

Минобрнауки России

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



**УТВЕРЖДАЮ**

Заведующий кафедрой  
Борисов Дмитрий Николаевич  
Кафедра информационных систем

05.05.2025

**РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ**

Б1.В.05 Программирование и управление системами реального времени

**1. Код и наименование направления подготовки/специальности:**

09.04.02 Информационные системы и технологии

**2. Профиль подготовки/специализация:**

**3. Квалификация (степень) выпускника:**

Магистратура

**4. Форма обучения:**

Заочная

**5. Кафедра, отвечающая за реализацию дисциплины:**

Кафедра информационных систем

**6. Составители программы:**

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

**7. Рекомендована:**

рекомендована НМС ФКН 05.05.2025, протокол № 7

**8. Учебный год:**

2026-2027

**9. Цели и задачи учебной дисциплины:**

Цель: формирование необходимых компетенций в предметной области программирования и управления задачами в системах реального времени на основе ядра Linux

Задачи учебной дисциплины:

- рассмотреть общие методы и базовые алгоритмы планирования работы потоков выполнения в операционных системах
- рассмотреть общие методы и базовые алгоритмы синхронизации потоков выполнения и межпроцессного взаимодействия в операционных системах
- рассмотреть особенности планирования и синхронизации потоков управления в ядре Linux
- рассмотреть API ядра Linux для работы с потоками управления
- рассмотреть методы повышения эффективности работы задач реального времени в операционных системах на основе ядра Linux

**10. Место учебной дисциплины в структуре ООП:**

Дисциплина относится к части, формируемой участниками образовательных отношений.

**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
ПК-3 Способен проектировать структуру программ и реализует алгоритмы с учётом требований к надёжности, модульности и масштабируемости	ПК-3.2 Использует современные языки программирования для реализации алгоритмов	Знает API и основные алгоритмы GNU/Linux в части управления потоками выполнения, включая синхронизацию и обмен данными. Умеет использовать типовые программные инструменты для редактирования исходного кода, сборки и отладки программ в GNU/Linux Имеет практические навыки 1) использования GCC, GDB и VSCode при разработке и отладке программ для GNU/Linux 2) использования API ядра Linux для синхронизации и обмена данными в задачах реального времени
ПК-4 Способен обеспечивать проверку программного обеспечения на различных этапах разработки и внедрения	ПК-4.2 Проводит тестирование ПО по разработанным тестовым случаям	Знает основные инструменты GNU/Linux для отладки трассировки программ, включая программы реального времени Умеет выполнять отладку и верификацию программ в GNU/Linux Имеет практические навыки отладки и верификации программ в GNU/Linux

**12. Объем дисциплины в зачетных единицах/час:**

4/144

**Форма промежуточной аттестации:**

Экзамен

**13. Трудоемкость по видам учебной работы**

Вид учебной работы	Курс 2		Всего
	Триместр 4	Триместр 5	
Аудиторные занятия	14	6	20
Лекционные занятия	6	2	8
Практические занятия	-	-	-
Лабораторные занятия	8	4	12
Самостоятельная работа	58	57	115
Курсовая работа	-	-	-
Промежуточная аттестация	-	-	-
Часы на контроль	-	9	9
Всего	72	72	144

