

Минобрнауки России

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



УТВЕРЖДАЮ

Заведующий кафедрой
Борисов Дмитрий Николаевич
Кафедра информационных систем

10.04.2024

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ

Б1.О.19 Операционные системы

1. Код и наименование направления подготовки/специальности:

09.03.03 Прикладная информатика

2. Профиль подготовки/специализация:

Прикладная информатика в экономике

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:

рекомендована НМС ФКН 05.03.2024, протокол № 5

8. Учебный год:

2026-2027

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 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:

Дисциплина обязательной части (Б1.О).

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-3 Способен решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности | ОПК-3.1 Знает принципы, методы и средства решения стандартных задач профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности. | Знает основы внутреннего устройства и основные алгоритмы работы операционных систем, методы разграничения доступа, основы системного программирования и системного администрирования |
| ОПК-3 Способен решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности | ОПК-3.2 Умеет решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности. | Умеет выполнять конфигурирование операционной системы для решения задач профессиональной деятельности с учетом основных требований информационной безопасности |
| ОПК-3 Способен решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности | ОПК-3.3 Владеет навыками подготовки обзоров, аннотаций, составления рефератов, научных докладов, публикаций, и библиографии по научно-исследовательской работе с учетом требований информационной безопасности. | Имеет навыки использования возможностей операционной системы для решения задач профессиональной деятельности |

| | | |
|--|---|--|
| Код и название компетенции | Код и название индикатора компетенции | Знания, умения, навыки |
| ОПК-5 Способен устанавливать программное и аппаратное обеспечение для информационных и автоматизированных систем | ОПК-5.1 Знает основы системного администрирования, администрирования СУБД, современные стандарты информационного взаимодействия систем. | Знает основы внутреннего устройства и основные алгоритмы работы операционных систем, основы системного программирования и системного администрирования |
| ОПК-5 Способен устанавливать программное и аппаратное обеспечение для информационных и автоматизированных систем | ОПК-5.2 Умеет выполнять параметрическую настройку информационных и автоматизированных систем. | Умеет выполнять конфигурирование операционной системы для решения задач профессиональной деятельности |
| ОПК-5 Способен устанавливать программное и аппаратное обеспечение для информационных и автоматизированных систем | ОПК-5.3 Владеет навыками установки программного и аппаратного обеспечения информационных и автоматизированных систем. | Имеет навыки практической разработки сценариев командной оболочки и практического использования API операционных систем |

12. Объем дисциплины в зачетных единицах/час:

3/108

Форма промежуточной аттестации:

Экзамен

13. Трудоемкость по видам учебной работы

| Вид учебной работы | Семестр 5 | Всего |
|--------------------------|-----------|-------|
| Аудиторные занятия | 52 | 52 |
| Лекционные занятия | 36 | 36 |
| Практические занятия | | 0 |
| Лабораторные занятия | 16 | 16 |
| Самостоятельная работа | 20 | 20 |
| Вид учебной работы | Семестр 5 | Всего |
| Курсовая работа | | 0 |
| Промежуточная аттестация | 36 | 36 |
| Часы на контроль | 36 | 36 |
| Всего | 108 | 108 |

13.1. Содержание дисциплины

| п/п | Наименование раздела дисциплины | Содержание раздела дисциплины | Реализация раздела дисциплины с помощью онлайн-курса, ЭУМК |
|-----|--|--|---|
| 1.1 | 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 | https://edu.vsu.ru/course/view.php?id=11316 |
| 1.2 | 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 | https://edu.vsu.ru/course/view.php?id=11316 |
| 1.3 | Архитектура ОС | Архитектура ОС с монолитным ядром, модульным ядром, микроядром и экзоядром, гипервизор, влияние архитектуры на характеристики ОС | https://edu.vsu.ru/course/view.php?id=11316 |
| 1.4 | Пользовательский интерфейс ОС | Классификация интерфейсов, интерфейсы, управляемые сообщениями, основы реализации пользовательского интерфейса в GNU/Linux и MS Windows | https://edu.vsu.ru/course/view.php?id=11316 |

