

Минобрнауки России

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «ВГУ»)

УТВЕРЖДАЮ



Заведующий кафедрой
Борисов Дмитрий Николаевич
Кафедра информационных систем

21.04.2021

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ

Б1.О.26 Операционные системы

1. Код и наименование направления подготовки/специальности:

09.03.04 Программная инженерия

2. Профиль подготовки/специализация: Информационные системы и сетевые технологии

3. Квалификация (степень) выпускника:

Бакалавриат

4. Форма обучения:

Очная

5. Кафедра, отвечающая за реализацию дисциплины:

Кафедра информационных систем

6. Составители программы:

Савинков Андрей Юрьевич, д.т.н., профессор

7. Рекомендована:

8. Учебный год:

2023-2024

9. Цели и задачи учебной дисциплины:

Целями освоения учебной дисциплины являются:

- Формирование необходимых знаний, умений и навыков для использования современных операционных систем при решении задач профессиональной деятельности
- Формирование необходимых знаний, умений и навыков для выбора операционной системы и ее конфигурации для реализации целевой информационной системы

Задачи учебной дисциплины:

- изучение архитектуры и базовых алгоритмов работы операционных систем
- изучение базовых принципов управления ресурсами в операционных системах
- изучение механизмов синхронизации и межпроцессного взаимодействия
- изучение принципов построения и работы подсистемы ввода-вывода
- изучение принципов построения и работы сетевой подсистемы
- изучение принципов построения и работы файловых систем
- изучение базовых механизмов обеспечения информационной безопасности в операционных системах
- изучение основ администрирования операционных систем и формирование практических навыков администрирования
- формирование практических навыков написания сценариев командной оболочки

- формирование практических навыков установки и конфигурирования операционных систем
- формирование практических навыков использования базовых системных утилит

10. Место учебной дисциплины в структуре ООП:

Обязательная

11. Планируемые результаты обучения по дисциплине/модулю (знания, умения, навыки),соотнесенные с планируемыми результатами освоения образовательной программы (компетенциями выпускников) и индикаторами их достижения:

Код и название компетенции	Код и название индикатора компетенции	Знания, умения, навыки
ОПК-2 Способен использовать современные информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности;	ОПК-2.1 Знает современные информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности.	Знать основы внутреннего устройства и основные алгоритмы работы операционных систем, методы разграничения доступа, основы системного программирования и системного администрирования
ОПК-7 Способен применять в практической деятельности основные концепции, принципы, теории и факты, связанные с информатикой;	ОПК-7.1 Знает основные языки программирования и работы с базами данных, операционные системы и оболочки, современные программные среды разработки информационных систем и технологий.	Знать основные команды интерфейса командной строки Unix, Linux и Windows, основы API операционных систем, основы синхронизации процессов и межпроцессного взаимодействия
ОПК-2 Способен использовать современные информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности;	ОПК-2.2 Умеет выбирать современные информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности.	Уметь правильно выбрать и конфигурировать операционную систему для решения задач профессиональной деятельности

ОПК-2 Способен использовать современные информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности;	ОПК-2.3 Имеет навыки применения современных информационных технологий и программных средств, в том числе отечественного производства, при решении задач профессиональной деятельности.	Иметь навыки практической разработки сценариев командной оболочки и практического использования API операционных систем
---	--	---

12. Объем дисциплины в зачетных единицах/час:

3/108

Форма промежуточной аттестации:

Зачет

13. Трудоемкость по видам учебной работы

Вид учебной работы	Семестр 5	Всего
Аудиторные занятия	50	50
Лекционные занятия		0
Практические занятия	34	34
Лабораторные занятия	16	16
Самостоятельная работа	58	58
Курсовая работа		0
Промежуточная аттестация	0	0
Часы на контроль		0
Всего	108	108

13.1. Содержание дисциплины

п/п	Наименование раздела дисциплины	Содержание раздела дисциплины	Реализация раздела дисциплины с помощью онлайн-курса, ЭУМК
1. Лекции			
2. Практические занятия			