### 13.1. Содержание дисциплины

п/п	Наименование раздела дисциплины	Содержание раздела дисциплины	Реализация раздела дисциплины с помощью онлайн-курса, ЭУМК
<b>1. Лекции</b>			
1.1	Основы 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)	<a href="https://edu.vsu.ru/course/view.php?id=31758">https://edu.vsu.ru/course/view.php?id=31758</a>
1.2	Основы управление памятью	Локализация ссылок, иерархия памяти, динамическое распределение памяти, распределитель SLAB, виртуальная память, стратегии и дисциплины управления виртуальной памятью	<a href="https://edu.vsu.ru/course/view.php?id=31758">https://edu.vsu.ru/course/view.php?id=31758</a>
1.3	Детали управления процессами и памятью в ядре Linux, в том числе в задачах реального времени	Политики планирования реального времени, POSIX.1b Realtime Extension, резервирование и прогрев ресурсов для задач реального времени, инструменты и утилиты для отладки и мониторинга задач реального времени	<a href="https://edu.vsu.ru/course/view.php?id=31758">https://edu.vsu.ru/course/view.php?id=31758</a>
<b>2. Практические занятия</b>			
<b>3. Лабораторные работы</b>			
3.1	Сборка и отладка программ с использованием GCC/GDB/VSCode	Изучение базовых программных инструментов для редактирования исходного кода, сборки и отладки программ в GNU/Linux	<a href="https://edu.vsu.ru/course/view.php?id=31758">https://edu.vsu.ru/course/view.php?id=31758</a>
3.2	Ограничение (квоты) на потребление системных ресурсов в Linux	Ограничение относительной доли процессорного времени и объема памяти,	<a href="https://edu.vsu.ru/course/view.php?id=31758">https://edu.vsu.ru/course/view.php?id=31758</a>

п/п	Наименование раздела дисциплины	Содержание раздела дисциплины	Реализация раздела дисциплины с помощью онлайн-курса, ЭУМК
		доступных группе потоков управления (группе задач), реакция на превышение квоты использования памяти	
3.3	Синхронизация потоков в Linux (spin-блокировки, мьютексы, семафоры, условные переменные, барьеры синхронизации)	Реализация объекта-монитора для потоково-безопасного доступа к данным, использование неблокирующих методов обновления данных (модель Read-Copy-Update)	<a href="https://edu.vsu.ru/course/view.php?id=31758">https://edu.vsu.ru/course/view.php?id=31758</a>
3.4	Межзадачное взаимодействие в Linux (сигналы, сигналы реального времени, очереди, разделяемая память)	Обработка сигнала с использованием файла сигнала (signalfd), прямое чтение сигналов из очереди, работа с сигналами в многопоточных приложениях (маскирование сигналов, выбор потока управления для обработки сигнала), передача данных между задачами через очереди и разделяемую память.	<a href="https://edu.vsu.ru/course/view.php?id=31758">https://edu.vsu.ru/course/view.php?id=31758</a>
3.5	Управление приоритетами и политиками планирования в Linux, политики планирования реального времени SCHED_RR и SCHED_FIFO	Назначение потоку управления политики планирования, управление приоритетами, особенности (различия) политик SCHED_RR и SCHED_FIFO, квота процессорного времени для политик реального времени	<a href="https://edu.vsu.ru/course/view.php?id=31758">https://edu.vsu.ru/course/view.php?id=31758</a>
3.6	Планирование задач реального времени по сроку завершения (политика SCHED_DEADLINE)	Планирование задачи реального времени по сроку завершения, назначение политики планирования SCHED_DEADLINE потоку выполнения, применение политики SCHED_DEADLINE для периодических и спорадических задач, работа политики SCHED_DEADLINE при превышении задачей заявленного предела потребления	<a href="https://edu.vsu.ru/course/view.php?id=31758">https://edu.vsu.ru/course/view.php?id=31758</a>

п/п	Наименование раздела дисциплины	Содержание раздела дисциплины	Реализация раздела дисциплины с помощью онлайн-курса, ЭУМК
		процессорного времени	
3.7	Резервирование процессоров и прогрев памяти для задачи реального времени в Linux	Резервирование процессорных ядер, резервирование памяти в куче и в стеке, прогрев кучи и стека, трассировка задач реального времени	<a href="https://edu.vsu.ru/course/view.php?id=31758">https://edu.vsu.ru/course/view.php?id=31758</a>

