

УТВЕРЖДАЮ:

Проректор по учебной работе

\_\_\_\_\_ / О.В. Юсупова

" \_\_\_\_ " \_\_\_\_\_ 20\_\_ г.

## РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

### Б1.О.03.07 «Операционные системы»

<b>Код и направление подготовки (специальность)</b>	01.03.02 Прикладная математика и информатика
<b>Направленность (профиль)</b>	Прикладная математика и информатика
<b>Квалификация</b>	Бакалавр
<b>Форма обучения</b>	Очная
<b>Год начала подготовки</b>	2022
<b>Институт / факультет</b>	Институт автоматизации и информационных технологий
<b>Выпускающая кафедра</b>	кафедра "Прикладная математика и информатика"
<b>Кафедра-разработчик</b>	кафедра "Прикладная математика и информатика"
<b>Объем дисциплины, ч. / з.е.</b>	216 / 6
<b>Форма контроля (промежуточная аттестация)</b>	Зачет с оценкой, Экзамен

### **Б1.О.03.07 «Операционные системы»**

Рабочая программа дисциплины разработана в соответствии с требованиями ФГОС ВО по направлению подготовки (специальности) **01.03.02 Прикладная математика и информатика**, утвержденного приказом Министерства образования и науки РФ от № 9 от 10.01.2018 и соответствующего учебного плана.

Разработчик РПД:

Доцент, кандидат физико-математических наук, доцент  
(должность, степень, ученое звание)

М.Н Саушкин

(ФИО)

Заведующий кафедрой

В.П. Радченко, доктор физико-математических наук, профессор

(ФИО, степень, ученое звание)

**СОГЛАСОВАНО:**

Председатель методического совета факультета / института (или учебно-методической комиссии)

Я.Г Стельмах, кандидат педагогических наук

(ФИО, степень, ученое звание)

Руководитель образовательной программы

В.П. Радченко, доктор физико-математических наук, профессор

(ФИО, степень, ученое звание)

## Содержание

1. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами освоения образовательной программы .....	4
2. Место дисциплины (модуля) в структуре образовательной программы .....	4
3. Объем дисциплины (модуля) в зачетных единицах с указанием количества академических часов, выделенных на контактную работу обучающихся с преподавателем (по видам учебных занятий) и на самостоятельную работу обучающихся .....	5
4. Содержание дисциплины (модуля), структурированное по темам (разделам), с указанием отведенного на них количества академических часов и видов учебных занятий .....	5
4.1 Содержание лекционных занятий .....	6
4.2 Содержание лабораторных занятий .....	9
4.3 Содержание практических занятий .....	13
4.4. Содержание самостоятельной работы .....	14
5. Перечень учебной литературы и учебно-методического обеспечения по дисциплине (модулю) .....	24
6. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень программного обеспечения .....	25
7. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», профессиональных баз данных, информационно-справочных систем .....	25
8. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю) .....	26
9. Методические материалы .....	27
10. Фонд оценочных средств по дисциплине (модулю) .....	28

# 1. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами освоения образовательной программы

Наименование категории (группы) компетенций	Код и наименование компетенции	Код и наименование индикатора достижения компетенции	Результаты обучения (знать, уметь, владеть, соотнесенные с индикаторами достижения компетенции)
Общепрофессиональные компетенции			
Теоретические и практические основы профессиональной деятельности	ОПК-2 Способен использовать и адаптировать существующие математические методы и системы программирования для разработки и реализации алгоритмов решения прикладных задач	ОПК-2.1 Знает существующие математические методы и системы программирования для разработки и реализации алгоритмов решения прикладных задач; методы и алгоритмы управления процессами и ресурсами операционной системы; основные принципы организации и управления памятью; основные принципы диспетчирования процессов и потоков в системах; взаимосвязь архитектурных особенностей аппаратуры ЭВМ и компонентов системного программного обеспечения; основные подходы к обеспечению безопасности функционирования операционных систем	Знать используемые методы и алгоритмы реализации системного программного обеспечения
		ОПК-2.2 Умеет разрабатывать и реализовывать алгоритмы решения прикладных задач, используя существующие математические методы, системы программирования и программные комплексы	Уметь реализовывать простейшие алгоритмы, используемые для построения системного программного обеспечения
Информационно-коммуникационные технологии для профессиональной деятельности	ОПК-4 Способен решать задачи профессиональной деятельности с использованием существующих информационно-коммуникационных технологий и с учетом основных требований информационной безопасности	ОПК-4.2 Умеет эффективно использовать современные информационные технологии для решения задач профессиональной деятельности	Уметь использовать информационные сервисы глобальных телекоммуникаций, базы данных, вебресурсы, системное и программное обеспечение;

## 2. Место дисциплины (модуля) в структуре образовательной программы

Место дисциплины (модуля) в структуре образовательной программы: **обязательная часть**

Код компетенции	Предшествующие дисциплины	Параллельно осваиваемые дисциплины	Последующие дисциплины
ОПК-2	Лабораторный практикум (практикум на ЭВМ); Учебная практика: научно-исследовательская работа (получение первичных навыков научно-исследовательской работы); Численные методы; Языки и методы программирования	Производственная практика: научно-исследовательская работа (рассредоточенная)	Выполнение и защита выпускной квалификационной работы; Математические основы методов защиты информации; Подготовка к сдаче и сдача государственного экзамена

ОПК-4	Информационные технологии и программирование; Лабораторный практикум (практикум на ЭВМ); Учебная практика: научно-исследовательская работа (получение первичных навыков научно-исследовательской работы); Языки и методы программирования	Производственная практика: научно-исследовательская работа (рассредоточенная)	Выполнение и защита выпускной квалификационной работы; Математические основы методов защиты информации; Подготовка к сдаче и сдача государственного экзамена
-------	---	---	--

### 3. Объем дисциплины (модуля) в зачетных единицах с указанием количества академических часов, выделенных на контактную работу обучающихся с преподавателем (по видам учебных занятий) и на самостоятельную работу обучающихся

Вид учебной работы	Всего часов / часов в электронной форме	5 семестр часов / часов в электронной форме	6 семестр часов / часов в электронной форме
<b>Аудиторная контактная работа (всего),</b> в том числе:	128	48	80
Лабораторные работы	80	32	48
Лекции	48	16	32
<b>Внеаудиторная контактная работа, КСР</b>	8	3	5
<b>Самостоятельная работа (всего),</b> в том числе:	44	21	23
выполнение курсовых проектов	26	13	13
подготовка к зачету	3	3	0
подготовка к лабораторным работам	5	2	3
подготовка к лекциям	6	3	3
подготовка к экзамену	4	0	4
<b>Контроль</b>	36	0	36
<b>Итого: час</b>	216	72	144
<b>Итого: з.е.</b>	6	2	4

### 4. Содержание дисциплины (модуля), структурированное по темам (разделам), с указанием отведенного на них количества академических часов и видов учебных занятий

№ раздела	Наименование раздела дисциплины	Виды учебной нагрузки и их трудоемкость, часы				
		ЛЗ	ЛР	ПЗ	СРС	Всего часов
1	Введение в операционные системы	2	10	0	2	14

2	Процессы операционной системы	10	14	0	9	33
3	Синхронизация процессов	10	12	0	11	33
4	Управление памятью	8	12	0	9	29
5	Файловая система	10	14	0	10	34
6	Сети и сетевые операционные системы	8	18	0	3	29
	<b>КСР</b>	0	0	0	0	8
	<b>Контроль</b>	0	0	0	0	36
	<b>Итого</b>	48	80	0	44	216

#### 4.1 Содержание лекционных занятий

№ занятия	Наименование раздела	Тема лекции	Содержание лекции (перечень дидактических единиц; рассматриваемых подтем, вопросов)	Количество часов / часов в электронной форме
<b>5 семестр</b>				
1	Введение в операционные системы	Понятие операционной системы	Структура вычислительной системы. Понятие ОС. Краткая история эволюции вычислительных систем и ОС. Основные понятия, концепции ОС. Архитектурные особенности ОС. Функции ОС. Подходы к построению ОС.	2
2	Процессы операционной системы	Процессы	Понятие процесса. Состояния процесса. Операции над процессами и связанные с ними понятия (набор операций; Process Control Block и контекст процесса; одноразовые операции; многократные операции; переключение контекста).	2
3	Процессы операционной системы	Планирование процессов	Уровни планирования. Критерии планирования и требования к алгоритмам. Параметры планирования. Вытесняющее и невытесняющее планирование.	2
4	Процессы операционной системы	Алгоритмы планирования	First-Come. First-Served. Round Robin. Shortest-Job-First. Гарантированное планирование.	2
5	Процессы операционной системы	Алгоритмы планирования (продолжение)	Приоритетное планирование. Multilevel Queue. Multilevel Feedback Queue.	2
6	Процессы операционной системы	Кооперация процессов	Взаимодействующие процессы. Категории средств обмена информацией. Логическая организация механизма передачи информации. Нити исполнения.	2
7	Синхронизация процессов	Активности процессов	Interleaving, race condition и взаимоисключения. Критическая секция.	2

8	Синхронизация процессов	Алгоритмы синхронизации	Программные алгоритмы организации взаимодействия процессов и требования, предъявляемые к алгоритмам. Запрет прерываний. Переменная-замок. Строгое чередование. Флаги готовности. Алгоритм Петерсона. Bakery Algorithm. Аппаратная поддержка взаимоисключений. Команда Test-and-Set.	2
<b>Итого за семестр:</b>				<b>16</b>
<b>6 семестр</b>				
9	Синхронизация процессов	Механизмы синхронизации	Семафоры (концепция семафоров; решение проблемы producer-consumer с помощью семафоров). Мониторы. Сообщения. Эквивалентность семафоров, мониторов и сообщений.	2
10	Синхронизация процессов	Механизмы синхронизации (продолжение)	Реализация мониторов и передачи сообщений с помощью семафоров. Реализация семафоров и передачи сообщений с помощью мониторов. Реализация семафоров и мониторов с помощью очередей сообщений.	2
11	Синхронизация процессов	Тупики	Условия возникновения тупиков. Основные направления борьбы с тупиками. Игнорирование проблемы тупиков. Способы предотвращения тупиков. Обнаружение тупиков. Восстановление после тупиков.	2
12	Управление памятью	Организация памяти компьютера	Простейшие схемы управления памятью. Физическая организация памяти компьютера. Логическая память. Связывание адресов. Функции системы управления памятью. Простейшие схемы управления памятью. Страничная память. Сегментная и сегментно-страничная организация памяти.	2
13	Управление памятью	Виртуальная память	Архитектурные средства поддержки виртуальной памяти. Понятие виртуальной памяти. Страничная виртуальная память. Сегментно-страничная организации виртуальной памяти. Структура таблицы страниц. Ассоциативная память. Инвертированная таблица страниц. Размер страницы.	2
14	Управление памятью	Аппаратно-независимый уровень управления виртуальной памятью	Исключительные ситуации при работе с памятью. Стратегии управления страничной памятью. Алгоритмы замещения страниц (алгоритм FIFO; алгоритм OPT; алгоритм LRU; алгоритм NFU).	2
15	Управление памятью	Аппаратно-независимый уровень управления виртуальной памятью (продолжение)	Управление количеством страниц, выделенных процессу. Модель рабочего множества. Страничные демоны. Программная поддержка сегментной модели памяти процесса. Функционирования менеджера памяти.	2

16	Файловая система	Файлы с точки зрения пользователя	Общие сведения о файлах. Организация файлов и доступ к ним. Операции над файлами. Директории. Логическая структура файлового архива. Операции над директориями. Защита файлов.	2
17	Файловая система	Синхронизация процессов при работе с файловой системой	Доступ процессов к файлам и каталогам. Процессы (запуск дочерних процессов; сигналы). Доступ к файлу и каталогу (чтение, запись и использование; доступ к файлу; сценарий; доступ к каталогу).	2
18	Файловая система	Реализация файловой системы	Общая структура файловой системы. Управление внешней памятью. Реализация директорий. Монтирование файловых систем. Связывание файлов. Кооперация процессов при работе с файлами.	2
19	Файловая система	Надежность и производительность файловых систем	Надежность файловой системы. Производительность файловой системы. Реализация некоторых операций над файлами. Современные архитектуры файловых систем.	2
20	Файловая система	Система управления вводом-выводом	Физические принципы организации ввода-вывода. Логические принципы организации ввода-вывода. Систематизация внешних устройств и интерфейс между базовой подсистемой ввода-вывода и драйверами. Функции базовой подсистемы ввода-вывода. Алгоритмы планирования запросов к жесткому диску.	2
21	Сети и сетевые операционные системы	Сетевые операционные системы и основные понятия	Сетевые и распределенные операционные системы. Взаимодействие удаленных процессов как основа работы вычислительных сетей. Основные вопросы логической организации передачи информации между удаленными процессами.	2
22	Сети и сетевые операционные системы	Интерфейсы и сетевые протоколы	Понятие протокола. Многоуровневая модель построения сетевых вычислительных систем. Проблемы адресации в сети. Проблемы маршрутизации в сетях. Связь с установлением логического соединения и передача данных с помощью сообщений. Синхронизация удаленных процессов.	2
23	Сети и сетевые операционные системы	Защитные механизмы операционных систем	Основные понятия информационной безопасности. Идентификация и аутентификация. Авторизация. Разграничение доступа к объектам ОС. Разграничение доступа к объектам ОС. Выявление вторжений. Аудит системы защиты.	2

24	Сети и сетевые операционные системы	Защищенность операционных систем	Анализ некоторых популярных ОС с точки зрения их защищенности. Угрозы безопасности. Формализация подхода к обеспечению информационной безопасности. Криптография как одна из базовых технологий безопасности ОС.	2
<b>Итого за семестр:</b>				<b>32</b>
<b>Итого:</b>				<b>48</b>

#### 4.2 Содержание лабораторных занятий

№ занятия	Наименование раздела	Тема лабораторного занятия	Содержание лабораторного занятия (перечень дидактических единиц: рассматриваемых подтем, вопросов)	Количество часов / часов в электронной форме
<b>5 семестр</b>				
1	Введение в операционные системы	Знакомство с операционной системой Linux	Системные вызовы и библиотека libc. Понятия login и password. Упрощенное понятие об устройстве файловой системы в Linux. Перенаправление ввода и вывода. Простейшие команды для работы с файлами. Команды chown, chgrp, umask. Права доступа к файлу. Системные вызовы getuid и getgid. Компиляция программ на языке C в Linux и запуск их на выполнение.	2
2	Введение в операционные системы	Командная оболочка bash (Часть 1)	Служебные символы. Переменные и параметры (подстановка переменных; присваивание значений переменным; специальные типы переменных). Завершение и код завершения. Проверка условий (конструкции проверки условий; операции проверки файлов; операции сравнения; вложенные условные операторы if/then). Операции и смежные темы (операторы; числовые константы).	2
3	Введение в операционные системы	Командная оболочка bash (Часть 2)	Переменные (внутренние переменные; работа со строка-ми; подстановка параметров; объявление переменных: declare и typeset; косвенные ссылки на переменные; \$RANDOM: генерация псевдослучайных целых чисел; двойные круглые скобки). Циклы и ветвления (циклы; вложенные циклы; управление ходом выполнения цикла; операторы выбора).	2
4	Введение в операционные системы	Командная оболочка bash (Часть 3)	Внутренние команды. Внешние команды, программы и утилиты. Команды системного администрирования. Подстановка команд. Арифметические подстановки. Перенаправление ввода/вывода.	2