2.1	Введение в ОС	Роль ОС в вычислительной системе, преимущества использования ОС, история возникновения и развития ОС, обзор версий UNIX, GNU/Linux, MS Windows и MAC OS	
2.2	Основные понятия и определения предметной области ОС	Понятие прерывания, виртуальной памяти, процесса, легковесного процесса (LWP), ядра ОС, системного вызова, принципы реализации многозадачности, понятие ресурса, основные принципы управления ресурсами, стратегии и дисциплины управления.	
2.3	Архитектура ОС	Архитектура ОС с монолитным ядром, модульным ядром, микроядром и экзоядром, гипервизор, влияние архитектуры на характеристики ОС.	
2.4	Пользовательский интерфейс ОС	Классификация интерфейсов, интерфейсы, управляемые сообщениями, основы реализации пользовательского интерфейса в Linux и Windows/	
2.5	Обработка прерываний в ОС	Модель прерывания, разделение канала запроса прерывания несколькими устройствами, уровни обработки прерывания, основы обработки прерываний в UNIX, Linux и Windows.	
2.6	Управление процессами	Состояния процесса, планирование доступа, планирование загрузки процессоров, особенности планирования загрузки процессоров в системах реального времени и многопроцессорных системах, взаимодействие процессов (IPC), взаимная блокировка процессов (тупики), основы реализации IPC в UNIX, Linux и Windows	

2.7	Управление памятью	Локализация ссылок, иерархия памяти, динамическое распределение памяти, распределитель SLAB, виртуальная память, стратегии и дисциплины управления виртуальной памятью, API управления памятью Linux и Windows	
2.8	Подсистема Ввода-Вывода	Взаимодействие процессора с внешними устройствами, драйвер устройства, сетевая подсистема, автоматическое определение подключенных устройств (P&P), основы реализации ввода-вывода и сети Linux и Windows.	
2.9	Файловая система	Общие принципы работы файловой системы, пространство имен файловой системы, монтирование файловой системы, надежность файловой системы и типовые отказы, проверка и восстановление файловой системы, RAID, основы файловых систем FAT, UFS, ext, NTFS.	
2.10	Основы информационной безопасности в ОС	Принципы разграничения доступа, дискреционная (избирательная) и мандатная модели, изолированная программная среда, принципы безопасного хранения паролей, основы разграничения доступа в Linux и Windows	
3. Лабораторные занятия			
3.1	Изучение интерфейса командной строки GNU/Linux и MS Windows	Получение информации о текущей конфигурации, создание, удаление, копирование и перемещение файлов и каталогов, поиск файлов по различным критериям, основы grep и vi, альтернативные потоки данных в файлах NTFS	

3.2	Сценарии командной оболочки	Основы программирования на языке сценариев командной оболочки: переменные, условия, циклы, работа с файлами, вызов внешних утилит.	
3.3	Сигналы Linux	Основы обработки сигналов в сценариях командной оболочки и в программах на языке C: переопределение обработчиков, блокировка отдельных сигналов, сигналы и потоки.	
3.4	Структурная обработка исключений (SEH) Windows	Основы структурной обработка исключений на языке C в ОС Windows: <code>__try __except</code> , <code>__try __finally</code> , поиск обработчика в стеке потока.	
3.5	Процессы и IPC в Linux	Создание процесса для исполнения указанной программы, наследование открытых файлов и использование унаследованных файловых дескрипторов для взаимодействия процессов, семафоры, разделяемая память, очереди сообщений, потоки POSIX, условные мьютексы и переменные в потоках POSIX.	
3.6	Процессы и IPC в Windows	Создание процесса для исполнения указанной программы, наследование открытых файлов и использование унаследованных файловых дескрипторов для взаимодействия процессов, семафоры, события, мьютексы, разделяемая память, потоки Windows, критические секции и условные переменные в потоках Windows.	

13.2. Темы (разделы) дисциплины и виды занятий

№ п/п	Наименование темы (раздела)	Лекционные занятия	Практические занятия	Лабораторные занятия	Самостоятельная работа	Всего
1	Введение в ОС	0	2	0	0	2

2	Основные понятия и определения предметной области ОС	0	4	0	4	8
3	Архитектура ОС	0	2	0	2	4
4	Пользовательский интерфейс ОС	0	2	0	2	4
5	Обработка прерываний в ОС	0	4	0	4	8
6	Управление процессами	0	6	0	6	12
7	Управление памятью	0	6	0	6	12
8	Подсистема Ввода-Вывода	0	2	0	2	4
9	Файловая система	0	4	0	4	8
10	Основы информационной безопасности в ОС	0	2	0	2	4
11	Изучение интерфейса командной строки GNU/Linux и MS Windows	0	0	2	2	4
12	Сценарии командной оболочки	0	0	6	6	12
13	Сигналы Linux	0	0	1	3	4

14	Структурная обработка исключений (SEH) Windows	0	0	1	3	4
15	Процессы и IPC в Linux	0	0	2	6	8
16	Процессы и IPC в Windows	0	0	4	6	10
		0	34	16	58	108

14. Методические указания для обучающихся по освоению дисциплины

Дисциплина требует работы с файлами-презентациями лекций и соответствующими главами рекомендованной основной литературы, а также, обязательного выполнения всех лабораторных заданий в компьютерном классе.