### 13.2. Темы (разделы) дисциплины и виды занятий

№ п/п	Наименование темы (раздела)	Лекционные занятия	Практические занятия	Лабораторные занятия	Самостоятельная работа	Всего
1	Основы управление процессами	3	-	-	12	15
2	Основы управление памятью	3	-	-	12	15
3	Детали управления процессами и памятью в ядре Linux, в том числе в задачах реального времени	2	-	-	24	26
4	Сборка и отладка программ с использованием GCC/GDB/VSCode	-	-	2	10	12
5	Ограничение (квоты) на потребление системных ресурсов в Linux	-	-	2	8	10
6	Синхронизация потоков в Linux (spin-блокировки, мьютексы, семафоры, условные переменные, барьеры синхронизации)	-	-	2	8	10
7	Межзадачное взаимодействие в Linux (сигналы, сигналы реального времени, очереди, разделяемая память)	-	-	2	8	10
8	Управление приоритетами и политиками планирования в Linux, политики планирования реального времени SCHED_RR и SCHED_FIFO	-	-	1	12	13

№ п/п	Наименование темы (раздела)	Лекционные занятия	Практические занятия	Лабораторные занятия	Самостоятельная работа	Всего
9	Планирование задач реального времени по сроку завершения (политика SCHED_DEADLINE)	-	-	1	12	13
10	Резервирование процессоров и прогрев памяти для задачи реального времени в Linux	-	-	2	9	11
		8	-	12	115	135

#### 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	Керрис М. Linux API. Исчерпывающее руководство / М. Керрис. — СПб.: Питер, 2025. — 1248 с.: ил. — (Серия «Библиотека программиста») — ISBN 978-5-4461-0985-2.
2	Таненбаум Э. Современные операционные системы. 4-е изд. / Таненбаум Э., Бос Х.— СПб.: Питер, 2025. — 1120 с.: ил. — (Серия «Классика computer science»). — ISBN 978-5-4461-1155-8.

##### б) дополнительная литература:

№ п/п	Источник
1	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
2	Савинков А. Ю. Основы управления памятью 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, 2024. — 172 с.: ил., табл.; 20 см.; ISBN 978-5-9273-4097-2

##### в) информационные электронно-образовательные ресурсы:

№ п/п	Источник
1	Библиотека ВГУ, <a href="http://www.lib.vsu.ru">http://www.lib.vsu.ru</a>
2	Сервер учебно-методических материалов ФКН, \\fs.cs.vsu.ru/Library

3	Образовательный портал "Электронный университет ВГУ", <a href="http://edu.vsu.ru">http://edu.vsu.ru</a>
---	---

**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", <a href="http://edu.vsu.ru">http://edu.vsu.ru</a>

**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](http://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 GNU/Linux с установленными инструментами пакетами программ GCC, GDB, VSCode.

**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	1. Основы управление процессами 2. Основы управление памятью 3. Детали управления процессами и памятью в ядре Linux, в том числе в задачах реального времени 4. Сборка и отладка программ с использованием GCC/GDB/VSCode 5. Ограничение (квоты) на потребление системных ресурсов в Linux 6. Синхронизация потоков в Linux (spin-блокировки, мьютексы, семафоры, условные переменные, барьеры синхронизации) 7. Межзадачное взаимодействие в Linux (сигналы, сигналы реального времени, очереди, разделяемая память)	ПК-3	ПК-3.2	Лабораторные работы
2	6. Синхронизация потоков в Linux (spin-блокировки, мьютексы, семафоры, условные переменные, барьеры синхронизации) 7. Межзадачное взаимодействие в Linux (сигналы, сигналы реального времени, очереди, разделяемая память) 8. Управление приоритетами и политиками планирования в Linux,	ПК-4	ПК-4.2	Лабораторные работы

№ п/п	Разделы дисциплины (модули)	Код компетенции	Код индикатора	Оценочные средства для текущей аттестации
	политики планирования реального времени SCHED_RR и SCHED_FIFO 9. Планирование задач реального времени по сроку завершения (политика SCHED_DEADLINE) 10. Резервирование процессоров и прогрев памяти для задачи реального времени в Linux			

Промежуточная аттестация

Форма контроля – Экзамен, курсовая работа

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.