| | | | |
|-----|---------------------------|---|---|
| 1.5 | Обработка прерываний в ОС | Модель прерывания, разделение канала запроса прерывания несколькими устройствами, уровни обработки прерывания, поддержка прерываний в распространенных ОС | https://edu.vsu.ru/course/view.php?id=11316 |
| 1.6 | 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), поддержка IPC в распространенных ОС | https://edu.vsu.ru/course/view.php?id=11316 |
| 1.7 | Управление памятью | Локализация ссылок, иерархия памяти, динамическое распределение памяти, распределитель SLAB, виртуальная память, стратегии и дисциплины управления виртуальной памятью, основы работы с памятью в распространенных ОС | https://edu.vsu.ru/course/view.php?id=11316 |

| | | | |
|--------------------------------|---|---|---|
| 1.8 | Подсистема Ввода-Вывода | Взаимодействие процессора с внешними устройствами, драйвер устройства, сетевая подсистема, автоматическое определение подключенных устройств (P&P), основы реализации ввода-вывода и сети в распространенных ОС | https://edu.vsu.ru/course/view.php?id=11316 |
| 1.9 | 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, организация распространенных файловых систем | https://edu.vsu.ru/course/view.php?id=11316 |
| 1.10 | Основы информационной безопасности в ОС | Принципы разграничения доступа, дискреционная (избирательная) и мандатная модели, изолированная программная среда, принципы безопасного хранения паролей, реализация разграничения доступа в распространенных ОС | https://edu.vsu.ru/course/view.php?id=11316 |
| 2. Практические занятия | | | |
| | | | |
| 3. Лабораторные работы | | | |

| | | | |
|-----|---|--|---|
| 3.1 | Изучение основ интерфейса командной строки GNU/Linux и MS Windows (CMD, PowerShell) | Получение информации о текущей конфигурации системы, создание, удаление, копирование и перемещение файлов и каталогов, поиск файлов по различным критериям | https://edu.vsu.ru/course/view.php?id=11316 |
| 3.2 | Изучение основ работы в текстовом редакторе vim | Изучение основных команд vim: сохранение файла, редактирование файла, поиск фрагмента текста, удаление, копирование и вставка фрагмента текста. Конфигурирование vim (файл .vimrc) | https://edu.vsu.ru/course/view.php?id=11316 |
| 3.3 | Сценарии командной оболочки в GNU/Linux | Основы программирования на языке сценариев командной оболочки: переменные, условия, циклы, работа с файлами, вызов внешних утилит | https://edu.vsu.ru/course/view.php?id=11316 |

| | | | |
|-----|--|--|---|
| 3.4 | Основы работы с процессами в ОС GNU/Linux | Получение информации о процессах и загрузке системы (ps, top), виртуальная файловая система rproc, создание нового процесса (fork, clone), замена программы процесса (exec), сигналы, приостановка и возобновление работы процесса (SIGSTOP, SIGCONT), управление приоритетами, завершение процесса (exit), принудительное завершение процесса (SIGKILL), ожидание завершения дочерних процессов (wait), синхронизация процессов | https://edu.vsu.ru/course/view.php?id=11316 |
| 3.5 | Основы работы с процессами ОС MS Windows | Диспетчер задач (Taskmgr.exe), утилита Process Monitor, команда TASKLIST, команда TASKKILL, создание и завершение процесса (CreateProcess, ExitProcess), создание потока (CreateThread), управление приоритетами, синхронизация процессов | https://edu.vsu.ru/course/view.php?id=11316 |
| 3.6 | Диагностика компьютера и анализ производительности | Диагностические сообщения Linux, диспетчер устройств Windows, системный журнал, виртуальная файловая система /sys в Linux, утилиты tcpdump и iperf, монитор ресурсов в Windows, утилита top в Linux | https://edu.vsu.ru/course/view.php?id=11316 |