Самостоятельная работа проводится в компьютерных классах ФКН с использованием методических материалов расположенных на учебно-методическом сервере ФКН fs.cs.vsu.ru/library и на сервере Moodle ВГУ moodle.vsu.ru. Во время самостоятельной работы студенты используют электроннобиблиотечные системы, доступные на портале Зональной Библиотеки ВГУ по адресу www.lib.vsu.ru. Часть заданий может быть выполнена вне аудиторий на домашнем компьютере, после копирования методических указаний и необходимого ПО с учебно-методического сервера ФКН.

При использовании дистанционных образовательных технологий и электронного обучения выполнять все указания преподавателей, вовремя подключаться к online занятиям, ответственно подходить к заданиям для самостоятельной работы.

15. Перечень основной и дополнительной литературы, ресурсов интернет, необходимых для освоения дисциплины

№ п/п	Источник
1	Сафонов В.О. Основы современных операционных систем / В.О. Сафонов. – Москва : Интернет-Университет Информационных Технологий, 2011. – 584 с. // ЭБС Университетская библиотека. – URL: http://biblioclub.ru/index.php?page=book&id=233210

б) дополнительная литература:

№ п/п	Источник
1	Назаров С. В. Современные операционные системы: учебное пособие / С.В. Назаров, А.И. Широков. – Москва : Интернет-Университет Информационных Технологий, 2011. – 280 с. // ЭБС Университетская библиотека. – https://biblioclub.ru/index.php?page=book&id=233197

в) информационные электронно-образовательные ресурсы:

№ п/п	Источник
1	Библиотека ВГУ, http://www.lib.vsu.ru
2	Сервер учебно-методических материалов ФКН, \\fs.cs.vsu.ru\Library
№ п/п	Источник
3	Образовательный портал "Электронный университет ВГУ", http://edu.vsu.ru

16. Перечень учебно-методического обеспечения для самостоятельной работы

№ п/п	Источник
1	Сервер учебно-методических материалов ФКН, \\fs.cs.vsu.ru\Library
2	Образовательный портал "Электронный университет ВГУ", http://edu.vsu.ru

17. Образовательные технологии, используемые при реализации учебной дисциплины, включая дистанционные образовательные технологии (ДОТ), электронное обучение (ЭО), смешанное обучение):

Презентации с демонстрацией иллюстративных и графических материалов, анимации, блок-схем алгоритмов и примеров исходного кода, демонстрацией выполнения команд операционной системой, лабораторные работы.

При реализации дисциплины могут использоваться технологии электронного обучения и дистанционные образовательные технологии на базе портала edu.vsu.ru, а также другие доступные ресурсы сети Интернет.

18. Материально-техническое обеспечение дисциплины:

- 1 Лекционная аудитория, оснащенная видеопроектором.
- 2 Компьютерный класс для проведения лабораторных занятий, оснащенный видеопроектором, компьютерами с ОС Windows с установленными средой разработки MS Visual Studio и виртуальной машиной VirtualBox с образом операционной системы GNU/Linux. Объем свободной после загрузки ОС оперативной памяти на рабочее место не менее 4 ГБ (требуется для виртуальных машин).

19. Оценочные средства для проведения текущей и промежуточной аттестаций

Порядок оценки освоения обучающимися учебного материала определяется содержанием следующих разделов дисциплины:

№ п/п	Разделы дисциплины (модули)	Код компетенции	Код индикатора	Оценочные средства для текущей аттестации

1	Основные понятия и определения предметной области ОС Архитектура ОС Пользовательский интерфейс ОС Обработка прерываний в ОС Управление процессами Управление памятью Подсистема Ввода-Вывода Файловая система Основы информационной безопасности в ОС	ОПК-2	ОПК-2.1	Собеседование
2	Введение в ОС	ОПК-7	ОПК-7.1	Собеседование
3	Изучение интерфейса командной строки GNU/Linux и MS Windows Сценарии командной оболочки	ОПК-2	ОПК-2.2	Контрольная работа
4	Сигналы Linux Структурная обработка исключений (SEH) Windows Процессы и IPC в Linux Процессы и IPC в Windows	ОПК-2	ОПК-2.3	Контрольная работа