#### Темы лабораторных работ для 1-й текущей аттестации

1. Сборка и отладка программ с использованием GCC/GDB/VSCode
2. Ограничение (квоты) на потребление системных ресурсов в Linux
3. Синхронизация потоков в Linux (spin-блокировки, мьютексы, семафоры, условные переменные, барьеры синхронизации)
4. Межзадачное взаимодействие в Linux (сигналы, сигналы реального времени, очереди, разделяемая память)

#### Темы лабораторных работ для 2-й текущей аттестации

1. Управление приоритетами и политиками планирования в Linux, политики планирования реального времени SCHED\_RR и SCHED\_FIFO
2. Планирование задач реального времени по сроку завершения (политика SCHED\_DEADLINE)
3. Резервирование процессоров и прогрев памяти для задачи реального времени в Linux

По каждой выполненной лабораторной 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 ПК-3**

#### Задания закрытого типа

- 1) Какие языки программирования поддерживают инструменты GCC (выберите наиболее полное соответствие)
  - a) C, C++
  - b) C, C++, Fortran, Ada
  - c) C, C++, Fortran, Ada, Go, D, Modula-2
  - d) C, C++, Fortran, Ada, Go, D, Modula-2, Pascal, Python
- 2) Что такое watchpoint
  - a) Условная точка останова программы

- b) Механизм остановки выполнения программы с передачей управления отладчику, активизируемый в указанном диапазоне времени
- c) То же самое, что и breakpoint
- d) Механизм остановки выполнения программы с передачей управления отладчику при изменении данных по указанному адресу в памяти
- 3) Что такое условная точка останова
- a) Точка останова (breakpoint), активизирующаяся только при выполнении дополнительного условия
- b) Механизм защиты от переполнения стека
- c) Механизм остановки выполнения программы с передачей управления отладчику при изменении данных по указанному адресу в памяти
- d) Последняя инструкция программы
- 4) Что такое условная переменная
- a) Переменная в программе, с которой связана watchpoint
- b) Переменная в программе, задействованная в условии условной точки останова
- c) Переменная в программе, определяющая условие завершения цикла
- d) Механизм синхронизации, обеспечивающий блокирование потока выполнения до выполнения заданного условия
- 5) Что такое наследование приоритета в системе синхронизации потоков выполнения
- a) Передача дочернему потоку приоритета родительского потока после завершения программы родительского потока
- b) Временное повышение приоритета потока-владельца мьютекса до максимального приоритета из приоритетов потоков выполнения, заблокированных на этом мьютексе
- c) Сохранение неизменной суммы числовых значений приоритетов всех потоков выполнения, имеющих общий родительский поток
- d) Временное повышение приоритета потока при переходе его родительского потока в состояние ожидания

Ответы на вопросы

Номер вопроса	Ответ (буква)
1	c
2	d
3	a
4	d
5	b

Задания открытого типа

- 1) Какой приоритет в системе планирования реального времени имеют задачи с политикой планирования SCHED\_NORMAL
- 2) Какой минимальный уровень приоритета может быть установлен для задачи с политикой планирования SCHED\_RR

Ответы на вопросы

Номер вопроса	Ответ
1	0
2	1

### Задания с развёрнутым ответом

1) Поясните сущность проблемы инверсии приоритета и общий подход к ее решению.

Ответы на вопросы

Номер вопроса	Ответ (буква)
1	<p>Сущность проблемы инверсии приоритета состоит в появлении зависимости выполнения высокоприоритетного процесса от работы процессов более низких приоритетов. Например, низкоприоритетный процесс захватил мьютекс и после этого был вытеснен с выполнения процессом среднего приоритета. Если теперь активируется высокоприоритетный процесс, которому требуется мьютекс, который уже удерживается низкоприоритетным процессом, то высокоприоритетный процесс будет заблокирован и процесс среднего приоритета вернется к выполнению. Таким образом, высокоприоритетный процесс сможет продолжить работу только после завершения или блокировки среднеприоритетного процесса и последующего освобождения мьютекса низкоприоритетным процессом.</p> <p>Решение проблемы инверсии приоритета состоит в использовании временного повышения приоритета процесса, владеющего мьютексом, до приоритета наиболее приоритетного процесса из числа ожидающих процессов (наследование приоритета). После освобождения мьютекса приоритет процесса вернется к его исходному уровню.</p>