5	Введение в операционные системы	Основы программирования в ОС Linux	Редактор nano и его настройки для подсветки синтаксиса. Проект Hello World. Мультифайловое программирование. Автоматическая сборка. Модель клиент-интерфейс-сервер (КИС). Библиотеки. Сборка проектов со статическими библиотеками. Сборка проектов с совместно используемыми библиотеками.	2
6	Процессы операционной системы	Процессы в ОС Linux (Часть 1)	Процессы в ОС Linux. Понятие процесса в ОС Linux, его контекст. Идентификация процесса. Состояния процесса. Диаграмма состояний.	2
7	Процессы операционной системы	Процессы в ОС Linux (Часть 2)	Иерархия процессов. Системные вызовы getpid(), getppid(). Создание процесса в Linux. Системный вызов fork(). Завершение процесса. Функция exit().	2
8	Процессы операционной системы	Процессы в ОС Linux (Часть 3)	Изменение пользовательского контекста процесса. Семейство функций для системного вызова exec().	2
9	Процессы операционной системы	Переменные среды и аргументы командной строки (Часть 1)	Параметры функции main() в языке C. Массив environ. Функции getenv(), setenv(). «Сырая» модификация окружения putenv().	2
10	Процессы операционной системы	Переменные среды и аргументы командной строки (Часть 2)	Удаление переменной окружения unsetenv(). Очистка окружения clearenv().	2
11	Процессы операционной системы	Низкоуровневый ввод-вывод (Часть 1)	Понятие потока ввода-вывода. Представление о работе с файлами через системные вызовы и стандартную библиотеку ввода-вывода.	2
12	Процессы операционной системы	Низкоуровневый ввод-вывод (Часть 2)	Понятие файлового дескриптора. Открытие файла. Системный вызов open(). Системные вызовы close(), read(), write().	2
13	Синхронизация процессов	Организация взаимодействия процессов через pipe в ОС Linux (Часть 1)	Понятие pipe. Системный вызов pipe(). Организация связи через pipe между процессом-родителем и процессом-потомком.	2
14	Синхронизация процессов	Организация взаимодействия процессов через pipe в ОС Linux (Часть 2)	Наследование файловых дескрипторов при вызовах fork() и exec(). Особенности поведения вызовов read() и write() для pip'a.	2
15	Синхронизация процессов	Организация взаимодействия процессов через FIFO в ОС Linux (Часть 1)	Понятие FIFO. Использование системного вызова mkfifo() для создания FIFO.	2

16	Синхронизация процессов	Организация взаимодействия процессов через FIFO в ОС Linux (Часть 2)	Функция mkfifo(). Особенности поведения вызова open() при открытии FIFO.	2
<b>Итого за семестр:</b>				<b>32</b>
<b>6 семестр</b>				
17	Синхронизация процессов	Очереди сообщений в ОС Linux (Часть 1)	Очереди сообщений в ОС Linux как средства связи и средства синхронизации процессов. Очереди сообщений в ОС Linux как составная часть System V IPC. Создание очереди сообщений или доступ к уже существующей. Системный вызов msgget(). Реализация примитивов send и receive.	2
18	Синхронизация процессов	Очереди сообщений в ОС Linux (Часть 2)	Сообщения Системные вызовы msgsnd() и msgrcv(). Удаление очереди сообщений из системы с помощью команды ipcrm или системного вызова msgctl(). Понятие мультиплексирования. Мультиплексирование сообщений. Модель взаимодействия процессов клиент-сервер. Неравноправность клиента и сервера. Использование очередей сообщений для синхронизации работы процессов.	2
19	Управление памятью	Организация работы с разделяемой памятью в ОС Linux (Часть 1)	Понятие нитей исполнения (thread). Понятие System V IPC. Пространство имен. Адресация в System V IPC. Функция ftok(). Дескрипторы System V IPC. Разделяемая память в Linux. Системные вызовы shmget(), shmat(), shmdt(). Команды ipc и ipcrm.	2
20	Управление памятью	Организация работы с разделяемой памятью в ОС Linux (Часть 2)	Использование системного вызова shmctl() для освобождения ресурса. Разделяемая память и системные вызовы fork(), exec() и функция exit(). Понятие о нити исполнения (thread) в ОС Linux. Идентификатор нити исполнения. Функция pthread - self(). Создание и завершение thread'a. Функции pthread - create(), pthread - exit(), pthread - join().	2
21	Управление памятью	Организация работы с разделяемой памятью в ОС Linux (Часть 3)	Необходимость синхронизации процессов и нитей исполнения, использующих общую память.	2
22	Управление памятью	Семафоры в ОС Linux (Часть 1)	Семафоры в ОС Linux как средство синхронизации процессов. Семафоры в ОС Linux. Отличие операций над Linux-семафорами от классических операций. Создание массива семафоров или доступ к уже существующему массиву. Системный вызов semget().	2

23	Управление памятью	Семафоры в ОС Linux (Часть 2)	Выполнение операций над семафорами. Системный вызов <code>semop()</code> . Удаление набора семафоров из системы с помощью команды <code>ipcrm</code> или системного вызова <code>semctl()</code> . Понятие о POSIX-семафорах.	2
24	Управление памятью	Организация ввода-вывода в Linux (Часть 3)	Системный вызов <code>kill()</code> и команда <code>kill</code> . Системный вызов <code>signal()</code> . Установка собственного обработчика сигнала. Восстановление предыдущей реакции на сигнал. Сигналы <code>SIGUSR1</code> и <code>SIGUSR2</code> . Использование сигналов для синхронизации процессов. Понятие о надежности сигналов. POSIX-функции для работы с сигналами.	2
25	Файловая система	Работа с файловой системой	Практическая работа с объектами файловой системы: перемещению по дереву каталогов, копированию, перемещению и удалению файлов, созданию жестких и символических ссылок.	2
26	Файловая система	Работа с файлами и директориями (Часть 1)	Понятие о <code>memory mapped</code> файлах. Разделы носителя информации ( <code>partitions</code> ) в Linux. Организация файла на диске в Linux на примере файловой системы <code>s5fs</code> . Понятие индексного узла ( <code>inode</code> ). Понятие суперблока. Операции над файлами и директориями.	2
27	Файловая система	Работа с файлами и директориями (Часть 2)	Системные вызовы и команды для выполнения операций над файлами и директориями. Системные вызовы <code>open()</code> , <code>close()</code> , <code>creat()</code> . Операции с атрибутами файла. Системные вызовы <code>stat()</code> , <code>fstat()</code> и <code>lstat()</code> . Системный вызов <code>lseek()</code> .	2
28	Файловая система	Работа с файлами и директориями (Часть 3)	Операции создания связей. Команда <code>ln</code> , системные вызовы <code>link()</code> и <code>symlink()</code> . Системный вызов <code>unlink()</code> . Понятие о файлах, отображаемых в память ( <code>memory mapped</code> файлах). Системные вызовы <code>mmap()</code> , <code>munmap()</code> .	2
29	Файловая система	Организация ввода-вывода в Linux (Часть 1)	Файлы устройств. Аппарат прерываний. Сигналы в Linux. Понятие виртуальной файловой системы. Операции над файловыми системами. Блочные, символические драйверы, драйверы низкого уровня. Файловый интерфейс.	2
30	Файловая система	Организация ввода-вывода в Linux (Часть 2)	Аппаратные прерывания ( <code>interrupt</code> ), исключения ( <code>exception</code> ), программные прерывания ( <code>trap</code> , <code>software interrupt</code> ). Понятие сигнала. Способы возникновения сигналов и виды их обработки. Понятия группы процессов, сеанса, лидера группы, лидера сеанса, управляющего терминала сеанса. Системные вызовы <code>getpgrp()</code> , <code>setpgrp()</code> , <code>getpgid()</code> , <code>setpgid()</code> , <code>getsid()</code> , <code>setsid()</code> .	2

31	Файловая система	Протокол IP	Структура пакета IP. Структура IP-адреса. Подсети. ifconfig и настройка протокола IP. Маршрутизация. Автономные области. M9. Команда route.	2
32	Сети и сетевые операционные системы	Основы сетевого программирования (Часть 1)	Семейство протоколов TCP/IP. Сокеты (sockets) в Linux и основы работы с ними. Общие сведения об архитектуре семейства протоколов TCP/IP. Протоколы IP, ICMP, ARP, RARP.	2
33	Сети и сетевые операционные системы	Основы сетевого программирования (Часть 2)	Протоколы TCP и UDP. UDP и TCP сокеты (sockets). Адресные пространства портов. Использование модели клиент-сервер при изучении сетевого программирования. Организация связи между удаленными процессами с помощью датаграмм. Функции htons(), htonl(), ntohs(), ntohl().	2
34	Сети и сетевые операционные системы	Основы сетевого программирования (Часть 3)	Функции преобразования IP-адресов inet - ntoa(), inet - aton(). Функция bzero(). Системные вызовы socket(), bind(), sendto(), recvfrom(). Системные вызовы connect(), listen(), accept(). Использование интерфейса сокетов для других семейств протоколов.	2
35	Сети и сетевые операционные системы	Сетевая модель OSI	Структура модели OSI. Семейство протоколов TCP/IP.	2
36	Сети и сетевые операционные системы	Протокол UDP	Структура полного адреса в протоколе UDP. Понятие UDP-портов. Структура UDP-пакета. Распространенные UDP-сервисы. DNS. Bind.	2
37	Сети и сетевые операционные системы	Протокол TCP (Часть 1)	Структура полного адреса в протоколе TCP. Понятие TCP-соединения. Структура TCP-пакета. Распространенные TCP-сервисы. Telnet: xinetd, in. telnetd. SSH: sshd.	2
38	Сети и сетевые операционные системы	Протокол TCP (Часть 2)	FTP. vsftpd. HTTP. Apache.	2
39	Сети и сетевые операционные системы	ICMP	Протокол ICMP. Типы пакетов. Утилиты ping, traceroute, tcptraceroute. Утилиты управления сетью. Nmap. NetCat. Netstat.	2
40	Сети и сетевые операционные системы	Iptables	Таблицы. Цепочки. Прохождение трафика. Механизм определения состояний. Базовый синтаксис и команды.	2
<b>Итого за семестр:</b>				<b>48</b>
<b>Итого:</b>				<b>80</b>

### 4.3 Содержание практических занятий

Учебные занятия не реализуются.

#### 4.4. Содержание самостоятельной работы

Наименование раздела	Вид самостоятельной работы	Содержание самостоятельной работы (перечень дидактических единиц: рассматриваемых подтем, вопросов)	Количество часов
<b>5 семестр</b>			
Введение в операционные системы	подготовка к лекциям, выполнение соответствующих заданий	Структура вычислительной системы. Понятие ОС. Краткая история эволюции вычислительных систем и ОС. Основные понятия, концепции ОС. Архитектурные особенности ОС. Функции ОС. Подходы к построению ОС.	1
Введение в операционные системы	подготовка к зачету	Структура вычислительной системы. Понятие ОС. Краткая история эволюции вычислительных систем и ОС. Основные понятия, концепции ОС. Архитектурные особенности ОС. Функции ОС. Подходы к построению ОС.	1
Процессы операционной системы	подготовка к лекциям, выполнение соответствующих заданий	Понятие процесса. Состояния процесса. Операции над процессами и связанные с ними понятия (набор операций; Process Control Block и контекст процесса; одноразовые операции; многократные операции; переключение контекста). Уровни планирования. Критерии планирования и требования к алгоритмам. Параметры планирования. Вытесняющее и невытесняющее планирование. First-Come. First-Served. Round Robin. Shortest-Job-First. Гарантированное планирование. Приоритетное планирование. Multilevel Queue. Multilevel Feedback Queue. Взаимодействующие процессы. Категории средств обмена информацией. Логическая организация механизма передачи информации. Нити исполнения.	1

<p>Процессы операционной системы</p>	<p>подготовка к зачету</p>	<p>Понятие процесса. Состояния процесса. Операции над процессами и связанные с ними понятия (набор операций; Process Control Block и контекст процесса; одноразовые операции; многократные операции; переключение контекста). Уровни планирования. Критерии планирования и требования к алгоритмам. Параметры планирования. Вытесняющее и невытесняющее планирование. First-Come. First-Served. Round Robin. Shortest-Job-First. Гарантированное планирование. Приоритетное планирование. Multilevel Queue. Multilevel Feedback Queue. Взаимодействующие процессы. Категории средств обмена информацией. Логическая организация механизма передачи информации. Нити исполнения.</p>	<p>1</p>
<p>Процессы операционной системы</p>	<p>выполнение курсовых проектов</p>	<p>Состояния процесса. Операции над процессами и связанные с ними понятия (набор операций; Process Control Block и контекст процесса; одноразовые операции; многократные операции; переключение контекста). Уровни планирования. Критерии планирования и требования к алгоритмам. Параметры планирования. Вытесняющее и невытесняющее планирование. First-Come. First-Served. Round Robin. Shortest-Job-First. Гарантированное планирование. Приоритетное планирование. Multilevel Queue. Multilevel Feedback Queue. Взаимодействующие процессы. Категории средств обмена информацией. Логическая организация механизма передачи информации. Нити исполнения.</p>	<p>6</p>

<p>Процессы операционной системы</p>	<p>подготовка к лабораторным работам, выполнение соответствующих заданий</p>	<p>Процессы в ОС Linux. Понятие процесса в ОС Linux, его контекст. Идентификация процесса. Состояния процесса. Диаграмма состояний. Иерархия процессов. Системные вызовы getpid(), getppid(). Создание процесса в Linux. Системный вызов fork(). Завершение процесса. Функция exit(). Изменение пользовательского контекста процесса. Семейство функций для системного вызова exec(). Параметры функции main() в языке C. Массив environ. Функции getenv(), setenv(). «Сырая» модификация окружения putenv(). Удаление переменной окружения unsetenv(). Очистка окружения clearenv(). Понятие потока ввода-вывода. Представление о работе с файлами через системные вызовы и стандартную библиотеку ввода-вывода. Понятие файлового дескриптора. Открытие файла. Системный вызов open(). Системные вызовы close(), read(), write().</p>	<p>1</p>
<p>Синхронизация процессов</p>	<p>подготовка к лабораторным работам, выполнение соответствующих заданий</p>	<p>Interleaving, race condition и взаимоисключения. Критическая секция. Программные алгоритмы организации взаимодействия процессов и требования, предъявляемые к алгоритмам. Запрет прерываний. Переменная-замок. Строгое чередование. Флаги готовности. Алгоритм Петерсона. Bakery Algorithm. Аппаратная поддержка взаимоисключений. Команда Test-and-Set. Семафоры (концепция семафоров; решение проблемы producer-consumer с помощью семафоров). Мониторы. Сообщения. Эквивалентность семафоров, мониторов и сообщений. Реализация мониторов и передачи сообщений с помощью семафоров. Реализация семафоров и передачи сообщений с помощью мониторов. Реализация семафоров и мониторов с помощью очередей сообщений. Условия возникновения тупиков. Основные направления борьбы с тупиками. Игнорирование проблемы тупиков. Способы предотвращения тупиков. Обнаружение тупиков. Восстановление после тупиков.</p>	<p>1</p>