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 | 2 | | | 2 | 3 |
| 2 | Основные понятия и определения предметной области ОС | 2 | | | 2 | 5 |
| 3 | Архитектура ОС | 2 | | | 2 | 5 |
| 4 | Пользовательский интерфейс ОС | 2 | | | 2 | 3 |
| 5 | Обработка прерываний в ОС | 2 | | | 2 | 5 |
| 6 | Управление процессами | 8 | | | 2 | 16 |
| 7 | Управление памятью | 8 | | | 2 | 13 |
| 8 | Подсистема ввода-вывода | 4 | | | 2 | 10 |
| 9 | Файловая подсистема | 2 | | | 2 | 7 |
| 10 | Основы информационной безопасности в ОС | 4 | | | 2 | 5 |
| 11 | Изучение основ интерфейса командной строки GNU/Linux и MS Windows (CMD, PowerShell) | | | 2 | | 4 |
| 12 | Изучение основ работы в текстовом редакторе vim | | | 1 | | 3 |
| 13 | Сценарии командной оболочки в GNU/Linux | | | 3 | | 7 |
| 14 | Основы работы с процессами в ОС GNU/Linux | | | 6 | | 12 |
| 15 | Основы работы с процессами ОС MS Windows | | | 3 | | 7 |
| 16 | Диагностика компьютера и анализ производительности | | | 1 | | 3 |
| | | 36 | 0 | 16 | 20 | 72 |

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

а) основная литература:

| № 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 |
| 2 | Савинков А. Ю. Основы управления процессами в операционных системах : учебное пособие / А. Ю. Савинков ; Министерство науки и высшего образования РФ, Федеральное государственное бюджетное образовательное учреждение высшего образования "Воронежский государственный университет". - Воронеж : Издат. дом ВГУ, 2022. - 184 с. : ил., табл.; 20 см.; ISBN 978-5-9273-3532-9 |

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 |
| 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):

Лекции-визуализации 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 и компьютерами с операционной системой GNU/Linux.

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 | Введение в ОС Основные понятия и определения предметной области ОС Основы информационной безопасности в ОС | ОПК-3 | ОПК-3.1 | Собеседование |
| 2 | Изучение основ интерфейса командной строки GNU/Linux и MS Windows (CMD, PowerShell) Изучение основ работы в текстовом редакторе vim | ОПК-3 | ОПК-3.2 | Лабораторные работы |
| 3 | Диагностика компьютера и анализ производительности | ОПК-3 | ОПК-3.3 | Лабораторные работы |
| 4 | Архитектура ОС Пользовательский интерфейс ОС Обработка прерываний в ОС Управление процессами Управление памятью Подсистема Ввода-Вывода Файловая система | ОПК-5 | ОПК-5.1 | Собеседование |
| 5 | Сценарии командной оболочки в GNU/Linux | ОПК-5 | ОПК-5.2 | Лабораторные работы |
| 6 | Основы работы с процессами в ОС GNU/Linux Основы работы с процессами ОС MS Windows | ОПК-5 | ОПК-5.3 | Лабораторные работы |

Промежуточная аттестация

Форма контроля - Экзамен

Оценочные средства для промежуточной аттестации

Собеседование

20 Типовые оценочные средства и методические материалы, определяющие процедуры оценивания

20.1 Текущий контроль успеваемости

Текущий контроль успеваемости выполняется по лабораторным работам.

По каждой выполненной работе должен быть предоставлен отчет, включающий исходный код разработанных программ и описание полученных результатов. По отчету преподаватель вправе задать дополнительные вопросы для уточнения уровня понимания материала. Лабораторная работа оценивается максимум в 100 баллов.

Приведённые ниже задания рекомендуется использовать при проведении диагностических работ для оценки остаточных знаний по дисциплине