Критерии оценивания	Шкала оценок (в баллах)
Обучающийся точно описал проблему инверсии приоритета и способ ее решения.	3 балла
Обучающийся точно описал проблему инверсии приоритета и способ ее решения. Ответ содержит незначительные неточности	2 балла
Обучающийся недостаточно точно описал проблему инверсии приоритета или способ ее решения. Ответ не содержит грубых ошибок или неточностей	1 балл
Обучающийся недостаточно точно описал проблему инверсии приоритета и способ ее решения. Ответ содержит грубые ошибки и неточности	0 баллов

### **Компетенция ПК-4**

#### Задания закрытого типа

1) Утилита `strace` в GNU/Linux позволяет

- a) Получить информацию о сделанных системных вызовах для указанного процесса
- b) Получить информацию о потреблении процессорного времени для указанного процесса
- c) Получить информацию о размере стека для указанного процесса
- d) Получить информацию об адресах обращений к памяти для указанного процесса

2) Сравните относительный приоритет политик планирования

- a) Политика планирования `SCHED_FIFO` обеспечивает более высокий приоритет, чем `SCHED_RR`
- b) Политика планирования `SCHED_RR` обеспечивает более высокий приоритет, чем `SCHED_FIFO`
- c) При выборе следующей задачи планировщик не делает различий между `SCHED_FIFO` и `SCHED_RR`
- d) Политики планирования `SCHED_FIFO` и `SCHED_RR` не могут использоваться одновременно

3) Прогрев памяти перед запуском задачи реального времени в Linux позволяет

- a) Привести температуру микросхем памяти к рабочему значению

- b) Исключить страничные ошибки при выполнении программы
  - c) Исключить ошибки в данных при выполнении программы
  - d) Проверить работу памяти
- 4) Системный вызов SYS\_Poll в Linux позволяет
- a) Прочитать из файла большой объем данных
  - b) Записать в файл большой объем данных без использования механизмов отложенной записи
  - c) Проверить целостность метаданных файловой системы
  - d) Заблокировать вызывающий поток управления до появления в файле данных, доступных для чтения
- 5) Системный вызов SYS\_Kill в Linux позволяет (выберите наиболее общее утверждение)
- a) Послать сигнал указанным процессам
  - b) Выключить компьютер
  - c) Завершить все дочерние процессы
  - d) Завершить родительский процесс

Ответы на вопросы

Номер вопроса	Ответ (буква)
1	a
2	c
3	b
4	d
5	a

Задания открытого типа

- 1) В файл /proc/sys/kernel/sched\_rt\_period\_us записано значение 100000, а в файл /proc/sys/kernel/sched\_rt\_runtime\_us – значение 95000. Какую долю процессорного времени (в процентах) могут занять задачи реального времени?
- 2) В системе определены три периодических процесса реального времени. Первый процесс активируется таймером каждые 10 мс на 1 мс, второй – каждый 15 мс на 2 мс, и третий каждые 20 мс на 3 мс. Какому из них должен быть назначен максимальный приоритет при использовании частотно-монотонного планирования (укажите номер процесса из диапазона [1, 3])

Ответы на вопросы

Номер вопроса	Ответ
1	95
2	1

Задания с развёрнутым ответом

- 1) В чем состоит сущность политики планирования по сроку завершения SCHED\_DEADLINE

Ответы на вопросы

Номер вопроса	Ответ (буква)
1	Политика планирования SCHED_DEADLINE в качестве основной дисциплины планирования использует дисциплину EDF (Earliest Deadline First) – на выполнение выбирается процесс с наиболее ранним предельным временем завершения (deadline). При назначении политики SCHED_DEADLINE задаче

	<p>реального времени необходимо указать предельный срок завершения, допустимый для данной задачи, и потребность в чистом процессорном времени, необходимом для выполнения задачи.</p> <p>Если задача с политикой SCHED_DEADLINE не завершается после исчерпания запрошенного интервала чистого процессорного времени, то ее планирование переключается на дисциплину CBS (Constant Bandwidth Server) и задача будет заблокирована до конца текущего периода планирования. На новом периоде планирования задача получит новый грант чистого процессорного времени в соответствии с заявленной потребностью. Таким образом, задача продолжит выполнение с фиксированной долей потребления чистого процессорного времени на каждом периоде планирования.</p>
--	---