Синхронизация процессов	подготовка к зачету	Interleaving, race condition и взаимоисключения. Критическая секция. Программные алгоритмы организации взаимодействия процессов и требования, предъявляемые к алгоритмам. Запрет прерываний. Переменная-замок. Строгое чередование. Флаги готовности. Алгоритм Петерсона. Bakery Algorithm. Аппаратная поддержка взаимоисключений. Команда Test-and-Set. Семафоры (концепция семафоров; решение проблемы producer-consumer с помощью семафоров). Мониторы. Сообщения. Эквивалентность семафоров, мониторов и сообщений. Реализация мониторов и передачи сообщений с помощью семафоров. Реализация семафоров и передачи сообщений с помощью мониторов. Реализация семафоров и мониторов с помощью очередей сообщений. Условия возникновения тупиков. Основные направления борьбы с тупиками. Игнорирование проблемы тупиков. Способы предотвращения тупиков. Обнаружение тупиков. Восстановление после тупиков.	1
Синхронизация процессов	выполнение курсовых проектов	Программные алгоритмы организации взаимодействия процессов и требования, предъявляемые к алгоритмам. Семафоры.	7
Синхронизация процессов	подготовка к лабораторным работам, выполнение соответствующих заданий	Понятие pipe. Системный вызов pipe(). Организация связи через pipe между процессом-родителем и процессом-потомком. Наследование файловых дескрипторов при вызовах fork() и exec(). Особенности поведения вызовов read() и write() для pip'a. Понятие FIFO. Использование системного вызова mknod() для создания FIFO. Функция mkfifo(). Особенности поведения вызова open() при открытии FIFO.	1
<b>Итого за семестр:</b>			<b>21</b>
<b>6 семестр</b>			

Синхронизация процессов	подготовка к экзамену	<p>Interleaving, race condition и взаимоисключения. Критическая секция. Программные алгоритмы организации взаимодействия процессов и требования, предъявляемые к алгоритмам. Запрет прерываний. Переменная-замок. Строгое чередование. Флаги готовности. Алгоритм Петерсона. Bakery Algorithm. Аппаратная поддержка взаимоисключений. Команда Test-and-Set. Семафоры (концепция семафоров; решение проблемы producer-consumer с помощью семафоров). Мониторы. Сообщения. Эквивалентность семафоров, мониторов и сообщений. Реализация мониторов и передачи сообщений с помощью семафоров. Реализация семафоров и передачи сообщений с помощью мониторов. Реализация семафоров и мониторов с помощью очередей сообщений. Условия возникновения тупиков. Основные направления борьбы с тупиками. Игнорирование проблемы тупиков. Способы предотвращения тупиков. Обнаружение тупиков. Восстановление после тупиков.</p>	1
Управление памятью	подготовка к лекциям, выполнение соответствующих заданий	<p>Простейшие схемы управления памятью. Физическая организация памяти компьютера. Логическая память. Связывание адресов. Функции системы управления памятью. Простейшие схемы управления памятью. Страничная память. Сегментная и сегментно-страничная организация памяти. Архитектурные средства поддержки виртуальной памяти. Понятие виртуальной памяти. Страничная виртуальная память. Сегментно-страничная организации виртуальной памяти. Структура таблицы страниц. Ассоциативная память. Инвертированная таблица страниц. Размер страницы. Исключительные ситуации при работе с памятью. Стратегии управления страничной памятью. Алгоритмы замещения страниц (алгоритм FIFO; алгоритм OPT; алгоритм LRU; алгоритм NFU). Управление количеством страниц, выделенных процессу. Модель рабочего множества. Страничные демоны. Программная поддержка сегментной модели памяти процесса. Функционирования менеджера памяти.</p>	1

Управление памятью	подготовка к лабораторным работам, выполнение соответствующих заданий	<p>Понятие нитей исполнения (thread).  Понятие System V IPC. Пространство имен. Адресация в System V IPC.  Функция ftok(). Дескрипторы System V IPC. Разделяемая память в Linux.  Системные вызовы shmget(), shmat(), shmdt(). Команды ipc и ipcrm.  Использование системного вызова shmctl() для освобождения ресурса.  Разделяемая память и системные вызовы fork(), exec() и функция exit().  Понятие о нити исполнения (thread) в ОС Linux. Идентификатор нити исполнения. Функция pthread – self().  Создание и завершение thread'a.  Функции pthread – create(), pthread – exit(), pthread – join(). Необходимость синхронизации процессов и нитей исполнения, использующих общую память. Семафоры в ОС Linux как средство синхронизации процессов.  Семафоры в ОС Linux. Отличие операций над Linux-семафорами от классических операций. Создание массива семафоров или доступ к уже существующему массиву. Системный вызов semget(). Выполнение операций над семафорами. Системный вызов semop(). Удаление набора семафоров из системы с помощью команды ipcrm или системного вызова semctl().  Понятие о POSIX-семафорах.</p>	1
Управление памятью	выполнение курсовых проектов	<p>Разделяемая память в Linux.  Системные вызовы shmget(), shmat(), shmdt(). Команды ipc и ipcrm.  Использование системного вызова shmctl() для освобождения ресурса.  Разделяемая память и системные вызовы fork(), exec() и функция exit().  Семафоры в ОС Linux. Системный вызов semget(). Выполнение операций над семафорами. Системный вызов semop().</p>	6

Управление памятью	подготовка к экзамену	<p>Простейшие схемы управления памятью. Физическая организация памяти компьютера. Логическая память. Связывание адресов. Функции системы управления памятью. Простейшие схемы управления памятью. Страничная память. Сегментная и сегментно-страничная организация памяти. Архитектурные средства поддержки виртуальной памяти. Понятие виртуальной памяти. Страничная виртуальная память. Сегментно-страничная организации виртуальной памяти. Структура таблицы страниц. Ассоциативная память. Инвертированная таблица страниц. Размер страницы. Исключительные ситуации при работе с памятью. Стратегии управления страничной памятью. Алгоритмы замещения страниц (алгоритм FIFO; алгоритм OPT; алгоритм LRU; алгоритм NFU). Управление количеством страниц, выделенных процессу. Модель рабочего множества. Страничные демоны. Программная поддержка сегментной модели памяти процесса. Функционирования менеджера памяти.</p>	1
Файловая система	подготовка к лекциям, выполнение соответствующих заданий	<p>Общие сведения о файлах. Организация файлов и доступ к ним. Операции над файлами. Директории. Логическая структура файлового архива. Операции над директориями. Защита файлов. Доступ процессов к файлам и каталогам. Процессы (запуск дочерних процессов; сигналы). Доступ к файлу и каталогу (чтение, запись и использование; доступ к файлу; сценарий; доступ к каталогу). Общая структура файловой системы. Управление внешней памятью. Реализация директорий. Монтирование файловых систем. Связывание файлов. Кооперация процессов при работе с файлами. Надежность файловой системы. Производительность файловой системы. Реализация некоторых операций над файлами. Современные архитектуры файловых систем. Физические принципы организации ввода-вывода. Логические принципы организации ввода-вывода. Систематизация внешних устройств и интерфейс между базовой подсистемой ввода-вывода и драйверами. Функции базовой подсистемы ввода-вывода. Алгоритмы планирования запросов к жесткому диску.</p>	1

Файловая система	подготовка к лабораторным работам, выполнение соответствующих заданий	<p>Практическая работа с объектами файловой системы: перемещению по дереву каталогов, копированию, перемещению и удалению файлов, созданию жестких и символьных ссылок. Понятие о memory mapped файлах. Разделы носителя информации (partitions) в Linux. Организация файла на диске в Linux на примере файловой системы s5fs. Понятие индексного узла (inode). Понятие суперблока. Операции над файлами и директориями. Системные вызовы и команды для выполнения операций над файлами и директориями. Системные вызовы open(), close(), creat(). Операции с атрибутами файла. Системные вызовы stat(), fstat() и lstat(). Системный вызов lseek(). Операции создания связей. Команда ln, системные вызовы link() и symlink(). Системный вызов unlink(). Понятие о файлах, отображаемых в память (memory mapped файлах). Системные вызовы mmap(), munmap().</p> <p>Файлы устройств. Аппарат прерываний. Сигналы в Linux. Понятие виртуальной файловой системы. Операции над файловыми системами. Блочные, символьные драйверы, драйверы низкого уровня. Файловый интерфейс. Аппаратные прерывания (interrupt), исключения (exception), программные прерывания (trap, software interrupt). Понятие сигнала. Способы возникновения сигналов и виды их обработки. Понятия группы процессов, сеанса, лидера группы, лидера сеанса, управляющего терминала сеанса. Системные вызовы getpgrp(), setpgrp(), getpgid(), setpgid(), getsid(), setsid().</p>	1
Файловая система	выполнение курсовых проектов	<p>Операции над файлами и директориями. Системные вызовы и команды для выполнения операций над файлами и директориями. Системные вызовы open(), close(), creat(). Операции с атрибутами файла.</p>	7

Файловая система	подготовка к экзамену	<p>Общие сведения о файлах. Организация файлов и доступ к ним. Операции над файлами. Директории. Логическая структура файлового архива. Операции над директориями. Защита файлов. Доступ процессов к файлам и каталогам. Процессы (запуск дочерних процессов; сигналы). Доступ к файлу и каталогу (чтение, запись и использование; доступ к файлу; сценарий; доступ к каталогу). Общая структура файловой системы. Управление внешней памятью. Реализация директорий. Монтирование файловых систем. Связывание файлов. Кооперация процессов при работе с файлами. Надежность файловой системы. Производительность файловой системы. Реализация некоторых операций над файлами. Современные архитектуры файловых систем. Физические принципы организации ввода-вывода. Логические принципы организации ввода-вывода. Систематизация внешних устройств и интерфейс между базовой подсистемой ввода-вывода и драйверами. Функции базовой подсистемы ввода-вывода. Алгоритмы планирования запросов к жесткому диску.</p>	1
Сети и сетевые операционные системы	подготовка к лекциям, выполнение соответствующих заданий	<p>Сетевые и распределенные операционные системы. Взаимодействие удаленных процессов как основа работы вычислительных сетей. Основные вопросы логической организации передачи информации между удаленными процессами. Понятие протокола. Многоуровневая модель построения сетевых вычислительных систем. Проблемы адресации в сети. Проблемы маршрутизации в сетях. Связь с установлением логического соединения и передача данных с помощью сообщений. Синхронизация удаленных процессов. Основные понятия информационной безопасности. Идентификация и аутентификация. Авторизация. Разграничение доступа к объектам ОС. Разграничение доступа к объектам ОС. Выявление вторжений. Аудит системы защиты. Анализ некоторых популярных ОС с точки зрения их защищенности. Угрозы безопасности. Формализация подхода к обеспечению информационной безопасности. Криптография как одна из базовых технологий безопасности ОС.</p>	1

<p>Сети и сетевые операционные системы</p>	<p>подготовка к лабораторным работам, выполнение соответствующих заданий</p>	<p>Семейство протоколов TCP/IP. Сокеты (sockets) в Linux и основы работы с ними. Общие сведения об архитектуре семейства протоколов TCP/IP. Протоколы IP, ICMP, ARP, RARP. Протоколы TCP и UDP. UDP и TCP сокеты (sockets). Адресные пространства портов. Использование модели клиент-сервер при изучении сетевого программирования. Организация связи между удаленными процессами с помощью датаграмм. Функции htons(), htonl(), ntohs(), ntohl(). Функции преобразования IP-адресов inet - ntoa(), inet - aton(). Функция bzero(). Системные вызовы socket(), bind(), sendto(), recvfrom(). Системные вызовы connect(), listen(), accept(). Использование интерфейса сокетов для других семейств протоколов. Структура модели OSI. Семейство протоколов TCP/IP. Структура пакета IP. Структура IP-адреса. Подсети. ifconfig и настройка протокола IP. Маршрутизация. Автономные области. M9. Команда route. Структура полного адреса в протоколе UDP. Понятие UDP-портов. Структура UDP-пакета. Распространенные UDP-сервисы. DNS. Bind. Структура полного адреса в протоколе TCP. Понятие TCP-соединения. Структура TCP-пакета. Распространенные TCP-сервисы. Telnet: xinetd, in, telnetd. SSH: sshd. FTP. vsftpd. HTTP. Apache. Протокол ICMP. Типы пакетов. Утилиты ping, traceroute, tcptraceroute. Утилиты управления сетью. Nmap. NetCat. Netstat. Таблицы. Цепочки. Прохождение трафика. Механизм определения состояний. Базовый синтаксис и команды.</p>	<p>1</p>
--	--	--	----------

Сети и сетевые операционные системы	подготовка к экзамену	Сетевые и распределенные операционные системы. Взаимодействие удаленных процессов как основа работы вычислительных сетей. Основные вопросы логической организации передачи информации между удаленными процессами. Понятие протокола. Многоуровневая модель построения сетевых вычислительных систем. Проблемы адресации в сети. Проблемы маршрутизации в сетях. Связь с установлением логического соединения и передача данных с помощью сообщений. Синхронизация удаленных процессов. Основные понятия информационной безопасности. Идентификация и аутентификация. Авторизация. Разграничение доступа к объектам ОС. Разграничение доступа к объектам ОС. Выявление вторжений. Аудит системы защиты. Анализ некоторых популярных ОС с точки зрения их защищенности. Угрозы безопасности. Формализация подхода к обеспечению информационной безопасности. Криптография как одна из базовых технологий безопасности ОС.	1
<b>Итого за семестр:</b>			<b>23</b>
<b>Итого:</b>			<b>44</b>