Компетенция ОПК-3

Задания закрытого типа

- 1) Выберите правильные утверждения относительно блочного устройства UNIX и UNIX-подобных операционных систем
 - a) Обмен данными с устройством выполняется только блоками фиксированного размера
 - b) Блоки данных на устройстве являются произвольно адресуемыми
 - c) Обмен данными с устройством выполняется неформатированным потоком байт
 - d) Данные на устройстве не являются адресуемыми и могут считываться или записываться только последовательно от начала до конца
- 2) GCC это
 - a) Компилятор для языка C
 - b) Набор компиляторов (GNU Compiler Collection) для различных языков программирования
 - c) Компоновщик программных модулей
- 3) Семафор это объект операционной системы, позволяющий
 - a) Реализовать ожидание процессами момента наступления запланированных событий
 - b) Исключить взаимную блокировку процессов
 - c) Обеспечить соблюдение приоритетов
 - d) Прервать выполнение процесса при ошибке
- 4) Критическая секция программного кода это
 - a) Участок кода программы, из которого выполняются действия с критическим ресурсом (ресурс, не допускающий одновременного использования несколькими процессами)
 - b) Участок кода программы, оказывающий критическое влияние на производительность
 - c) Участок кода программы, содержащий ошибку
- 5) Взаимная блокировка процессов это
 - a) Ситуация, когда два или более процессов заблокированы в ожидании действий друг от друга
 - b) Попытка одновременного доступа двух или более процессов к критическому ресурсу
 - c) Невозможность завершить процессы до запланированного срока (deadline) в операционной системе реального времени из-за недостатка производительности
- 6) Какие из необходимых условия взаимной блокировки (условия Коффмана, Элфика и Шошани) могут быть нарушены (исключены) в реальной системе для исключения взаимной блокировки процессов
 - a) Условие взаимного исключения (только один процесс может получить доступ к критическому ресурсу)

- b) Условие ожидания ресурсов (ожидая недостающие ресурсы, процессы удерживают за собой ранее полученные ресурсы)
- c) Условие неперераспределяемости ресурсов (ресурсы, выделенные процессу, не могут быть перераспределены)
- d) Условие кругового ожидания (существует кольцевая цепочка процессов, каждый из которых удерживает ресурсы, необходимые следующему процессу в цепочке)
- 7) Индексный узел (inode) в файловых системах, используемых в GNU/Linux, содержит
- Имя файла
 - Маску разрешенных видов доступа
 - Информацию о времени последнего обращения к файлу
 - Идентификатор владельца файла
 - Счетчик ссылок на файл
- 8) С помощью какого из перечисленных далее системных вызов может быть установлена политика планирования для процесса в операционной системе GNU/Linux
- sched_setaffinity()
 - sched_setscheduler()
 - sched_setparam()
 - sched_yield()
- 9) Выберите правильные утверждения относительно дискового кэша операционной системы
- Дисковый кэш сокращает среднее время доступа к данным файлов
 - Дисковый кэш снижает надежность хранения данных
 - Дисковый кэш повышает надежность хранения данных
 - Дисковый кэш увеличивает максимальный размер файлов
- 10) С помощью какой из перечисленных далее утилит может быть получен список запущенных процессов в операционной системе GNU/Linux
- ps
 - ls
 - ss
 - rm

Ответы на вопросы

| Номер вопроса | Ответ (буква) |
|---------------|---------------|
| 1 | a, b |
| 2 | b |
| 3 | a |
| 4 | a |
| 5 | a |
| 6 | b, c, d |
| 7 | b, c, d, e |

| | |
|----|------|
| 8 | b |
| 9 | a, b |
| 10 | a |

Задания открытого типа

- 1) В виртуальном адресе поле смещение имеет размер 12 бит. Какой может быть максимальный размер страницы в виртуальном адресном пространстве. Ответ выразите в кибибайтах (KiB)
- 2) Чему равно численное значение файлового дескриптора потока стандартного вывода (STDOUT) в операционных системах GNU/Linux
- 3) Укажите числовое значение параметра предпочтения (nice), которое требуется установить для получения максимального приоритета процесса с политикой планирования SCHED_NORMAL

Ответы на вопросы

| Номер вопроса | Ответ |
|---------------|-------|
| 1 | 4 |
| 2 | 1 |
| | -20 |

Задания с развёрнутым ответом

- 1) Дайте определение стека и опишите его типичное использование
- 2) Проблема инверсии приоритетов и ее решение
- 3) Что такое прерывание

