

Документ подписан простой электронной подписью  
Информация о владельце:  
ФИО: Косенок Сергей Михайлович  
Должность: ректор  
Дата подписания: 20.06.2025 08:42:52  
Уникальный программный ключ:  
e3a68f3eaa1e62674b54f4998099d3d6bfdcf836

**Оценочные материалы для текущего контроля и промежуточной аттестации по дисциплине**

**Операционные системы**

Код, направление подготовки	09.03.04 Программная инженерия
Направленность (профиль)	Программное обеспечение компьютерных систем
Форма обучения	очная
Кафедра-разработчик	Автоматики и компьютерных систем
Выпускающая кафедра	Автоматики и компьютерных систем

### Типовые контрольные задания

Текущий контроль в форме контрольной работы с открытыми вопросами. Примерное содержание вопросов представлено ниже:

- приведен пример фрагмент кода многопоточной программы (не более 20 строк).. Необходимо указать, что будет результатом работы программы (например, результаты вывода на экран) и аргументировать свой ответ;
- заданы некоторые исходные данные, необходимо выполнить сравнительный анализ (архитектур различных операционных систем, их подсистем, алгоритмов работы и др.);
- заданы некоторые исходные данные, необходимо выполнить проанализировать текущего состояния системы и/или ее свойств и сделать выводы;
- заданы характеристики процессов и алгоритм планирования, использующийся в системе, необходимо построить временную диаграмму выполнения процессов;
- необходимо построить программу (возможно используя псевдокод), решающую некоторую задачу из области системного программного обеспечения (организовать связь между процессами с использованием именного/анонимного канала, почтового ящика и пр.).

### Примерный перечень экзаменационных вопросов:

1. Назначение и функции операционной системы. Основные компоненты операционной системы.
2. Системные вызовы. Пользовательский и привилегированный режим.
3. Мультипрограммирование: понятие, назначение, необходимые сопутствующие механизмы.
4. Микроядерная и монолитная структура ОС: принципы реализации, достоинства и недостатки.
5. Прерывания.
6. Процессы и потоки: понятие, используемые структуры данных, принципы управления.
7. Модель состояний процесса. Связь модели состояний процесса с планированием процессов и механизмом прерываний.
8. Переключение контекста процесса (потока).
9. Операции над процессами.
10. Блок управления процессом: назначение, структура, состав.
11. Планирование процессов: назначение, параметры, цели.
12. Вытесняющее и невытесняющее планирование. Приоритетное и неприоритетное планирование.
13. Основные алгоритмы планирования: FCFS, SJF.
14. Основные алгоритмы планирования: RR, планирование с использованием приоритетов.
15. Алгоритмы планирования в системах реального времени.
16. Гарантированное планирование.
17. Планирование процессов на основе многоуровневых очередей.
18. Взаимодействие процессов. Логические аспекты взаимодействия процессов.
19. Средства взаимодействия процессов: именованные и анонимные каналы, почтовые ящики, сокеты.
20. Синхронизация процессов. Атомарность. Детерминированность.
21. Программные алгоритмы синхронизации.
22. Высокоуровневые средства синхронизации.
23. Тупики. Понятие ресурса. Условия возникновения.
24. Тупики. Направления борьбы. Алгоритм банкира.
25. Управление памятью: назначение, функции и базовые концепции.
26. Связывание логических и физических адресов.
27. Распределение памяти фиксированными разделами.
28. Распределение памяти динамическими разделами.
29. Распределение памяти перемещаемыми разделами.
30. Страничное распределение.
31. Сегментное распределение.

32. Сегментно-страничное распределение.
33. Алгоритм обработки страничного прерывания. Стратегии выгрузки и подкачки.
34. Файловая система: назначение, функции.
35. Логические аспекты организации файловых систем.
36. Физические аспекты организации файловых систем.
37. Файлы: именование, атрибуты, операции над файлами.
38. Фрагментация: понятие, области возникновения, методы борьбы с фрагментацией.
39. Windows: реализация процессов, планирование процессов.
40. Средства синхронизации Windows.
41. Трансляторы: понятие, принципы работы.
42. Современные системы программирования.

### **Примерные практические задания промежуточной аттестации (экзамен):**

Формулировка задания: заданы некоторые исходные данные, требуется сформулировать ответ, содержащий решение поставленной задачи: построить временную диаграмму, указать формулу (математическую модель) и/или численное значение, разработать алгоритм (программу) и др.

Решаемая задача направлена на выявление следующих знаний, умений, навыков (по отдельности или в сочетаниях):

- формализации вычислительной задачи;
- умение выполнить анализ представленного описания подсистемы (алгоритма, сложившегося в системе состояния и пр.), в том числе с целью инсталляции/настройки системы;
- выделить основные свойства подсистемы (алгоритма, состояния процесса и пр.), рассматриваемой в задаче;
- показать знания современных тенденций развития системного программного обеспечения и операционных систем.

Практические задания промежуточной аттестации формулируются в следующем виде:

- заданы характеристики процессов и алгоритм планирования, использующийся в системе, необходимо построить временную диаграмму выполнения процессов и/или ответить на вопрос о некоторых интегральных значениях характеристик процессов;
- заданы значения служебных таблиц (фрагменты) данных подсистемы управления памятью, необходимо связать значения виртуальных и физических адресов;
- задан набор процессов и ресурсов, которые либо распределены между указанными процессами, либо требуются им для дальнейшего выполнения, необходимо определить состояние системы (наличие тупика) и аргументировать ответ;
- задан набор процессов (программ) и их характеристики (требующиеся время и оперативная память), указан алгоритм распределения памяти, необходимо определить последовательность выполнения процессов (программ);
- задан набор процессов и их характеристики, необходимо определить способность алгоритма планирования реального времени «выполнить» заданные процессы;
- представлены исходные данные некоторой задачи взаимодействия процессов, необходимо решить задачу синхронизации процессов (построить алгоритм, написать псевдокод) с использованием заданных механизмов синхронизации высокого уровня.

При выполнении задания обучающийся демонстрирует знание структуры системного программного обеспечения, архитектуры операционных систем, принципов реализации алгоритмов и структур данных, которые используются в операционных системах, умения и навыки анализа данной предметной области, умение найти решение с использованием знаний алгоритмов работы различных подсистем операционных систем; знание принципов функционирования и настройки системного программного обеспечения, что свидетельствует о наличии соответствующих компетенций.