### 5. Перечень учебной литературы и учебно-методического обеспечения по дисциплине (модулю)

№ п/п	Библиографическое описание	Ресурс НТБ СамГТУ (ЭБС СамГТУ, IPRbooks и т.д.)
Основная литература		
1	Карпов, В.Е. Основы операционных систем : Курс лекций:Учеб.пособие / В.Е.Карпов,К.А.Коньков.- М., Интернет-Ун-т Информ.Технологий, 2004.- 628 с.	Электронный ресурс
2	Операционная система Linux. Курс лекций; Профобразование, 2019.- Режим доступа: <a href="https://elib.samgtu.ru/getinfo?uid=els_samgtu iprbooks 88000">https://elib.samgtu.ru/getinfo?uid=els_samgtu iprbooks 88000</a>	Электронный ресурс
3	Операционные системы. Часть 1. Операционная система Linux; Сибирский государственный университет телекоммуникаций и информатики, 2012.- Режим доступа: <a href="https://elib.samgtu.ru/getinfo?uid=els_samgtu iprbooks 40540">https://elib.samgtu.ru/getinfo?uid=els_samgtu iprbooks 40540</a>	Электронный ресурс
4	Открытые системы и Linux-технологии; Санкт-Петербургский политехнический университет Петра Великого, 2018.- Режим доступа: <a href="https://elib.samgtu.ru/getinfo?uid=els_samgtu iprbooks 83320">https://elib.samgtu.ru/getinfo?uid=els_samgtu iprbooks 83320</a>	Электронный ресурс

5	Управление процессами в операционных системах Windows и Linux; Московский государственный строительный университет, ЭБС АСВ, 2015.- Режим доступа: <a href="https://elib.samgtu.ru/getinfo?uid=els_samgtu  iprbooks  30450">https://elib.samgtu.ru/getinfo?uid=els_samgtu  iprbooks  30450</a>	Электронный ресурс
Дополнительная литература		
6	ОС Windows & ОС Linux; Российский новый университет, 2009.- Режим доступа: <a href="https://elib.samgtu.ru/getinfo?uid=els_samgtu  iprbooks  21256">https://elib.samgtu.ru/getinfo?uid=els_samgtu  iprbooks  21256</a>	Электронный ресурс
7	Администрирование ОС Linux; <b>Интернет-Университет Информационных Технологий</b> (ИНТУИТ), Ай Пи Ар Медиа, 2020.- Режим доступа: <a href="https://elib.samgtu.ru/getinfo?uid=els_samgtu  iprbooks  89414">https://elib.samgtu.ru/getinfo?uid=els_samgtu  iprbooks  89414</a>	Электронный ресурс
8	Средства безопасности операционной системы ROSA Linux; Омский государственный технический университет, 2017.- Режим доступа: <a href="https://elib.samgtu.ru/getinfo?uid=els_samgtu  iprbooks  78474">https://elib.samgtu.ru/getinfo?uid=els_samgtu  iprbooks  78474</a>	Электронный ресурс
9	Столлингс, В. Операционные системы : Внутреннее устройство и принципы проектирования:[Пер. с англ.] / В. Столлингс .- 4-е изд.- Киев, Вильямс, 2002.- 843 с.	Электронный ресурс
10	Таненбаум, Э. Современные операционные системы : пер. с англ. / Э. Таненбаум .- 3-е изд.- М., Питер, 2012.- 1115 с.	Электронный ресурс

Доступ обучающихся к ЭР НТБ СамГТУ ([elib.samgtu.ru](http://elib.samgtu.ru)) осуществляется посредством электронной информационной образовательной среды университета и сайта НТБ СамГТУ по логину и паролю.

## 6. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень программного обеспечения

При проведении лекционных занятий используется мультимедийное оборудование.

Организовано взаимодействие обучающегося и преподавателя с использованием электронной информационной образовательной среды университета.

№ п/п	Наименование	Производитель	Способ распространения
1	GNU Compiler Collection (набор компиляторов)	входит в состав Astra Linux Common Edition (Орёл) (Отечественный)	Свободно распространяемое
2	LibreOffice	входит в состав Astra Linux Common Edition (Орёл) (Отечественный)	Свободно распространяемое
3	Astra Linux Common Edition (Орёл)	ООО «РусБИТех-Астра» (Отечественный)	Свободно распространяемое

## 7. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», профессиональных баз данных, информационно-справочных систем

№ п/п	Наименование	Краткое описание	Режим доступа
-------	--------------	------------------	---------------

1	Национальный открытый университет ИНТУИТ	<a href="http://www.intuit.ru">http://www.intuit.ru</a>	Ресурсы открытого доступа
2	Проект OpenNet: портал по открытому ПО, Linux, BSD и Unix системам	<a href="http://www.opennet.ru">http://www.opennet.ru</a>	Ресурсы открытого доступа
3	СамГТУ кафедра «Прикладная математика и информатика»	<a href="http://pm.samgtu.ru/node/6">http://pm.samgtu.ru/node/6</a>	Ресурсы открытого доступа
4	Библиотека учебно-методической литературы системы "Единое окно"	<a href="http://window.edu.ru/">http://window.edu.ru/</a>	Ресурсы открытого доступа
5	eLIBRARY.ru	<a href="http://www.eLIBRARY.ru/">http://www.eLIBRARY.ru/</a>	Российские базы данных ограниченного доступа
6	Электронная библиотека изданий СамГТУ	<a href="http://irbis.samgtu.local/cgi-bin/irbis64r_01/cgiirbis_64.exe">http://irbis.samgtu.local/cgi-bin/irbis64r_01/cgiirbis_64.exe</a>	Российские базы данных ограниченного доступа

## 8. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю)

### Лекционные занятия

- лекционная аудитория каф. ПМИИ (ауд. 501/8к или аналогичная не менее чем на 25 посадочных мест), оснащенная презентационной техникой (проектор, экран, компьютер/ноутбук); комплект учебной мебели;
- комплект электронных презентаций/слайдов (при наличии).

### Лабораторные занятия

- компьютерные аудитории выпускающей кафедры «Прикладная математика и информатика» с установленным лицензионным программным обеспечением (минимальный перечень которого представлен в пункте 6 РП), доступом в Интернет.

### Самостоятельная работа

- для самостоятельной работы обучающихся предусмотрены рабочие места в читальных залах научно-технической библиотеки СамГТУ и классах ПЭВМ каф. ПМИИ и информационно-вычислительного центра СамГТУ, оснащенные компьютерами с доступом в сеть «Интернет», предназначенные для работы в электронно-информационной образовательной среде СамГТУ.

## 9. Методические материалы

### Методические рекомендации при работе на лекции

До лекции студент должен просмотреть учебно-методическую и научную литературу по теме лекции с тем, чтобы иметь представление о проблемах, которые будут разбираться в лекции.

Перед началом лекции обучающимся сообщается тема лекции, план, вопросы, подлежащие рассмотрению, доводятся основные литературные источники. Весь учебный материал, сообщаемый преподавателем, должен не просто прослушиваться. Он должен быть активно воспринят, т.е. услышан, осмыслен, понят, зафиксирован на бумаге и закреплен в памяти. Приступая к слушанию нового учебного материала, полезно мысленно установить его связь с ранее изученным. Следя за техникой чтения лекции (акцент на существенном, повышение тона, изменение ритма, пауза и т.п.), необходимо вслед за преподавателем уметь выделять основные категории, законы и определять их содержание, проблемы, предполагать их возможные решения, доказательства и выводы. Осуществляя такую работу, можно значительно облегчить себе понимание учебного материала, его конспектирование и дальнейшее изучение.

Конспектирование лекции позволяет обработать, систематизировать и лучше сохранить полученную информацию с тем, чтобы в будущем можно было восстановить в памяти основные, содержательные моменты. Типичная ошибка, совершаемая обучающимся, дословное конспектирование речи преподавателя. Как правило, при записи «слово в слово» не остается времени на обдумывание, анализ и синтез информации. Отбирая нужную информацию, главные мысли, проблемы, решения и выводы, необходимо сокращать текст, строить его таким образом, чтобы потом можно было легко в нем разобраться. Желательно оставить в рабочих конспектах поля, на которых можно будет делать пометки из рекомендованной литературы, дополняющие материал прослушанной лекции, а также подчеркивающие особую важность тех или иных теоретических положений. С окончанием лекции работа над конспектом не может считаться завершенной. Нужно еще восстановить отдельные места, проверить, все ли понятно, уточнить что-то на консультации и т.п. с тем, чтобы конспект мог быть использован в процессе подготовки к практическим занятиям, зачету, экзамену. Конспект лекции – незаменимый учебный документ, необходимый для самостоятельной работы.

### Методические рекомендации при работе на лабораторном занятии

Проведение лабораторной работы делится на две условные части: теоретическую и практическую.

Необходимыми структурными элементами занятия являются проведение лабораторной работы, проверка усвоенного материала, включающая обсуждение теоретических основ выполняемой работы.

Перед лабораторной работой, как правило, проводится технико-теоретический инструктаж по использованию необходимого оборудования. Преподаватель корректирует деятельность обучающегося в процессе выполнения работы (при необходимости). После завершения лабораторной работы подводятся итоги, обсуждаются результаты деятельности.

Возможны следующие формы организации лабораторных работ: фронтальная, групповая и индивидуальная. При фронтальной форме выполняется одна и та же работа (при этом возможны различные варианты заданий). При групповой форме работа выполняется группой (командой). При индивидуальной форме обучающимся выполняются индивидуальные работы.

По каждой лабораторной работе имеются методические указания по их выполнению, включающие необходимый теоретический и практический материал, содержащие элементы и последовательную инструкцию по проведению выбранной работы, индивидуальные варианты заданий, требования и форму отчетности по данной работе.

### Методические рекомендации по выполнению самостоятельной работы

Организация самостоятельной работы обучающихся ориентируется на активные методы овладения знаниями, развитие творческих способностей, переход от поточного к индивидуализированному обучению с учетом потребностей и возможностей обучающегося.

Самостоятельная работа с учебниками, учебными пособиями, научной, справочной литературой, материалами периодических изданий и Интернета является наиболее эффективным методом получения

дополнительных знаний, позволяет значительно активизировать процесс овладения информацией, способствует более глубокому усвоению изучаемого материала. Все новые понятия по изучаемой теме необходимо выучить наизусть и внести в глоссарий, который целесообразно вести с самого начала изучения курса.

Самостоятельная работа реализуется:

- непосредственно в процессе аудиторных занятий;
- на лекциях, практических занятиях;
- в контакте с преподавателем вне рамок расписания;
- на консультациях по учебным вопросам, в ходе творческих контактов, при ликвидации задолженностей, при выполнении индивидуальных заданий и т.д.;
- в библиотеке, дома, на кафедре при выполнении обучающимся учебных и практических задач.

Эффективным средством осуществления обучающимся самостоятельной работы является электронная информационно-образовательная среда университета, которая обеспечивает доступ к учебным планам, рабочим программам дисциплин (модулей), практик, к изданиям электронных библиотечных систем.

## **10. Фонд оценочных средств по дисциплине (модулю)**

Фонд оценочных средств представлен в приложении № 1.

**Фонд оценочных средств  
по дисциплине  
Б1.О.03.07 «Операционные системы»**

<b>Код и направление подготовки (специальность)</b>	01.03.02 Прикладная математика и информатика
<b>Направленность (профиль)</b>	Прикладная математика и информатика
<b>Квалификация</b>	Бакалавр
<b>Форма обучения</b>	Очная
<b>Год начала подготовки</b>	2022
<b>Институт / факультет</b>	Институт автоматизации и информационных технологий
<b>Выпускающая кафедра</b>	кафедра "Прикладная математика и информатика"
<b>Кафедра-разработчик</b>	кафедра "Прикладная математика и информатика"
<b>Объем дисциплины, ч. / з.е.</b>	216 / 6
<b>Форма контроля (промежуточная аттестация)</b>	Зачет с оценкой, Экзамен

## Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами освоения образовательной программы

Наименование категории (группы) компетенций	Код и наименование компетенции	Код и наименование индикатора достижения компетенции	Результаты обучения (знать, уметь, владеть, соотнесенные с индикаторами достижения компетенции)
Общепрофессиональные компетенции			
Теоретические и практические основы профессиональной деятельности	ОПК-2 Способен использовать и адаптировать существующие математические методы и системы программирования для разработки и реализации алгоритмов решения прикладных задач	ОПК-2.1 Знает существующие математические методы и системы программирования для разработки и реализации алгоритмов решения прикладных задач; методы и алгоритмы управления процессами и ресурсами операционной системы; основные принципы организации и управления памятью; основные принципы диспетчирования процессов и потоков в системах; взаимосвязь архитектурных особенностей аппаратуры ЭВМ и компонентов системного программного обеспечения; основные подходы к обеспечению безопасности функционирования операционных систем	Знать используемые методы и алгоритмы реализации системного программного обеспечения
		ОПК-2.2 Умеет разрабатывать и реализовывать алгоритмы решения прикладных задач, используя существующие математические методы, системы программирования и программные комплексы	Уметь реализовывать простейшие алгоритмы, используемые для построения системного программного обеспечения
Информационно-коммуникационные технологии для профессиональной деятельности	ОПК-4 Способен решать задачи профессиональной деятельности с использованием существующих информационно-коммуникационных технологий и с учетом основных требований информационной безопасности	ОПК-4.2 Умеет эффективно использовать современные информационные технологии для решения задач профессиональной деятельности	Уметь использовать информационные сервисы глобальных телекоммуникаций, базы данных, webресурсы, системное и программное обеспечение;

### Матрица соответствия оценочных средств запланированным результатам обучения

Код индикатора достижения компетенции	Результаты обучения	Оценочные средства	Текущий контроль успеваемости	Промежуточная аттестация
<b>Введение в операционные системы</b>				

ОПК-2.1 Знает существующие математические методы и системы программирования для разработки и реализации алгоритмов решения прикладных задач; методы и алгоритмы управления процессами и ресурсами операционной системы; основные принципы организации и управления памятью; основные принципы диспетчирования процессов и потоков в системах; взаимосвязь архитектурных особенностей аппаратуры ЭВМ и компонентов системного программного обеспечения; основные подходы к обеспечению безопасности функционирования операционных систем	<b>Знать</b> используемые методы и алгоритмы реализации системного программного обеспечения	Зачет с оценкой	Нет	Да
		выполнение лабораторных работ, курсовой проект	Да	Да
ОПК-2.2 Умеет разрабатывать и реализовывать алгоритмы решения прикладных задач, используя существующие математические методы, системы программирования и программные комплексы	<b>Уметь</b> реализовывать простейшие алгоритмы, используемые для построения системного программного обеспечения	выполнение лабораторных работ, курсовой проект	Да	Да
		Зачет с оценкой	Нет	Да
ОПК-4.2 Умеет эффективно использовать современные информационные технологии для решения задач профессиональной деятельности	<b>Уметь</b> использовать информационные сервисы глобальных телекоммуникаций, базы данных, вебресурсы, системное и программное обеспечение;	выполнение лабораторных работ, курсовой проект, устный опрос	Да	Да
		Зачет с оценкой	Нет	Да
<b>Процессы операционной системы</b>				