Промежуточная аттестация

Форма контроля - Зачет

Оценочные средства для промежуточной аттестации

- 1 Собеседование
- 2 Практическое задание

20 Типовые оценочные средства и методические материалы, определяющие процедуры оценивания

20.1 Текущий контроль успеваемости

Задания к контрольной работе

1. Вывести на экран имя текущего каталога
2. В домашнем каталоге создать пустой файл с именем file1
3. В домашнем каталоге создать жесткую ссылку HL1 на файл file1
4. В домашнем каталоге создать символическую ссылку SL1 на файл file1
5. В домашнем каталоге создать каталог DIR1 и переместить в него файлы file1, HL1 и SL1
6. Вывести на экран подробные сведения о файлах каталога DIR1
7. Удалить DIR1 и его содержимое
8. Найти в каталоге /bin все файлы, размером менее 1 кб, кроме символических ссылок и вывести на экран подробные сведения о них

9. Найти в каталоге /bin все файлы, содержащие в имени подстроку 'ls'
10. Найти в каталоге /bin все файлы, содержащие в имени слово 'ls'
11. Написать сценарий, выводящий на экран приветствие текущему пользователю
12. Написать сценарий, выводящий на экран переданные ему параметры в обратном порядке
13. Написать сценарий, выводящий на экран переданные ему параметры в алфавитном порядке по возрастанию
14. Написать сценарий, реализующий калькулятор для арифметических операций
15. Написать сценарий `mkfile name [mode]`, где `name` - имя файла и `mode` - опциональный аргумент, определяющий режим доступа к файлу в формате `chmod`, файл должен создаваться, только если он еще не существует, необходимо проверить успех создание файла, существование второго аргумента и успешность изменения режима доступа
16. Написать сценарий `prn -n <number> -t <timeout> -- text`, который печатает текст "text" number раз с задержкой `timeoqt` (секунд) перед каждым выводом текста
17. Написать сценарий, выводящий на экран имена всех исполняемых файлов из каталогов, перечисленных в переменной среды `PATH`
18. Написать сценарий, реализующий вычисление факториала числа
19. Написать сценарий, реализующий аналог утилиты `cat`
20. Написать сценарий, реализующий упрощенный аналог утилиты `tree` (не нужно рисовать дерево, но необходимо обеспечить отступ при выводе файлов из вложенных каталогов, пропорциональный глубине вложения)
21. Написать на языке C программу для ОС Linux, блокирующую сигнал `SIGINT`
22. Написать на языке C программу для ОС Windows, обрабатывающую исключение типа "деление на ноль" для целых чисел
23. Написать на языке C программу для ОС Linux, которая создает два дочерних процесса и устанавливает их взаимодействие через `pipe`, устанавливая один файловый дескриптор `pipe` в качестве стандартного вывода первого процесса, а второй файловый дескриптор `pipe` в качестве стандартного ввода второго процесса, после чего ждет завершения дочерних процессов, первый дочерний процесс передает текстовый сообщение второму процессу и завершается, второй процесс выводит на экран полученное сообщение и завершается
24. Написать на языке C программу для ОС Windows, которая создает два дочерних процесса и устанавливает их взаимодействие через `pipe`, устанавливая один файловый дескриптор `pipe` в качестве стандартного вывода первого процесса, а второй файловый дескриптор `pipe` в качестве стандартного ввода второго процесса, после чего ждет завершения дочерних процессов, первый дочерний процесс передает текстовое сообщение второму процессу и завершается, второй процесс выводит на экран полученное сообщение и завершается
25. Написать на языке C программу с использованием потоков POSIX, реализующую монитор для доступа к очереди сообщений

Описание технологии проведения

Контрольная работа выполняется на компьютере и на проверку сдается исходный код.

Требования к выполнению заданий (или шкалы и критерии оценивания)

В контрольной работе задания 1-10 оцениваются в 1 бал, задания 11-20 – 2 балла, задания 21-25 – 4 балла (максимально возможная сумма при выполнении всех заданий – 50 баллов). При ошибках в выполнении задания или не полном выполнении оценка за задание снижается. Оценка за контрольную работу определяется как сумма баллов, набранных за все задания.

20.2 Промежуточная аттестация

Перечень вопросов к собеседованию

1. Понятие ядра операционной системы. Пользовательский и привилегированный режимы работы процессора.
2. Системные вызовы.
3. Понятие процесса. Основные состояния процесса. Легковесные процессы.

