процессов. Атрибуты процесса. Жизненный путь процесса. Создание процессов. Основы управления процессом. Утилиты UNIX для управления процессами. Состояния процесса.
3. Сигналы. Группы и сеансы. Управление сигналами.
4. Взаимодействие между процессами. (каналы и сообщения).
5. Взаимодействие между процессами. (семафоры и разделяемая память).
6. Пример использования сокетов TCP и UDP.
7. Файловые системы Unix. System V.
8. Файловые системы Unix. BSD(FFS).
9. Подсистема ввода-вывода. Управление передачей данных. Драйверы и модули.
10. Поддержка сети в Unix. Программный интерфейс сокетов.

7.4. Методические материалы, определяющие процедуру оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций.

Общий результат выводится как интегральная оценка, складывающаяся из текущего контроля - 50% и промежуточного контроля - 50%.

Текущий контроль по дисциплине включает:

- посещение занятий - 10 баллов,
- подготовка реферата – 20 баллов,
- выполнение лабораторных заданий – 20 баллов,
- выполнение домашних (аудиторных) контрольных работ - 50 баллов.

Промежуточный контроль по дисциплине включает:

- прием лабораторных работ - 30 баллов,
- письменная контрольная работа - 50 баллов,
- тестирование - 20 баллов.

8. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины.

а) основная литература:

1. Робачевский, А. М. Операционная система UNIX [Текст] / А. М. Робачевский, С. А. Немнюгин, О. Л. Стесик. — СПб.: БХВ-Петербург, 2010. – 656 с.
2. Таненбаум, Э. Современные операционные системы [Текст] / Э. Таненбаум. – СПб.: Питер, 2011. – 1120 с.
3. Таненбаум, Э. Распределенные системы. Принципы и парадигмы [Текст] / Э. Таненбаум, М. Э. Ван-Стеен. – СПб.: Питер, 2003. – 877 с.

б) дополнительная литература:

4. Соловьёв В.М. Практикум по операционным системам. Подготовка системных администраторов информационной компьютерной сети Саратовского государственного университета. Linux-системы — Саратов: Изд-во Сарат. ун-та, 2009.
5. Гласс Г., Эйблс К. UNIX для программистов и пользователей. — СПб.: БХВ-Петербург, 2004.

6. Курячий Г. В., Маслинский К. А. Операционная система Linux. — 2005.

9. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины.

Видеокурсы лекций:

- 1) <https://www.coursera.org/>
- 2) <https://www.udacity.com/>
- 3) <https://www.intuit.ru/>

Форумы по компьютерным наукам и программированию:

- 1) www.stackoverflow.com
- 2) <http://www.cyberforum.ru/>

10. Методические указания для обучающихся по освоению дисциплины.

Все упражнения, приведенные на лекции с решениями, следует прорабатывать сразу после лекции. Самостоятельная работа студентов складывается из:

- проработки материала практических занятий (настоятельно рекомендуется самостоятельное практическое решение всех разобранных упражнений);
- изучения рекомендованной литературы и материалов соответствующих форумов интернет;
- подготовки к отчетам по лабораторным работам;
- подготовки к сдаче промежуточных форм контроля (контрольных работ и сдаче реферата).

Пакет лабораторных заданий рассчитан на семестр. Рекомендуется выполнять и сдавать задания своевременно с прохождением соответствующего материала.

Модули и темы для самостоятельного изучения	Виды и содержание самостоятельной работы
Модуль 1. Принципы создания ОС Тема 1. Введение в архитектуру UNIX	Проработка материала практических занятий. Изучение рекомендованной литературы и материалов соответствующих форумов интернет для подготовки реферата.
Модуль 1. Принципы создания ОС Тема 2. Ядро ОС: управление процессами и ветвями	Проработка материала практических занятий. Изучение рекомендованной литературы и материалов соответствующих форумов интернет для подготовки реферата.
Модуль 1. Принципы создания ОС Тема 3. Ядро ОС: взаимодействие между процессами	Проработка материала практических занятий. Подготовка к отчетам по лабораторным работам. Подготовка к сдаче промежуточных форм контроля.
Модуль 1. Принципы создания ОС Тема 4. Файловая система и средства ввода/вывода	Проработка материала практических занятий. Изучение рекомендованной литературы и материалов соответствующих форумов интернет для подготовки реферата. Подготовка к отчетам по лабора-

	торным работам
Модуль 1. Принципы создания ОС Тема 4. Файловая система и средства ввода/вывода	Проработка материала практических занятий. Изучение рекомендованной литературы и материалов соответствующих форумов интернет для подготовки реферата. Подготовка к отчетам по лабораторным работам. Подготовка к сдаче промежуточных форм контроля.
Модуль 2. Базовые механизмы сетевых взаимодействий Тема 5. Программный интерфейс сокетов	Проработка материала практических занятий. Изучение рекомендованной литературы и материалов соответствующих форумов интернет для подготовки реферата. Подготовка к отчетам по лабораторным работам.
Модуль 2. Базовые механизмы сетевых взаимодействий Тема 6. Удаленный вызов процедур	Проработка материала практических занятий. Изучение рекомендованной литературы и материалов соответствующих форумов интернет для подготовки отчетов по лабораторным работам.

11. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень программного обеспечения и информационных справочных систем.

При осуществлении образовательного процесса студентами и профессорско-преподавательским составом используются следующее программное обеспечение: MicrosoftWindows, UbuntuLinux, VMware Workstation Pro, MicrosoftVisualStudioUltimate, RationalRose, Skype. Также студентам предоставляется доступ к российским и международным электронным библиотекам через компьютеры университета.

12. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине.

Имеется необходимая литература в библиотеке, медиапроектор и компьютер для проведения занятий-презентаций.

Лабораторные занятия проводятся в компьютерных классах с необходимым программным обеспечением.

Вся основная литература предоставляется студенту в электронном формате.