ОПК-2.1 Знает существующие математические методы и системы программирования для разработки и реализации алгоритмов решения прикладных задач; методы и алгоритмы управления процессами и ресурсами операционной системы; основные принципы организации и управления памятью; основные принципы диспетчирования процессов и потоков в системах; взаимосвязь архитектурных особенностей аппаратуры ЭВМ и компонентов системного программного обеспечения; основные подходы к обеспечению безопасности функционирования операционных систем	<b>Знать</b> используемые методы и алгоритмы реализации системного программного обеспечения	выполнение лабораторных работ, курсовой проект	Да	Да
		Зачет с оценкой	Нет	Да
ОПК-2.2 Умеет разрабатывать и реализовывать алгоритмы решения прикладных задач, используя существующие математические методы, системы программирования и программные комплексы	<b>Уметь</b> реализовывать простейшие алгоритмы, используемые для построения системного программного обеспечения	выполнение лабораторных работ, курсовой проект	Да	Да
		Зачет с оценкой	Нет	Да
ОПК-4.2 Умеет эффективно использовать современные информационные технологии для решения задач профессиональной деятельности	<b>Уметь</b> использовать информационные сервисы глобальных телекоммуникаций, базы данных, вебресурсы, системное и программное обеспечение;	выполнение лабораторных работ, курсовой проект, устный опрос	Да	Да
		Зачет с оценкой	Нет	Да
<b>Синхронизация процессов</b>				

ОПК-2.1 Знает существующие математические методы и системы программирования для разработки и реализации алгоритмов решения прикладных задач; методы и алгоритмы управления процессами и ресурсами операционной системы; основные принципы организации и управления памятью; основные принципы диспетчирования процессов и потоков в системах; взаимосвязь архитектурных особенностей аппаратуры ЭВМ и компонентов системного программного обеспечения; основные подходы к обеспечению безопасности функционирования операционных систем	<b>Знать</b> используемые методы и алгоритмы реализации системного программного обеспечения	выполнение лабораторных работ, курсовой проект	Да	Да
		Экзамен	Нет	Да
ОПК-2.2 Умеет разрабатывать и реализовывать алгоритмы решения прикладных задач, используя существующие математические методы, системы программирования и программные комплексы	<b>Уметь</b> реализовывать простейшие алгоритмы, используемые для построения системного программного обеспечения	выполнение лабораторных работ, курсовой проект	Да	Да
		Экзамен	Нет	Да
ОПК-4.2 Умеет эффективно использовать современные информационные технологии для решения задач профессиональной деятельности	<b>Уметь</b> использовать информационные сервисы глобальных телекоммуникаций, базы данных, вебресурсы, системное и программное обеспечение;	выполнение лабораторных работ, курсовой проект, устный опрос	Да	Да
		Экзамен	Нет	Да
<b>Управление памятью</b>				

ОПК-2.1 Знает существующие математические методы и системы программирования для разработки и реализации алгоритмов решения прикладных задач; методы и алгоритмы управления процессами и ресурсами операционной системы; основные принципы организации и управления памятью; основные принципы диспетчирования процессов и потоков в системах; взаимосвязь архитектурных особенностей аппаратуры ЭВМ и компонентов системного программного обеспечения; основные подходы к обеспечению безопасности функционирования операционных систем	<b>Знать</b> используемые методы и алгоритмы реализации системного программного обеспечения	выполнение лабораторных работ, курсовой проект	Да	Да
		Экзамен	Нет	Да
ОПК-2.2 Умеет разрабатывать и реализовывать алгоритмы решения прикладных задач, используя существующие математические методы, системы программирования и программные комплексы	<b>Уметь</b> реализовывать простейшие алгоритмы, используемые для построения системного программного обеспечения	выполнение лабораторных работ, курсовой проект	Да	Да
		Экзамен	Нет	Да
ОПК-4.2 Умеет эффективно использовать современные информационные технологии для решения задач профессиональной деятельности	<b>Уметь</b> использовать информационные сервисы глобальных телекоммуникаций, базы данных, вебресурсы, системное и программное обеспечение;	выполнение лабораторных работ, курсовой проект, устный опрос	Да	Да
		Экзамен	Нет	Да
<b>Файловая система</b>				

ОПК-2.1 Знает существующие математические методы и системы программирования для разработки и реализации алгоритмов решения прикладных задач; методы и алгоритмы управления процессами и ресурсами операционной системы; основные принципы организации и управления памятью; основные принципы диспетчирования процессов и потоков в системах; взаимосвязь архитектурных особенностей аппаратуры ЭВМ и компонентов системного программного обеспечения; основные подходы к обеспечению безопасности функционирования операционных систем	<b>Знать</b> используемые методы и алгоритмы реализации системного программного обеспечения	выполнение лабораторных работ, курсовой проект	Да	Да
		Экзамен	Нет	Да
ОПК-2.2 Умеет разрабатывать и реализовывать алгоритмы решения прикладных задач, используя существующие математические методы, системы программирования и программные комплексы	<b>Уметь</b> реализовывать простейшие алгоритмы, используемые для построения системного программного обеспечения	выполнение лабораторных работ, курсовой проект	Да	Да
		Экзамен	Нет	Да
ОПК-4.2 Умеет эффективно использовать современные информационные технологии для решения задач профессиональной деятельности	<b>Уметь</b> использовать информационные сервисы глобальных телекоммуникаций, базы данных, вебресурсы, системное и программное обеспечение;	выполнение лабораторных работ, курсовой проект, устный опрос	Да	Да
		Экзамен	Нет	Да
<b>Сети и сетевые операционные системы</b>				

ОПК-2.1 Знает существующие математические методы и системы программирования для разработки и реализации алгоритмов решения прикладных задач; методы и алгоритмы управления процессами и ресурсами операционной системы; основные принципы организации и управления памятью; основные принципы диспетчирования процессов и потоков в системах; взаимосвязь архитектурных особенностей аппаратуры ЭВМ и компонентов системного программного обеспечения; основные подходы к обеспечению безопасности функционирования операционных систем	<b>Знать</b> используемые методы и алгоритмы реализации системного программного обеспечения	выполнение лабораторных работ, курсовой проект	Да	Да
		Экзамен	Нет	Да
ОПК-2.2 Умеет разрабатывать и реализовывать алгоритмы решения прикладных задач, используя существующие математические методы, системы программирования и программные комплексы	<b>Уметь</b> реализовывать простейшие алгоритмы, используемые для построения системного программного обеспечения	выполнение лабораторных работ, курсовой проект	Да	Да
		Экзамен	Нет	Да
ОПК-4.2 Умеет эффективно использовать современные информационные технологии для решения задач профессиональной деятельности	<b>Уметь</b> использовать информационные сервисы глобальных телекоммуникаций, базы данных, вебресурсы, системное и программное обеспечение;	выполнение лабораторных работ, курсовой проект, устный опрос	Да	Да
		Экзамен	Нет	Да

**Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующие процесс формирования компетенций в ходе освоения образовательной программы**

**Устный опрос для текущего контроля успеваемости**

**Раздел «Введение в операционные системы»**

1. Что такое операционная система?
2. Что называется шиной?
3. Кто и когда разработал Ассемблер для IBM-701?
4. Что такое системный вызов?
5. Что такое прерывание?
6. В чем главная задача файловой системы?
7. Что такое исключительная ситуация?
8. Монолитное ядро – это...
9. Многозадачный режим называется вытесняющим (preemptive), когда...
10. Основное достоинство микроядерной архитектуры – это...
11. Что такое учетная запись в операционной системе Linux?
12. Что такое UID?
13. Что такое полное имя?
14. Что такое GID?
15. Что такое домашний каталог?
16. Что такое командная оболочка?
17. Что такое виртуальные консоли?
18. Что такое Сеанс работы?
19. Какие полномочия у администратора?

**Раздел «Процессы операционной системы»**

1. Что такое процесс?
2. Новый процесс, появляющийся в системе первоначально помещается в состояние...
3. Выберете одноразовые операции: Создание процесса; Запуск процесса; Блокировка процесса; Изменение состояние процесса.
4. Приостановка процесса – это ...
5. Переключение контекста – это..
6. Время, потраченное на переключение контекста, варьируется от... до...
7. Найдите лишнее в числе многократных операций: Запуск; Блокирование; Введение дополнительных условий; Разблокирование.
8. Когда один процесс выполняется, остальные...
9. Какие из перечисленных компонентов входят в регистровый контекст процесса: Информация о всех устройствах ввода-вывода; Программный счетчик процесса; Код и данные в адресном пространстве процесса.
10. Из какого состояния процесс может перейти в состояние «ожидание»?
11. Какие цели не преследует планирование?
12. Какими свойствами не должны обладать алгоритмы планирования?
13. Какие характеристики не относятся к статическим параметрам процессов: Каким пользователем запущен процесс или сформировано задание; Количество свободных ресурсов на данный момент; Насколько важной является поставленная задача, т. е. каков приоритет ее выполнения; Сколько процессорного времени запрошено пользователем для решения задачи; Какие ресурсы вычислительной системы и в каком количестве необходимы заданию.
14. Что такое невытесняющее планирование?

15. Какое из этих утверждений не относится к алгоритму First-Come, First-Served (FCFS): Процесс, получивший в свое распоряжение процессор, занимает его до истечения текущего; Осуществляет вытесняющее планирование; Среднее время ожидания и среднее полное время выполнения для этого алгоритма существенно зависят от порядка расположения процессов в очереди; Осуществляет невытесняющее планирование; Алгоритм FCFS практически неприменим для систем разделения времени – слишком большим получается среднее время отклика в интерактивных.
16. Какое из высказываний верно для алгоритма Round Robin (RR): Это вид детской карусели в Америке; Такой алгоритм реализуется с помощью организации процессов, находящихся в состоянии готовности, в очередь FIFO; Планировщик выбирает для очередного исполнения процесс, расположенный в начале очереди, и устанавливает таймер для генерации прерывания по истечении определенного кванта времени; Представляет собой режим невытесняющего планирования; Такой алгоритм реализуется с помощью организации процессов, находящихся в состоянии готовности, в очередь FIFO; Планировщик выбирает для очередного исполнения процесс, расположенный в начале очереди, и устанавливает таймер для генерации прерывания по истечении определенного кванта времени; Представляет собой режим вытесняющего планирования; Является самым быстрым алгоритмом для сортировки массивов; Является важнейшим алгоритмом интеллектуального поиска.
17. Какое из утверждений неверно для алгоритма Shortest-Job-First (SJF): Алгоритм выбирает процесс с наименьшим значением CPU burst и исполняет его и так далее. Во время исполнения процесса не происходит передача исполнения другим процессам; Алгоритм выбирает процесс с наименьшим значением CPU burst и исполняет его и так далее. Во время исполнения процесса происходит передача исполнения другим процессам; Является алгоритмом невытесняющего планирования; Является алгоритмом вытесняющего планирования; Алгоритм постоянно передает исполнение процессов, стоит минимальный квант времени.
18. Что такое гарантированное планирование?
19. Какое из утверждений верно для понятия Многоуровневые очереди (Multilevel Queue): Алгоритм, реализованный только в Unix системах; Алгоритм долгосрочного планирования; Класс алгоритмов для Unix систем; Абстрактный тип данных в программировании; Класс алгоритмов, характерной особенностью которых является фиксированный приоритет процессов.
20. Какой самый простейший алгоритм планирования?

### **Раздел «Синхронизация процессов»**

1. Что такое активность?
2. Что такое атомарная операция?
3. Что есть interleaving?
4. Когда набор активностей детерминирован?
5. Для чего нужны достаточные условия Бернштейна?
6. Что такое критическая секция?
7. Что из перечисленного не является требованием, предъявляемым к алгоритмам синхронизации: Выполнение достаточных условий Бернштейна; Не должно возникать неограниченно долгого ожидания для входа одного из процессоров в свой критический участок; В один и тот же момент времени существует не более одного процесса, выполняющегося в критической секции; Не должно существовать никаких предположений об относительных скоростях выполняющихся процессов или числе процессоров, на которых они исполняются.
8. Какому требованию не удовлетворяет использование «Переменной-замка»: Условию взаимного исключения; Алгоритм ничего не знает о скоростях выполнения процессов; Условию ограниченного ожидания; Условию прогресса.

9. Что из перечисленного ниже неверно: Алгоритм булочной используется для  $n$  процессоров; Алгоритм булочной не удовлетворяет условию ограниченного ожидания; Алгоритм Петерсона удовлетворяет всем требованиям, которые предъявляются к алгоритмам синхронизации; Алгоритм Петерсона используется для двух процессоров.
10. Что из перечисленного ниже неверно: Команды Test-and-Set и Swap существенно упрощают алгоритмы синхронизации процессов; Команды Test-and-Set и Swap выполняют ряд действий как атомарную операцию; Команда Test-and-Set обменивает два значения, находящиеся в памяти; Команда Swap обменивает два значения, находящиеся в памяти.
11. Семафор представляет собой ...
12. Через какие операции может осуществляться доступ к семафору?
13. При реализации семафоров, за что отвечают операции P и V?
14. Если при выполнении операции P над семафором заблокированными оказались несколько процессов, то ...
15. При использовании мониторов обязанность конструирования механизма взаимного исключения возложена на ...
16. Значения переменных монитора извне могут быть изменены только с помощью вызова ...
17. Что нужно предпринять для того, чтобы не оказалось двух процессов (разбудившего и пробужденного) одновременно активных внутри монитора (подход Хансена)?
18. Мониторы (механизм синхронизации) представляют собой ...
19. При реализации сообщений, для прямой и не прямой адресации достаточно ... примитивов.
20. Если запрашиваемый процессом ресурс недоступен, ОС переводит данный процесс в состояние ожидания. В случае, когда требуемый ресурс удерживается другим ожидающим процессом, первый процесс не сможет сменить свое состояние. Как называется данная ситуация?
21. Какие ресурсы являются разделяемыми?
22. Продолжи предложение. Взаимоблокировки можно избежать, если распределять ресурсы, придерживаясь определенных правил. Среди такого рода алгоритмов наиболее известен алгоритм...
23. Ситуация, когда конечный результат обработки данных зависит от соотношения скоростей потоков, называется...
24. Что такое безопасное состояние в ОС?
25. Когда возникают тупики?

### **Раздел «Управление памятью»**

1. Как называется часть ОС, которая отвечает за управление памятью?
2. Что представляет собой основная память?
3. Что обычно НЕ содержат сегменты?
4. Свопинг – это ...
5. Совокупность всех логических адресов называется логическим (виртуальным) ...
6. Чем обусловлена эффективность иерархической схемы памяти?
7. Как называются адреса в основной памяти, характеризующие реальное расположение данных в физической памяти?
8. С помощью какой стратегии процесс помещается в тот раздел, где после его загрузки останется меньше всего свободного места?
9. На каких(ом) этапах(е) может осуществляться связывание адресов?
10. В отличие от ..., где линейный адрес преобразован в двумерный операционной системой для удобства отображения, в сегментной организации двумерность адреса является следствием представления пользователя о процессе не в виде линейного массива байтов, а как набор сегментов переменной размера (данные, код, стек...).