4. Принципы реализации многозадачности. Вытесняющая и не вытесняющая многозадачность.
5. Принципы управления ресурсами. Делимые и не делимые ресурсы. Стратегия и дисциплина управления. Приоритетное управление. Проблема инверсии приоритетов.
6. Справедливость и предсказуемость дисциплин управления ресурсами. Методы обеспечения гарантированной справедливости обслуживания.
7. Архитектура операционной системы. Влияние архитектуры на производительность, расширяемость, переносимость, надежность и уязвимость операционной системы.
8. Пользовательский интерфейс операционной системы. Классификация пользовательских интерфейсов. Интерфейсы, управляемые сообщениями. Оконная система X.
9. Основы информационной безопасности в операционных системах. Принципы реализации разграничения доступа.
10. Понятие прерывания. Общие сведения об обработке прерываний. Приоритеты прерываний. Маскирование прерываний. Внутренние и внешние прерывания. Командные (программные) прерывания.
11. Основы управления процессами. Уровни планирования. Планирование на верхнем уровне.
12. Планирование загрузки процессоров. Стратегии и дисциплины планирования загрузки процессоров.
13. Особенности планирования загрузки процессоров в многопроцессорных системах и системах реального времени.
14. Синхронизация выполнения процессов. Необходимость синхронизации. Общие вопросы синхронизации: гонки, взаимоисключающий доступ, тупики.
15. Локализация ссылок при обращении к памяти. Иерархическая модель памяти. Выборка, размещение и замещение данных.
16. Динамическое распределение памяти. Специальные алгоритмы динамического распределения памяти. Метод двойников (близнецов). Распределитель SLAB.
17. Виртуальная память. Принципы работы. Трансляция адресов.
18. Основные алгоритмы замещения в виртуальной памяти. Замещение случайных данных, FIFO, LRU, NRU, часовой алгоритм.
19. Рабочее множество страниц процесса. Резидентное множество страниц процесса. Управление резидентным множеством на основе концепции рабочего множества.
20. Системный дисковый кэш. Надежность операционной системы при использовании системного дискового кэша.
21. Процессорный кэш. Влияние процессорного кэша на операционную систему и другое ПО.
22. Загрузка программы в память. Способы загрузки. Основы реализации перемещающего загрузчика. Загрузка программ в системах с виртуальной памятью.
23. Компоновка (связывание) программных модулей. Основы реализации компоновщика.
24. Хранение программ в операционной системе. Основные форматы файлов исполняемых программ и библиотек.
25. Начальная загрузка операционных систем. Этапы загрузки (bootstrap-процедура).
26. Взаимодействие процессов. Передача данных между процессами.
27. Общие принципы организации и работы подсистемы ввода-вывода. Прямой доступ к памяти. Драйверы устройств.
28. Основы организации и работы файловых систем

Практические задания

1. В ОС Linux найти в каталоге /bin все файлы, размером более 10 кб
2. Написать на языке C программу для ОС Linux, блокирующую сигнал SIGINT
3. В ОС Linux найти в каталоге /bin все символические ссылки и вывести на экран их имена
4. В ОС Linux вывести подробные сведения о файлах домашнего каталога и всех его подкаталогов в файл list
5. В ОС Linux вывести на экран список всех запущенных процессов в системе
6. В ОС Linux вывести на экран список всех процессов, запущенных в текущем терминале
7. В ОС Linux вывести на экран таблицу процессов с сортировкой по нагрузке на процессор
8. В ОС Windows определить процесс, создающий наибольшую нагрузку на процессор

9. В ОС Windows создать файл, содержащий в двух потоках различные данные
10. Написать на языке C программу для ОС Windows, допускающую запуск только одного экземпляра программы

Описание технологии проведения

1. Собеседование производится в форме устного ответа на заданный вопрос. При необходимости преподаватель может задавать уточняющие вопросы. Ответ оценивается по 100 бальной шкале.
2. Практические задания выполняются на компьютере и на проверку сдается исходный код. Выполнение задания оценивается по 50 бальной шкале.

Требования к выполнению заданий, шкалы и критерии оценивания

Оценка знаний, умений и навыков, характеризующая этапы формирования компетенций в рамках изучения дисциплины, осуществляется в ходе текущей и промежуточной аттестаций.

При оценивании результатов промежуточной аттестации используется количественная шкала оценок.

Оценки за контрольную работу, практическое задание и собеседование складываются и результат нормируется к 100 бальной шкале. Полученное значение определяет уровень сформированности компетенций и итоговую оценку (достаточный – зачтено или недостаточный – не зачтено) согласно следующей шкале:

- оценка «зачтено» - 50..100 баллов
- оценка «не зачтено» - 0..49 баллов