Ответы на вопросы

| Номер вопроса | Ответ (буква) |
|---------------|---|
| 1 | Стек – это контейнер данных, работающий по принципу LIFO (последним вошел – первым вышел). В современных компьютерах стек используется для хранения адресов возврата из подпрограмм и прерываний, для передачи аргументов в функции, для хранения промежуточных результатов вычислений и временных (автоматических) переменных |

| Критерии оценивания | Шкала оценок (в баллах) |
|--|-------------------------|
| Обучающийся дал определение стека и описал его типичное использование. | 3 балла |
| Обучающийся дал правильное определение стека и описал его типичное использование. Ответ содержит незначительные неточности. | 2 балла |
| Обучающийся не дал точное определение стека и привел неполное описание его типичного использования. Ответ не содержит грубых ошибок и неточностей. | 1 балл |
| Обучающийся не дал точное определение стека и не описал его типичное использование. Ответ содержит грубые ошибки и неточности. | 0 баллов |

| Номер вопроса | Ответ (буква) |
|---------------|--|
| 2 | <p>Пусть в системе существует три процесса с низким, средним и высоким приоритетом. Предположим, что низкоприоритетный процесс получил доступ к некоторому критическому ресурсу и вошел в критическую секцию своего кода. Предположим, что высокоприоритетному процессу понадобился критический ресурс, удерживаемый низкоприоритетным процессом. Но поскольку ресурс занят, высокоприоритетный процесс блокируется в ожидании освобождения ресурса. Процессорное время после блокировки высокоприоритетного процесса в основном будет потреблять процесс среднего приоритета. Низкоприоритетный процесс получает мало процессорного времени и его программа выполняется медленно, вследствие чего удерживаемый им критический ресурс долго не освобождается. Получается, что выполнение высокоприоритетного процесса лимитируется скоростью выполнения низкоприоритетного процесса. Описанная проблема известна под названием инверсия приоритета.</p> <p>Проблему инверсии приоритета можно решить за счет наследования приоритетов. При блокировке высокоприоритетного процесса на некотором ресурсе, его приоритет временно передается процессу, удерживающему этот ресурс. Процесс, удерживающий ресурс, теперь будет выполняться быстрее и быстрее освободит ресурс, вследствие чего высокоприоритетный процесс быстрее получит доступ к ресурсу. Например, описанный механизм наследования приоритетов реализован в мьютексах Linux и FreeRTOS.</p> |

| Критерии оценивания | Шкала оценок (в баллах) |
|--|-------------------------|
| Обучающийся точно и однозначно описал проблему инверсии приоритетов, описал метод решения проблемы на основе наследования приоритетов, привел примеры. | 3 балла |
| Обучающийся точно и однозначно описал проблему инверсии приоритетов, описал метод решения проблемы на основе наследования приоритетов, привел примеры. Ответ содержит незначительные неточности. | 2 балла |
| Обучающийся недостаточно точно описал проблему инверсии приоритетов, не полно описал метод решения проблемы, не привел примеры. Ответ не содержит грубых ошибок и неточностей. | 1 балл |
| Обучающийся не точно описал проблему инверсии приоритетов, не полно описал метод решения проблемы, не привел примеры. Ответ содержит грубые ошибки и неточности. | 0 баллов |

| Номер вопроса | Ответ (буква) |
|---------------|--|
| 3 | <p>Механизм прерываний – это один из базовых аппаратных механизмов, реализуемых любым процессором.</p> <p>Механизм прерываний при наступлении некоторого заранее ожидаемого события передает управление соответствующей заранее подготовленной подпрограмме – обработчику прерывания. При этом все возможные события</p> |

| | |
|--|---|
| | <p>заранее определены, но моменты их возникновения априорно неизвестны. Адреса обработчиков прерываний (векторы прерываний) на этапе инициализации системы размещают в специальном массиве – таблице векторов прерываний.</p> <p>Прерывания можно классифицировать как внешние (запрашиваются внешними устройствами) и внутренние, возникающие в самом процессоре при различных ошибках (исключения) или по инициативе выполняющейся программы (программное прерывание).</p> <p>Без поддержки механизма прерываний невозможно реализовать вытесняющую многозадачность, виртуальную память, системные вызовы, точки останова при отладке программ и эффективную работу с периферийными устройствами.</p> |
|--|---|

| Критерии оценивания | Шкала оценок (в баллах) |
|--|-------------------------|
| Обучающийся дал правильное определение прерывания, пояснил роль прерываний в системе и принципы их обработки. | 3 балла |
| Обучающийся дал правильное определение прерывания, пояснил роль прерываний в системе и принципы их обработки. Ответ содержит незначительные неточности. | 2 балла |
| Обучающийся дал недостаточно точное определение прерывания, недостаточно пояснил роль прерываний в системе и принципы их обработки. Ответ содержит незначительные неточности. Ответ не содержит грубых ошибок и неточностей. | 1 балл |
| Обучающийся дал недостаточно точное определение прерывания, недостаточно пояснил роль прерываний в системе и принципы их обработки. Ответ содержит грубые ошибки и неточности. | 0 баллов |