11. Информация, с которой работает активный процесс, должна располагаться...
12. Какой объем оперативной памяти мог иметь 16-разрядный компьютер PDP-11/70 с 64 Кбайт логической памяти?
13. Для организации какой памяти пригодна любая из трех схем управления памятью – страничная, сегментная и сегментно-страничная?
14. Как в целом называется система поддержки страничной виртуальной памяти?
15. В популярных ОС Linux и Windows 2000 все сегменты процесса, а также область памяти ядра ограничены виртуальным адресным пространством объемом... GB
16. Как можно назвать запись в таблице страниц, которая содержит информацию об атрибутах страницы?
17. Как называется небольшая, быстрая кэш-память, хранящая необходимую на данный момент часть таблицы страниц?
18. Сколько записей на каждый страничный кадр физической памяти содержится в инвертированной таблице страниц?
19. Размер страницы – это степень какого числа?
20. Какой существенный минус имеет применение инвертированной таблицы?
21. Отображение виртуального адреса осуществляется при помощи...
22. Что происходит с виртуальной страницей, когда нужной страницы в памяти нет или операция обращения к памяти недопустима?
23. Строковое обращение – это...
24. Какие два флага включают аппаратные средства?
25. Трешинг – это ...
26. Набор страниц, активно использующихся вместе, который позволяет процессу в момент времени в течение некоторого периода производительно работать, избегая большого количества страничных нарушений называется...
27. Алгоритмы, обеспечивающие поддержку системы в состоянии отсутствия трешинга, реализованы в составе ...
28. Чаще всего виртуальная память процесса ОС разбивается на ...
29. Сегмент в ОС – это ...

#### **Раздел «Файловая система»**

1. Файловая система – это часть операционной системы, назначение которой состоит в том, чтобы ...
2. Какая из функций НЕ является функцией файловой системы?
3. Какая длина имен установлена в соответствии со стандартом POSIX?
4. Что содержат регулярные файлы?
5. Что не является атрибутом файла?
6. Файл прямого доступа – это ...
7. Система управления файлами НЕ берет на себя ответственность за ...
8. Считывание очередного байта осуществляется с так называемой ... позиции, которая характеризуется смещением от начала файла.
9. Какой длины в Linux могут быть имена файлов и каталогов?
10. Файл – это ...
11. Файловая система – это...
12. Каталог в файловой системе — это ...
13. Какой каталог называется «текущим каталогом»?
14. “./home/method/examples” – это ...
15. “/home/method/examples” – это ...
16. При удалении символической ссылки на файл ...
17. Уникальным идентификатором файла в файловой системе является ...
18. Домашний каталог – это ...
19. При помощи утилиты mkdir можно ...

20. При помощи утилиты ls можно ...
21. Деревом каталогов в файловой системе Linux называется ...

### **Раздел «Сети и сетевые операционные системы»**

1. Какой уровень в слоеных сетевых вычислительных системах, является самым нижним?
2. Что такое сетевой протокол?
3. Что такое сокет?
4. Какую функцию выполняет сетевой уровень модели ISO?
5. Почему обычно назначается несколько серверов DNS?
6. Как формируется маршрут передачи пакета по сети при маршрутизации от источника данных?
7. Как формируется маршрут передачи пакета по сети при одношаговой маршрутизации?
8. Что необходимо обеспечить для корректной работы сетевых средств связи?
9. Для чего компьютеры объединяют в сеть?
10. Какие имена узлов сети используются для удаленной адресации?
11. Что такое информационная безопасность?
12. Чем реализуются технические средства для обеспечения информационной безопасности?
13. Целостность системы подразумевает, что...
14. Угроза безопасности – это ...
15. Атака – это ...
16. Какой вид системы обеспечивает уверенность в том, что секретные данные будут доступны только тем пользователям, которым этот доступ разрешен?
17. Какая наиболее распространенная угроза?
18. Процесс преобразования сообщения из открытого текста в шифротекст – это ...
19. Процесс преобразования зашифрованного сообщения в открытый текст – это ...
20. Открытый (несекретный) ключ используется для ...
21. Что такое аутентификация?
22. Что такое идентификация?
23. Что такое авторизация?
24. Что может быть идентификатором в обычной жизни (примеры)?
25. Что является авторизацией?
26. Какие события входят в аудит системы защиты?

### **Лабораторные работы для текущего контроля успеваемости**

## Указания к лабораторным работам 1

# Знакомство с операционной системой Linux

**Дидактические единицы.** Системные вызовы и библиотека `libc`. Понятия `login` и `password`. Упрощенное понятие об устройстве файловой системы в Linux. Перенаправление ввода и вывода. Простейшие команды для работы с файлами. Команды `chown`, `chgrp`, `umask`. Права доступа к файлу. Системные вызовы `getuid` и `getgid`. Компиляция программ на языке C в Linux и запуск их на выполнение.

### 1.1. Цель работы

Лабораторная работа знакомит студентов с операционной системой Linux и основами её использования. Студенту даются основные понятия ОС Linux. Студент знакомится с компиляцией программ на языке C в Linux и их запуском на выполнение.

Для выполнения лабораторной работы будет полезно (но необязательно) знание английского языка на уровне чтения технических текстов.

### 1.2. Порядок выполнения работы (задания)

1. Осуществите свой первый вход в операционную систему:
  - скачайте и запустите portable-версию ssh-клиента PuTTY;
  - войдите в систему с выданными учетными данными по ssh на ip `194.190.143.48` или по доменному имени `stud.pmi.samgtu.ru`<sup>1</sup> (пароль при наборе не отображается);
  - при первом входе система запросит смену пароля (необходимо один раз ввести выданный пароль и два раза – новый<sup>2</sup>; пароли при наборе не отображаются);
  - войдите в систему, используя новый пароль.
2. Регистрация на сайте INTUIT и запись на курс «Основы операционных систем. Практикум»:<sup>3</sup>
  - зарегистрируйтесь на сайте INTUIT (<http://www.intuit.ru>);

---

<sup>1</sup>При первом входе примите согласие на прием ssh-сертификата.

<sup>2</sup>Не забудьте его!

<sup>3</sup>Задания 2 и 3 можно выполнять параллельно, так как материал курсов взаимодополняем.

- запишитесь на курс «Основы операционных систем. Практикум» (<https://www.intuit.ru/studies/courses/2249/52/info>)
  - ознакомьтесь с материалом лекции 1 «Введение в курс практических занятий. Знакомство с операционной системой UNIX»;
  - пройдите Тест 1.
3. Регистрация на сайте STEPİK<sup>4</sup> и запись на курс «Введение в Linux»:
- зарегистрируйтесь на сайте STEPİK (<https://stepik.org/>);
  - запишитесь на курс «Введение в Linux» (<https://stepik.org/course/73/syllabus>);
  - ознакомьтесь с материалом разделов 1.4–1.9;
  - выполните все задания указанных разделов.
4. Разберитесь, для чего используются каталоги<sup>5</sup> /bin, /boot, /dev, /etc, /home, /lib, /lost+found, /media, /mnt, /opt, /proc, /root, /sbin, /srv, /sys, /tmp, /usr, /var/log.
5. Разберитесь, что делают команды man, pwd, cd, ls, cat, cp, rm, mkdir, mv. Попробуйте их в действии.<sup>6</sup>
6. Создайте в домашней директории файл list.txt, содержащий список файлов и каталогов из корня.
7. Установите зависимость и правила перехода от числовой маски прав к символьной в формате ugo.
8. Создайте в своем домашнем каталоге непустой каталог (15–20 файлов). Затем с помощью команды chmod дайте права rwx для группы на все файлы в этом каталоге (рекурсивно).
9. Наберите, откомпилируйте и запустите программу «Hello, World!»:

```
/* hello.c */
#include <stdio.h>
int main(void)
{
    printf("Hello, World!\n");
    printf("Привет, Мир!\n");
    return 0;
}
```

---

<sup>4</sup>Изучение материалов сайта STEPİK осуществляется студентом по желанию.

<sup>5</sup>Можно найти в интернете много информации о каждом из этих каталогов.

<sup>6</sup>Не обязательно читать man'ы. Можно также поgoogle'ть.

Для набора программы используйте программу `nano`. Для компиляции используйте команду `gcc hello.c -o hello`.

10. Напишите, откомпилируйте и запустите программу, которая печатала бы идентификатор пользователя, запустившего программу, и идентификатор его группы.

### 1.3. Порядок подготовки и сдачи письменного отчета

Отчет должен быть подготовлен средствами  $\text{\LaTeX}$ , например в системе `Overleaf`. Можно использовать шаблон <https://www.overleaf.com/latex/templates/shablon-dlia-laboratornykh-rabot/sqvxbnhgxxdm>.

Отчет должен описывать содержание работы студента при выполнении заданий 4–10:

- Для чего предназначены каталоги, перечисленные в задании 4 (ответ оформите в виде таблицы)?
- Что делают команды, перечисленные в задании 5? Приведите примеры выполнения для одной-двух команд (скриншот(ы) терминального окна после их выполнения) на выбор.
- Приведите команду, с помощью которой можно выполнить задание 6.
- Приведите таблицы переходов от числовых значений маски прав к формату `ugo`.
- Приведите команду, с помощью которой рекурсивно изменяли права. Приведите скриншоты вывода команды `ls -al` до и после приведенной команды.
- Приведите скриншот выполнения программы «Hello, World!».
- Приведите листинг программы для задания 9 и скриншот её выполнения.

#### Структура архива отчета:

- скриншот, подтверждающий прохождения теста из задания 2 (должны быть видны имя и отчество студента);
- скриншот, подтверждающий выполнение задания 3;
- отчет о проделанной работе в формате `pdf`;
- `tar.gz`-архив с исходным кодом программы, созданной при выполнении задания 10.

## Указания к лабораторным работам 2

# Командная оболочка `bash`

**Дидактические единицы.** Основы скриптов. Переменные в скриптах. Переменные и вывод команд. Параметры скрипта. Управляющие конструкции в скриптах. Циклы в скриптах. Внутренние команды. Внешние команды, программы и утилиты. Команды системного администрирования. Подстановка команд. Арифметические подстановки. Перенаправление ввода/вывода.

### 2.1. Цель работы

Лабораторная работа знакомит студентов с мощным инструментом — командной оболочкой `bash`.

Студент пишет свои первые скрипты.

### 2.2. Порядок выполнения работы (задания)

1. Изучите материал «Написание скриптов на Bash» <https://losst.ru/napisanie-skriptov-na-bash>.
2. Автоматизируйте с помощью скриптов выполнение следующих двух задач из предыдущего занятия:
  - (a) Создайте в домашней директории файл `list2.txt`, содержащий список файлов и каталогов из корня.
  - (b) Создайте в своем домашнем каталоге непустой каталог (15–20 файлов). Затем с помощью команды `chmod` дайте права `rx` для группы на все файлы в этом каталоге (рекурсивно).
3. На сайте <https://stepik.org> записаться на курс «Основы программирования для Linux» (<https://stepik.org/course/548/syllabus>). Изучить темы «Введение в разработку для Linux» (1.1–1.3) и «Файлы и файловые системы» (2.1–2.5). Выполнить все задания по этим темам.

### 2.3. Порядок подготовки и сдачи письменного отчета

Отчет должен описывать содержание работы студента при выполнении заданий 2:

- Содержать листинг скрипта для выполнения (a).

- Содержать листинг скрипта для выполнения (b).

**Структура архива отчета:**

- скриншоты, подтверждающие выполнение задания 3 (факт выполнения всех заданий разделов «Введение в разработку для Linux» (1.1–1.3) и «Файлы и файловые системы» (2.1–2.5));
- отчет о проделанной работе в формате pdf;
- tar.gz-архив с исходными кодами листингов.

## Указания к лабораторным работам 3

# Введение в курс программирования под Linux

**Дидактические единицы.** Редактор nano и его настройки для подсветки синтаксиса. Проект Hello World. Мультифайловое программирование. Автоматическая сборка. Модель клиент-интерфейс-сервер (КИС). Библиотеки. Сборка проектов со статическими библиотеками. Сборка проектов с совместно используемыми библиотеками.

### 3.1. Цель работы

Лабораторная работа знакомит студентов с основами и существующими моделями программирования на языке C в ОС Linux. Студенты приобретают основные навыки программирования на языке C в ОС Linux и автоматической сборки проектов с помощью утилиты `make`.

### 3.2. Порядок выполнения работы и задания

1. Ознакомиться с дополнительным материалом к лабораторной работе из файлов `lesson02-add.pdf` и `lesson02.tar.gz`.
2. Напишите, программу, которая печатает идентификатор пользователя, запустившего программу, и идентификатор его группы с использованием рассмотренных в дополнительном материале подходов. Исходные коды программ сохраняйте в соответствующих каталогах: `printid.multi`, `printid.csi`, `printid.static`, `printid.dinamic`. Измените соответствующим образом `Makefile`'ы.
3. Исходные коды программ вместе с `Makefile`'ами (каждый проект отдельно) заархивируйте в архив `tar.gz`. Для этого обязательно используйте способ передачи информации каналами.

### 3.3. Порядок подготовки и сдачи письменного отчета

Отчет должен быть подготовлен средствами `LaTeX`, например в системе `Overleaf`. Можно использовать шаблон <https://www.overleaf.com/latex/templates/shablon-dlia-laboratorykh-rabot/sqvxbnhgxxdm>.

Отчет должен описывать содержание работы студента при выполнении задания 2:

- Содержать листинги всех файлов проектов `printid.multi`, `printid.csi`, `printid.static`, `printid.dinamic`, включая `Makefile`'ы.

Выполненную работу (`tar.gz`-архив с приведенной ниже структурой) вкладываем в соответствующее занятие в «Систему сопровождения дистанционного обучения» до **27 октября, 23:55**.

#### Структура архива:

- отчет о проделанной работе в формате `pdf`;
- `tar.gz`-архив с исходными кодами проектов `printid.multi`, `printid.csi`, `printid.static`, `printid.dinamic`, включая `Makefile`'ы.

## Указания к лабораторным работам 4

# Процессы в операционной системе Linux (Часть 1)

**Дидактические единицы.** Процессы в ОС Linux. Понятие процесса в ОС Linux, его контекст. Идентификация процесса. Состояния процесса. Диаграмма состояний. Иерархия процессов. Системные вызовы `getpid()`, `getppid()`. Создание процесса в Linux. Системный вызов `fork()`. Завершение процесса. Функция `exit()`.

### 4.1. Цель работы

Лабораторная работа знакомит студентов со следующими понятиями: процесс в UNIX, его контекст; идентификация процесса; состояния процесса; диаграмма состояний; иерархия процессов. Студенты учатся работать с системными вызовами `getpid()`, `getppid()`, `fork()`, `exec()`.