Критерии оценивания	Шкала оценок (в баллах)
Обучающийся точно описал сущность политики планирования по SCHED_DEADLINE.	3 балла
Обучающийся точно описал сущность политики планирования по SCHED_DEADLINE. Ответ содержит незначительные неточности	2 балла
Обучающийся недостаточно точно описал сущность политики планирования по SCHED_DEADLINE. Ответ не содержит грубых ошибок или неточностей	1 балл
Обучающийся недостаточно точно описал сущность политики планирования по SCHED_DEADLINE. Ответ содержит грубые ошибки и неточности	0 баллов

## 20.2 Промежуточная аттестация

### Примерный перечень курсовых работ

1. Изучение политик планирования реального времени SGED\_FIFO и SCHED\_RR в Linux
2. Изучение политик планирования реального времени SGED\_DEADLINE в Linux
3. Изучение механизмов резервирования процессоров (процессорных ядер) для задач реального времени в Linux
4. Изучение механизмов ограничения использования процессорного времени в Linux
5. Изучение механизмов ограничения использования памяти в Linux
6. Изучение механизмов изоляции процессов в Linux
7. Блокировка страниц в физической памяти и прогрев памяти для снижения вероятности блокировки процесса реального времени в Linux
8. Изучение методов неблокирующей синхронизации для доступа к разделяемым структурам данных
9. Изучение расширенных возможностей мьютексов и условных переменных в Linux
10. Изучение методов межпроцессного взаимодействия задач реального времени в Linux
11. Изучение механизмов барьеров памяти для неблокирующей синхронизации и гарантированной когерентности разделяемых данных в многопроцессорных (многоядерных) системах
12. Изучение механизмов userfaultfd Linux для делегирования обработки страничных ошибок системы виртуальной памяти задачам пользовательского уровня
13. Изучение низкоуровневых механизмов Linux для доступа к GPIO и внутрисхемным интерфейсам I2C и SPI из программ пользовательского режима
14. Изучение механизмов Netlink для передачи сообщений между процессами пользовательского режима и процессами ядра Linux
15. Изучение файловой системы /proc

### Вопросы к собеседованию

1. Ядра операционной системы

2. Системные вызовы.
3. Принципы реализации многозадачности.
4. Архитектура операционной системы и ее влияние на производительность и надежность.
5. Прерывания.
6. Процессы и их состояния.
7. Планирование загрузки процессоров.
8. Приоритетное планирование, проблема инверсии приоритетов.
9. Планирование загрузки процессоров в системах реального времени.
10. Синхронизация выполнения процессов.
11. Передача данных между процессами.
12. Локализация ссылок при обращении к памяти.
13. Иерархическая модель памяти.
14. Динамическое распределение памяти.
15. Виртуальная память.
16. Процессорный кэш и его влияние на согласованность данных.
17. Сигналы в Linux.
18. Резервирование процессоров и прогрев памяти для задач реального времени.
19. Ожидание на файловых дескрипторах.
20. Группы управления в Linux.

#### **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.

#### Примеры КИМ для экзамена

Контрольно-измерительный материал № 1

1. Ядра операционной системы.
2. Синхронизация выполнения процессов.

Контрольно-измерительный материал № 2

1. Принципы реализации многозадачности.
2. Передача данных между процессами.

Контрольно-измерительный материал № 3

1. Системные вызовы.
2. Иерархическая модель памяти.

Контрольно-измерительный материал № 4

1. Архитектура операционной системы и ее влияние на производительность и надежность.
2. Сигналы в Linux.

#### **Требования к выполнению заданий, шкалы и критерии оценивания**

Оценка знаний, умений и навыков, характеризующая этапы формирования компетенций в рамках изучения дисциплины, осуществляется в ходе двух текущих и промежуточной аттестаций. При оценивании результатов промежуточной аттестации используется количественная шкала оценок. Оценки за текущие аттестации, курсовую работу и собеседование на экзамена суммируются, 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