Компетенция ОПК-5

Задания закрытого типа

- 1) Какие функции реализует в операционной системе подсистема управления процессами
 - a) Планирование загрузки процессоров
 - b) Создание и удаление процессов
 - c) Выделение памяти
 - d) Планирование выполнения запросов ввода-вывода
 - e) Синхронизация процессов
- 2) Какие функции выполняет в операционной системе подсистема ввода-вывода
 - a) Выделение памяти
 - b) Изоляция программ от особенностей аппаратной реализации устройств ввода-вывода
 - c) Планирование загрузки процессоров
 - d) Планирование выполнения запросов ввода-вывода
 - e) Организация бесконфликтной работы устройств ввода-вывода в многозадачных системах
- 3) Механизм прямого доступа к памяти (DMA) позволяет

- a) Периферийным устройствам читать и записывать данные в памяти без участия центрального процессора
 - b) Центральному процессору напрямую обращаться к физической памяти минуя систему трансляции адресов виртуальной памяти
- 4) Что такое точка входа в программу (entry point)
- a) Адрес загрузки программы в память
 - b) Имя файла программы
 - c) Адрес первой инструкции программы, с которой должно начаться выполнение программы после загрузки в память
 - d) Адрес вершины стека
- 5) При использовании вытесняющей многозадачности
- a) Время переключения процессов определяет текущая пользовательская программа после чего планировщик операционной системы определяет следующий процесс
 - b) Время переключения процессов и следующий процесс определяет планировщик операционной системы
- 6) На каком языке программирования написано ядро Linux
- a) Преимущественно на C
 - b) Весь код написан на ассемблере
 - c) На Python
 - d) На Pascal
- 7) При использовании корпоративной многозадачности
- a) Время переключения процессов определяет текущая пользовательская программа после чего планировщик операционной системы определяет следующий процесс
 - b) Время переключения процессов и следующий процесс определяет планировщик операционной системы
- 8) Зачем нужна операционная система
- a) Для изоляции задач и взаимодействие между задачами
 - b) Для повышения производительности вычислений
 - c) Для реализации многозадачности и распределения ресурсов между задачами
 - d) Для изоляции программ от деталей реализации аппаратной платформы
 - e) Для инициализации аппаратной платформы
- 9) Отметьте основные состояния процессов в системе планирования загрузки процессоров
- a) Загрузка
 - b) Готовность
 - c) Выполнение
 - d) Переключение
 - e) Блокировка (или ожидание)
- 10) Зачем нужна многозадачность

- a) Позволяет сократить простой процессора повысить коэффициент использования процессора
- b) Позволяет одновременно решать несколько задач на одном компьютере
- c) Повышает производительность компьютера
- d) Снижает энергопотребление компьютера

Ответы на вопросы

| Номер вопроса | Ответ (буква) |
|---------------|---------------|
| 1 | a,b,e |
| 2 | b,d,e |
| 3 | a |
| 4 | c |
| 5 | b |
| 6 | a |
| 7 | a |
| 8 | a,c,d,e |
| 9 | b,c,e |
| 10 | a,b |

Задания открытого типа

- 1) Сколько различных символов может быть представлено в кодировке ASCII(7)
- 2) В 32-битном режиме в адресном пространстве процесса для нужд ядра зарезервирован сегмент размером 1 ГБ, какой объем виртуального адресного пространства (ГБ) остается доступен приложению

Ответы на вопросы

| Номер вопроса | Ответ |
|---------------|-------|
| 1 | 128 |
| 2 | 3 |

Задания с развёрнутым ответом

- 1) Какие основные файловые операции должны быть реализованы файловой системой операционной системы
- 2) Дайте определения процесса и легковесного процесса
- 3) Что такое системный вызов