### 4.2. Порядок выполнения работы и задания<sup>1</sup>

1. Изучить в курсе «**Основы операционных систем. Практикум**» (<http://www.intuit.ru/studies/courses/2249/52/info>) Лекцию 2 «**Понятие процесса в UNIX. Его контекст**». Пройдите **Тест 2**.
2. Ознакомьтесь с описанием системных вызовов `getpid()`, `getppid()` и самостоятельно напишите программу, печатающую значения PID и PPID для текущего процесса. Запустите ее несколько раз подряд. Посмотрите, как меняется идентификатор текущего процесса. Объясните наблюдаемые изменения. Определите, какой процесс является родительским для запускаемого процесса?
3. Реализуйте программу из задания 2 в виде проекта с использованием модели КИС со статическими библиотеками.
4. Ознакомьтесь с описанием системного вызова `fork()`. Наберите программу 03-1<sup>2</sup> (Листинг 3.1. **Программа 03-1.c – пример создания нового процесса с одинаковой работой процессов ребенка и родителя**), откомпилируйте ее и запустите на исполнение. Проанализируйте полученный результат.

---

<sup>1</sup>Каждое задание рекомендуется выполнять в отдельной директории. Например, задание 2 может находиться в директории `lesson4task2`. Такой подход позволит ускорить подготовку отчета и не запутаться в массе создаваемых файлов.

<sup>2</sup>При необходимости исправьте ошибки.

5. Ознакомьтесь со схемой использования системного вызова `fork()` для случая, когда родитель и ребенок выполняют разные действия. Напишите программу, чтобы родитель и ребенок совершали разные действия (какие — не важно). Изучите описание функции `exit()`. В коде Вашей программы необходимо воспользоваться этой функцией. В каком месте программы её следует использовать?

### 4.3. Порядок подготовки и сдачи письменного отчета

Отчет должен описывать содержание работы студента при выполнении заданий:

- Содержать листинг программы из задания 2 и результаты её работы.
- Содержать листинги всех файлов проекта из задания 3, включая `Makefile`'ы.
- Содержать листинг программы из задания 4 и результаты её работы.
- Содержать листинг программы из задания 5 и результаты её работы.

#### Структура архива отчета:

- скриншот, подтверждающий прохождения теста из задания 1 (должны быть видны имя и отчество студента);
- отчет о проделанной работе в формате `pdf`;
- `tar.gz`-архив с исходным кодом программ, созданных при выполнении заданий 2, 3, 4, 5 (директории `lesson4task2`, `lesson4task3`, `lesson4task4` и `lesson4task5`).

## Указания к лабораторным работам 5

# Процессы в операционной системе Linux (Часть 2)

### 5.1. Цель работы

Лабораторная работа знакомит студентов со следующими понятиями: процесс в UNIX, его контекст; идентификация процесса; состояния процесса; диаграмма состояний; иерархия процессов. Студенты учатся работать с системными вызовами `getpid()`, `getppid()`, `fork()`, `exec()`; учатся работать с переменными среды и аргументами командной строки; учатся изменять пользовательский контекст процесса.

### 5.2. Порядок выполнения работы и задания

1. Изучите в курсе «[Основы операционных систем. Практикум](http://www.intuit.ru/studies/courses/2249/52/info)» (<http://www.intuit.ru/studies/courses/2249/52/info>) прототип функции `main()`. Напишите программу, распечатывающую значения *аргументов командной строки* и *параметров окружающей среды* для текущего процесса.
2. Изучите информацию о семействе функций для системного вызова `exec()`. Откомпилируйте и запустите на исполнение программу 03-2 (Листинг 3.2 [Программа 03-2.c, изменяющая пользовательский контекст процесса](#)). Что произойдёт, если переместить или переименовать файл `03-2.c`? Проанализируйте результат.
3. Модифицируйте программу из задания 5 предыдущего занятия так, чтобы порожденный процесс с помощью системного вызова `exec()` запускал на исполнение новую (любую) внешнюю программу.
4. На сайте <https://stepik.org> на курсе «[Основы программирования для Linux](https://stepik.org/course/Основы-программирования-для-Linux-548/)» (<https://stepik.org/course/Основы-программирования-для-Linux-548/>) изучить тему «Процессы и потоки». Выполнить все задания по этой теме.

### 5.3. Порядок подготовки и сдачи письменного отчета

Отчет должен описывать содержание работы студента при выполнении заданий:

- Содержать листинг программы из задания 1 и результаты её работы.

- Содержать листинг программы из задания 2 и результаты её работы. Проанализируйте результаты работы.
- Содержать листинг программы из задания 3 и результаты её работы.

**Структура архива отчета:**

- скриншот(ы), подтверждающий прохождение теста из задания 4 (факт выполнения всех заданий темы «Процессы и потоки»);
- отчет о проделанной работе в формате pdf;
- tar.gz-архив с исходным кодом программ, созданных при выполнении заданий 1, 2, 3 (директории lesson5task1, lesson5task2, lesson5task3).

## Указания к лабораторным работам 6

# Процессы в операционной системе Linux (Часть 3). Реализация генеалогического дерева процессов из задания курсового проекта

### 6.1. Цель работы

С помощью сведений и навыков, полученных и приобретенных при выполнении лабораторных работ 4, 5, реализовать генеалогическое дерево процессов из задания курсового проекта.

### 6.2. Порядок выполнения работы и задания

1. Ознакомиться с заданием на курсовой проект <https://clck.ru/JpCeD>. Найти свой вариант, который совпадает с порядковым номером студента в группе.
2. Реализовать генеалогическое дерево процессов своего варианта. В качестве образца можно воспользоваться проектом из архива, доступного по ссылке <https://clck.ru/JpCpr>. Код одного полностью реализованного варианта и его результаты выполнения можете посмотреть в архиве <https://clck.ru/JpCzE>.

### 6.3. Порядок подготовки и сдачи письменного отчета

1. Оформить выполнение работы в качестве первого подраздела «Реализация генеалогического дерева процессов» в основном разделе курсового проекта «Разработка алгоритмов функционирования и структур данных».

#### Структура архива отчета:

- отчет о проделанной работе в формате pdf (первого подраздел курсового проекта);
- tar.gz-архив с текущим исходным кодом курсового проекта.

## Указания к лабораторным работам 7

# Организация взаимодействия процессов в ОС Linux. Часть 1: Системные вызовы системы ввода-вывода

### 7.1. Цель работы

Лабораторная работа знакомит студентов с понятием потока ввода-вывода, файлового дескриптора; даёт представление о работе с файлами через системные вызовы и стандартную библиотеку ввода-вывода с помощью системных вызовов `open()`, `close()`, `read()`, `write()`.

### 7.2. Порядок выполнения работы и задания

1. Изучить в курсе «**Основы операционных систем. Практикум**» (<http://www.intuit.ru/studies/courses/2249/52/info>) **Лекцию 3**. После изучения лекции необходимо пройти **Тест 3**.
2. Изучите информацию о том, как использовать системные вызовы `read()`, `write()`, `close()`. Наберите, откомпилируйте программу `05-1.c` (**Листинг 5.1**), иллюстрирующую использование этих системных вызовов, и запустите ее на исполнение. Модифицируйте программу так, чтобы в файл записывалась Ваше ФИО. Назовём эту программу `05-1mod.c`.
3. Напишите программу, чтобы она читала записанную программой `05-1mod.c` информацию из файла и печатала ее на экране.
4. Реализуйте программу из предыдущего задания с использованием модели КИС со статическими библиотеками.

### 7.3. Порядок подготовки и сдачи письменного отчета

Отчет должен описывать содержание работы студента при выполнении заданий:

- Содержать листинги программ из заданий **2**, **3** и результаты их работы.
- Содержать листинги всех файлов проекта из задания **4**, включая `Makefile`'ы, и результаты его работы.

**Структура архива отчета:**

- скриншот, подтверждающий прохождения теста из задания 1 (должны быть видны имя и отчество студента);
- отчет о проделанной работе в формате pdf;
- tar.gz-архив с исходным кодом программ, созданных при выполнении заданий 2, 3, 4 (директории lesson7task2, lesson7task3 и lesson7task4).

## Указания к лабораторным работам 8

# Организация взаимодействия процессов в ОС Linux. Часть 2: Работа с pipe и FIFO

### 8.1. Цель работы

Для организации взаимодействия процессов предполагается использование pipe'a и FIFO с помощью системных вызовов `pipe()` и `mknod()`. рассматриваются некоторые особенности вызовов `fork()`, `exec()`, `read()`, `write()`, `open()` при работе с взаимодействующими процессами.

### 8.2. Порядок выполнения работы и задания

1. Изучите информацию по использованию pipe. Наберите, откомпилируйте программу 05-2.c (Листинг 5.2), иллюстрирующую работу с pipe'ом, и запустите ее на исполнение. Модифицируйте программу так, чтобы через pipe передавалось Ваше ФИО. Назовём эту программу 05-2mod.c.
2. Напишите программу, реализующую двунаправленный pipe между родственными процессами (см программу 05-3.c (Листинг 5.3)): в одну сторону передаётся Ваше ФИО, в другую сторону — аббревиатура Вашей группы.
3. Изучите информацию по использованию FIFO. Наберите, откомпилируйте программу 05-4.c (Листинг 5.4), иллюстрирующую работу с FIFO между родственными процессами, и запустите ее на исполнение. Модифицируйте программу так, чтобы через FIFO передавалось Ваше ФИО. Назовём эту программу 05-4mod.c.
4. Напишите на базе предыдущего примера две программы, одна из которых пишет информацию (Ваше ФИО) в FIFO, а вторая — читает из него.

### 8.3. Порядок подготовки и сдачи письменного отчета

Отчет должен описывать содержание работы студента при выполнении заданий:

- Содержать листинги программ из заданий 1-4 и результаты их работы.

**Структура архива отчета:**

- отчет о проделанной работе в формате pdf;
- tar.gz-архив с исходным кодом программ, созданных при выполнении заданий 1–4 (директории lesson8task1, lesson8task2, lesson8task3 и lesson8task4).

## Указания к лабораторным работам 9

# Организация взаимодействия процессов в ОС Linux. Часть 3: Взаимодействие процессов через FIFO

### 9.1. Цель работы

С помощью сведений и навыков, полученных и приобретенных при выполнении предыдущих лабораторных работ, необходимо реализовать взаимодействие процессов через FIFO с выполнением ими «специализированных действий» из курсового проекта.

### 9.2. Порядок выполнения работы и задания

«Специализированные действия» должны быть реализованы в отдельных файлах с подходом мультифайлового программирования в виде «библиотек»<sup>1</sup>, функции которых будут использоваться в основном коде процессов. Механизмы взаимодействия процессов должны быть реализованы в двух файлах — `first.c` и `second.c`. `main` должен содержаться в `first.c`. Сборку осуществлять с помощью `make` со статическими библиотеками.

#### Варианты 1, 9, 17, 25, 31, 33

1. `First` получает со стандартного потока ввода список файлов каталога, и выводит их на экран, добавляя перед каждым именем порядковый номер; после этого общее количество полученных файлов `first` передаёт `second`'у (через FIFO). `Second` полученное количество возводит в квадрат и выводит значение в консоль.
2. `Second` получает со стандартного потока ввода список файлов каталога, и выводит их на экран, добавляя перед каждым именем порядковый номер; после этого общее количество полученных файлов `second` передаёт `first`'у (через FIFO). `First` полученное количество возводит в квадрат и выводит значение в консоль.

---

<sup>1</sup>Не стоит все «лепить» в один большой файл!!!, в котором описываются и механизмы взаимодействия процессов и их специфические действия. **Divide et impera!!!**

**Варианты 2, 7, 10, 18, 26, 32, 34**

1. **First** получает со стандартного потока ввода права доступа к файлам каталога, и выводит их на экран, подменяя все группы прав **rwX** словом **Все**, а через **FIFO** передаёт **second**'у в неизменном виде. **Second** в полученных строках заменяет группы прав --- на **Нет** и выводит строку в консоль.
2. **Second** получает со стандартного потока ввода права доступа к файлам каталога, и выводит их на экран, подменяя все группы прав **rwX** словом **Все**, а через **FIFO** передаёт **first**'у в неизменном виде. **First** в полученных строках заменяет группы прав --- на **Нет** и выводит строку в консоль.

**Варианты 3, 11, 15, 19, 24, 27, 35**

1. **First** получает со стандартного потока ввода любое руководство, и передает в **FIFO** заголовки всех секций данного руководства. **Second** их оттуда забирает и выводит на экран (в консоль), добавляя к каждой строке порядковый номер.
2. **Second** получает со стандартного потока ввода любое руководство, и передает в **FIFO** заголовки всех секций данного руководства. **First** их оттуда забирает и выводит на экран (в консоль), добавляя к каждой строке порядковый номер.

**Варианты 4, 12, 16, 20, 28, 36, 40**

1. **First** получает со стандартного потока информацию о файлах каталога, и, во-первых, выводит на экран (в консоль) строки о тех из них, у которых установлен бит запуска владельцем; во-вторых, при помощи **FIFO** передает размер каждого из таких файлов **second**'у, который суммирует значения и выводит полученное значение на экран.
2. **Second** получает со стандартного потока информацию о файлах каталога, и, во-первых, выводит на экран (в консоль) строки о тех из них, у которых установлен бит запуска владельцем; во-вторых, при помощи **FIFO** передает размер каждого из таких файлов **first**'у, который суммирует значения и выводит полученное значение на экран.

**Варианты 5, 8, 13, 21, 29, 37, 39**

1. **First** получает со стандартного потока ввода список всех активных процессов, и, во-первых, выводит на экран (в консоль) процессы только с четными **PID**; во-вторых, при помощи **FIFO** передает каждый из этих номеров **second**'у, который суммирует значения и выводит полученное значение на экран.

2. **Second** получает со стандартного потока ввода список всех активных процессов, и, во-первых, выводит на экран (в консоль) процессы только с четными PID; во-вторых, при помощи **FIFO** передает каждый из этих номеров **first**'у, который суммирует значения и выводит полученное значение на экран.

### Варианты 6, 14, 22, 23, 30, 38

1. **First** получает со стандартного потока ввода содержимое любого текстового файла и выводит на экран (в консоль) те его строки, которые начинаются с цифры, остальные строки передавая через **FIFO second**'у. **Second** заменяет в полученных строках X на Y и выводит результирующие строки на экран (в консоль).
2. **Second** получает со стандартного потока ввода содержимое любого текстового файла и выводит на экран (в консоль) те его строки, которые начинаются с цифры, остальные строки передавая через **FIFO first**'у. **First** заменяет в полученных строках X на Y и выводит результирующие строки на экран (в консоль).

## 9.3. Порядок подготовки и сдачи письменного отчета

Отчет должен описывать содержание работы студента при выполнении заданий:

- Содержать листинги программ из заданий своего варианта, включая **Makefile**'ы, и результаты его работы.

### Структура архива отчета:

- отчет о проделанной работе в формате **pdf**;
- **tar.gz**-архив с исходным кодом проекта, созданного при выполнении задания (директория **lesson9taskV**, где **V** — номер варианта).