Ответы на вопросы

| Номер вопроса | Ответ (буква) |
|---------------|--|
| 1 | Создание – новый пустой файл создается в файловой системе Удаление – файл исключается из файловой системы и занимаемые файлом ресурсы освобождаются Копирование – в файловой системе создается новый файл, содержимое которого дублирует некоторый существующий файл Перемещение – изменяется расположение файла в файловой системе |

| | |
|--|---|
| | <p>Чтение – данные файла копируются в память</p> <p>Запись – новые данные из памяти заменяют данные файла</p> <p>Дополнение – новые данные из памяти дописываются в конец файла</p> |
|--|---|

| Критерии оценивания | Шкала оценок (в баллах) |
|---|-------------------------|
| Обучающийся правильно перечислил все операции | 3 балла |
| Обучающийся правильно перечислил все операции. Допускаются незначительные неточности | 2 балла |
| Обучающийся перечислил только некоторые операции. Ответ не содержит грубых ошибок и неточностей | 1 балл |
| Обучающийся перечислил только некоторые операции. Ответ содержит грубых ошибок и неточности | 0 баллов |

| Номер вопроса | Ответ (буква) |
|---------------|---|
| 2 | <p>Процесс – это абстрактный объект ядра операционной системы, представляющий исполняющуюся программу.</p> <p>Легковесный процесс (LWP) – это процесс, который не имеет собственного виртуального адресного пространства. Несколько легковесных процессов совместно используют общее виртуальное адресное пространство.</p> |

| Критерии оценивания | Шкала оценок (в баллах) |
|---|-------------------------|
| Обучающийся правильно дал определение процесса и легковесного процесса. | 3 балла |
| Обучающийся в целом правильно дал определение процесса и легковесного процесса. Допускаются незначительные неточности | 2 балла |
| Обучающийся не точно дал определение процесса и легковесного процесса. Ответ не содержит грубых ошибок и неточностей | 1 балл |
| Обучающийся не точно дал определение процесса и легковесного процесса. Ответ содержит грубых ошибок и неточности | 0 баллов |

| Номер вопроса | Ответ (буква) |
|---------------|---|
| 3 | <p>Системный вызов – это обращение прикладной программы к ядру операционной системы для выполнения какого-либо действия (создание процесса, обмен данными с другими процессами, запись файла и т.п.). Системный вызов принципиально отличается от простого вызова функции. Для выполнения системного вызова процессор переходит в режим ядра и управление передается из пользовательской программы в ядро операционной системы.</p> <p>Обычно для реализации системных вызовов используется командные прерывания или специальная инструкция процессора, которая фактически является оптимизированной версией командного прерывания.</p> <p>Каждая операционная система предоставляет прикладным программам собственный набор системных вызовов, вместе составляющих интерфейс прикладного программирования (API) операционной системы</p> |

| Критерии оценивания | Шкала оценок (в баллах) |
|--|-------------------------|
| Обучающийся дал определение системного вызова, пояснил отличие системного вызова от простого вызова функции. | 3 балла |
| Обучающийся дал определение системного вызова, пояснил отличие системного вызова от простого вызова функции. Ответ содержит незначительные неточности | 2 балла |
| Обучающийся не дал точное определение системного вызова, не пояснил отличие системного вызова от простого вызова функции. Ответ не содержит грубых ошибок и неточностей. | 1 балл |
| Обучающийся не дал точное определение системного вызова, не пояснил отличие системного вызова от простого вызова функции. Ответ содержит грубые ошибки и неточности. | 0 баллов |

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

Описание технологии проведения

Собеседование производится в форме устного ответа на заданный вопрос. При необходимости преподаватель может задавать уточняющие вопросы. Ответ оценивается по 100 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 лабораторные работы складываются с оценкой, полученной на собеседовании, и результат нормируется к 100 бальной шкале. Полученное значение определяет уровень сформированности компетенций и итоговую оценку (достаточный – удовлетворительно, хорошо, отлично или недостаточный – неудовлетворительно) согласно следующей шкале:

- оценка «отлично» – 90...100 баллов
- оценка «хорошо» — 70...89 баллов
- оценка «удовлетворительно» – 50...69 баллов
- оценка «неудовлетворительно» – 0...49 баллов