## Указания к лабораторным работам 10

# Организация работы с разделяемой памятью в ОС Linux

### 10.1. Цель работы

Лабораторная работа знакомит студентов с понятием разделяемая память, даёт представление о работе с системными вызовами `shmget()`, `shmat()`, `shmdt()`; рассматриваются команды `ipcs` и `ipcrm` и некоторые другие команды и системные вызовы.

### 10.2. Порядок выполнения работы и задания

1. Изучить в курсе «**Основы операционных систем. Практикум**» (<http://www.intuit.ru/studies/courses/2249/52/info>) **Лекцию 4**. После изучения лекции необходимо пройти **Тест 4**.
2. Изучите информацию о том, как использовать системные вызовы `shmget()`, `shmat()`, `shmdt()`.

Наберите программы (**Листинг 6.1a** и **Листинг 6.1b**), сохраните под именами `06-1a.c` и `06-1b.c` соответственно, откомпилируйте их и запустите несколько раз. Проанализируйте полученные результаты. Что произойдет, если удалить или переместить файл `06-1a.c`?

3. Изучите синтаксис команд `ipcs` и `ipcrm`. Удалите созданный Вами при выполнении задания **2** сегмент разделяемой памяти из операционной системы (если было создано более одного сегмента, например, при некорректных запусках программы, то удалите их все). По каким критериям Вы определили, что это Ваш сегмент разделяемой памяти?
4. Изучите информацию о том, как использовать системные вызовы `shmctl()`. Напишите программу для удаления созданного вами при выполнении задания **2** сегмента разделяемой памяти из операционной системы.
5. Для закрепления полученных знаний напишите две программы, осуществляющие взаимодействие через разделяемую память. Первая программа должна создавать сегмент разделяемой памяти и копировать туда собственный исходный текст, вторая программа должна брать оттуда этот текст, печатать его на экране и удалять сегмент разделяемой памяти из системы.

**Облегченный вариант задания 5.** Первая программа должна создавать сегмент разделяемой памяти и копировать туда строчку из файла `myfile.txt`.<sup>1</sup> Вторая программа должна брать из памяти эту строчку, печатать её на экране и удалять сегмент разделяемой памяти из системы.

### 10.3. Порядок подготовки и сдачи письменного отчета

Отчет должен описывать содержание работы студента при выполнении заданий:

- Содержать листинги программ из заданий 2, 4, 5, и результаты его работы.
- Содержать скриншоты выполнения заданий 1 (прохождение Теста 4) и 3 (выполнение соответствующих команд)

#### Структура архива отчета:

- отчет о проделанной работе в формате `pdf`;
- `tar.gz`-архив с исходным кодом проекта, созданного при выполнении заданий (директории `lesson10task2`, `lesson10task4`, `lesson10task5`).

---

<sup>1</sup>Файл `myfile.txt` содержит фамилию, имя и отчество, студента выполняющего лабораторную работу, преподаватель может изменить эту строчку по своему усмотрению. Не смотря на это обе программы должны работать корректно.

## Указания к лабораторным работам 11

# Синхронизации процессов с помощью семафоров

### 11.1. Цель работы

Лабораторная работа знакомит студентов с понятием семафоры и основными принципами работы с ними, даёт представление о работе с системными вызовами `semget()`, `semop()`, `semctl()`.

### 11.2. Порядок выполнения работы и задания

1. Изучить в курсе «[Основы операционных систем. Практикум](http://www.intuit.ru/studies/courses/2249/52/info)» (<http://www.intuit.ru/studies/courses/2249/52/info>) Лекцию 5. После изучения лекции необходимо пройти Тест 5.
2. Изучите информацию о том, как использовать системные вызовы `semget()`, `semop()`, `semctl()`.

Наберите программы ([Листинг 8.1a](#) и [Листинг 8.1b](#)), сохраните под именами `08-1a.c` и `08-1b.c` соответственно, откомпилируйте их и запустите несколько раз. Проанализируйте полученные результаты. Что произойдет, если удалить или переместить файл `08-1a.c`?

В отчёте приведите информацию об изученных системных вызовах, приведите выводы программ.

3. Измените программы из предыдущего пункта (сохраните их под именами `08-1a_mod.c` и `08-1b_mod.c`) так, чтобы первая программа могла работать без блокировки после не менее 5 запусков второй программы. В отчете приведите коды программ.
4. Изучите поведение программ [Листинг 6.3a](#) и [Листинг 6.3b](#), которые иллюстрируют некорректную работу через разделяемую память. Обеспечьте с помощью семафоров взаимного исключения для их правильной работы (назовите программы `06-3a_mod.c` и `06-3b_mod.c`). В отчете приведите коды программ и выводы программ.
5. Организуйте двустороннюю поочередную связь процесса-родителя и процесса-ребенка через `pipe`, используя для синхронизации семафоры, модифицировав программу [Листинг 5.3](#). В отчете приведите код программы.

## 11.3. Порядок подготовки и сдачи письменного отчета

Отчет должен описывать содержание работ студента при выполнении заданий **2–5**.

### Структура архива отчета:

- скриншот, подтверждающий прохождения теста (должны быть видны имя и отчество студента);
- отчет о проделанной работе в формате **pdf**;
- директория `lesson11task2`, в которой размещается код задания п. **2**;
- директория `lesson11task3`, в которой размещается код задания п. **3**;
- директория `lesson11task4`, в которой размещается код задания п. **4**;
- директория `lesson11task5`, в которой размещается код задания п. **5**;

## Указания к лабораторным работам 12

# Синхронизации процессов с помощью сообщений

### 12.1. Цель работы

Лабораторная работа знакомит студентов с очередями сообщений и основными принципами работы с ними, даёт представление о работе с системными вызовами `msgget()`, `msgsnd()`, `msgrcv()` и `msgctl()`.

### 12.2. Порядок выполнения работы и задания

1. Изучить в курсе «[Основы операционных систем. Практикум](http://www.intuit.ru/studies/courses/2249/52/info)» (<http://www.intuit.ru/studies/courses/2249/52/info>) Лекцию 6. После изучения лекции необходимо пройти Тест 6.
2. Изучите информацию о том, как использовать системные вызовы `msgget()`, `msgsnd()`, `msgrcv()` и `msgctl()`.

Наберите программы ([Листинг 9.1a](#) и [Листинг 9.1b](#)), сохраните под именами `09-1a.c` и `09-1b.c` соответственно, откомпилируйте и проверьте правильность их поведения.?

В отчёте приведите информацию об изученных системных вызовах, приведите выводы программ.

3. Измените программы из предыдущего пункта (сохраните их под именами `09-1a_mod.c` и `09-1b_mod.c`) для передачи нетекстовых сообщений. Воспользуйтесь предложенными ниже структурами:

```
struct date {
    int day;
    char month[20];
    int year;
};
struct persone {
    char firstname[20];
    char lastname[20];
    struct date bd;
};
```

В отчете приведите коды программ.

4. Измените программы из пункта 2 (сохраните их под именами 09-1a\_mod2.c и 09-1b\_mod2.c) для осуществления двусторонней связи через одну очередь сообщений. В отчете приведите коды программы.

### 12.3. Порядок подготовки и сдачи письменного отчета

Отчет должен описывать содержание работ студента при выполнении заданий 2–4.

#### Структура архива отчета:

- скриншот, подтверждающий прохождения теста (должны быть видны имя и отчество студента);
- отчет о проделанной работе в формате pdf;
- директория lesson12task2, в которой размещается код задания п. 2;
- директория lesson12task3, в которой размещается код задания п. 3;
- директория lesson12task4, в которой размещается код задания п. 4;

## Указания к лабораторным работам 13

# Организация файловой системы в Linux. Часть 1. Работа с файлами и директориями. Символьные ссылки

### 13.1. Цель работы

Лабораторная работа знакомит студентов с логической структурой файловой системы в ОС Linux. Закрепляет практические навыки работы с файловой системой с точки зрения пользователя.

### 13.2. Порядок выполнения работы и задания

1. Изучить в курсе «[Основы операционных систем. Практикум](http://www.intuit.ru/studies/courses/2249/52/info)» (<http://www.intuit.ru/studies/courses/2249/52/info>) Лекцию 8.
2. Изучите информацию о работе с файловой системой с точки зрения пользователя [https://www.intuit.ru/studies/courses/23/23/print\\_lecture/27141](https://www.intuit.ru/studies/courses/23/23/print_lecture/27141).
3. Рассмотрите следующую структуру директорий и файлов в домашней директории (/home/username/):

```
/home/username/  
├── dir1  
│   ├── dir2  
│   │   ├── dir3  
│   │   ├── file3  
│   │   ├── file4  
│   │   └── ~dir1 -> /home/username/dir1  
│   ├── dir4  
│   │   ├── ~file1 -> /home/username/dir1/file1  
│   │   └── ~file2 -> /home/username/dir1/file2  
│   ├── file1  
│   ├── file2  
│   ├── ~dir5 -> /home/username/dir1/dir4  
│   └── ~file3 -> /home/username/dir1/dir2/file3
```

Ответьте на вопросы. Является ли путь /home/username/dir1/dir2/dir1/dir2/ корректным? Что выдаст команда `ls` после перехода по указанному пути?

4. Создайте структуру директорий и файлов из предыдущего задания в своей домашней директории `username` = ваше имя в системе. Приведите в отчете последовательность всех команд, которые были использованы при создании этой структуры.

5. Добавьте в созданную структуру директорий и файлов 2 новых «объекта» (директорию и ссылку), чтобы получился «бесконечный цикл директорий»: `/home/username/dir1/loop/.../loop/loop`. Приведите последовательность команд, с помощью которых это было сделано.

### 13.3. Порядок подготовки и сдачи письменного отчета

Отчет должен описывать содержание работ студента при выполнении заданий 2–5 (обоснованные ответы на вопросы, последовательности команд).

#### Структура архива отчета:

- отчет о проделанной работе в формате `pdf`;
- `tar.gz`-архив директории `dir1`, созданной при выполнении заданий.

## Указания к лабораторным работам 14

# Организация файловой системы в Linux. Часть 2. Анализ содержимого директории

### 14.1. Цель работы

Лабораторная работа знакомит студентов с функциями языка C, которые позволяют анализировать содержимое директории: `opendir()`, `readdir()`, `rewinddir()`, `closedir()`. Закрепляет полученные знания написанием программы, которая распечатывает список файлов, входящих в директорию, с указанием их типов.

### 14.2. Порядок выполнения работы и задания

1. Изучить в курсе «**Основы операционных систем. Практикум**» (<http://www.intuit.ru/studies/courses/2249/52/info>) материал лекции 8, относящийся к функциям `opendir()`, `readdir()`, `rewinddir()`, `closedir()`.
2. Напишите одну из программ реализующих вывод списка файлов, входящих в директорию, с указанием их типов на основании поля `st_mode` структуры `stat`:

на оценку «уд»: Напишите, откомпилируйте и прогоните программу, распечатывающую список файлов домашней директории, с указанием их типов.

на оценку «хор»: Напишите, откомпилируйте и прогоните программу, распечатывающую список файлов, входящих в директорию, с указанием их типов. Имя директории задается как параметр командной строки в виде полного пути. Если оно отсутствует, то выбирается текущая директория. На вход программы может поступать любой корректный полный путь к любой директории.

на оценку «отл»: Напишите программу, распечатывающую содержимое заданной директории в формате, аналогичном формату выдачи команды `ls -al`. Имя директории задается как параметр командной строки в виде полного или относительного пути. Если оно отсутствует, то выбирается текущая директория. На вход программы может поступать любой корректный полный путь к любой директории.

Для реализации программы изучите функцию `ctime(3)` и системные вызовы `time(2)`, `readlink(2)`.

### **14.3. Порядок подготовки и сдачи письменного отчета**

Отчет должен описывать содержание работ студента при выполнении заданий 2.

#### **Структура архива отчета:**

- отчет о проделанной работе в формате pdf;
- tar.gz-архив с исходным кодом программы, созданной при выполнении задания 2.

## Указания к лабораторным работам 15

# Организация файловой системы в Linux. Часть 3. Memory mapped файлы

### 15.1. Цель работы

Лабораторная работа знакомит студентов с системными вызовами `mmap()`, `munmap()`. Закрепляет полученные знания написанием программы, которая иллюстрирует работу с memory mapped файлами.

### 15.2. Порядок выполнения работы и задания

1. Изучить в курсе «[Основы операционных систем. Практикум](http://www.intuit.ru/studies/courses/2249/52/info)» (<http://www.intuit.ru/studies/courses/2249/52/info>) материал лекции 8, относящийся к системным вызовам `mmap()`, `munmap()`.
2. Напишите одну из программ для работы с memory mapped файлами:

на оценку «уд»: Откомпилируйте и прогоните программу из [Листинга 11.1](#) (заполнить образ файла числами от 1 до 1000 и их квадратами). Исправьте ошибки в коде листинга программы.

на оценку «хор»: Напишите программу, которая будет заполнять образ файла числами от 1 до 1000, их квадратами и суммами по следующему принципу:

```
1   1   1   1
2   4   3   5
3   9   6  14
4  16  10  30
. . . . .
```

на оценку «отл»: Напишите две программы, использующие memory mapped файл для обмена информацией. Первая заполняет образ файла числами от 1 до 1000, их квадратами и суммами по принципу указанному выше. Вторая считывает данные из memory mapped файла и записывает их в обычный текстовый файл.

### 15.3. Порядок подготовки и сдачи письменного отчета

Отчет должен описывать содержание работ студента при выполнении заданий 2.

**Структура архива отчета:**

- отчет о проделанной работе в формате pdf;
- tar.gz-архив с исходным кодом программы, созданной при выполнении задания 2.

Не забывайте про курсовой проект, задания которого доступны по ссылке [https://drive.google.com/open?id=1U\\_pQkayY22b3EeaIriPwfbPTCKb7HU2S](https://drive.google.com/open?id=1U_pQkayY22b3EeaIriPwfbPTCKb7HU2S)

## Указания к лабораторным работам 16

# Аппарат прерываний. Работа с сигналами

### 16.1. Цель работы

Лабораторная работа знакомит студентов с понятием сигнала, способами возникновения сигналов и видами их обработки. Рассматриваются системные вызовы `getpgrp()`, `setpgrp()`, `getpgid()`, `setpgid()`, `getsid()`, `setsid()`. Практическая часть работы направлена на использование системных вызовов `kill()`, `signal()` и команды `kill`.

### 16.2. Порядок выполнения работы и задания

1. Изучить в курсе «[Основы операционных систем. Практикум](http://www.intuit.ru/studies/courses/2249/52/info)» (<http://www.intuit.ru/studies/courses/2249/52/info>) материал лекции 9, относящийся к аппарату прерываний и работе с сигналами.
2. Изучите, наберите, откомпилируйте и прогоните программы из листингов [13-14.1](#), [13-14.2](#), [13-14.3](#), [13-14.4](#), [13-14.5](#), [13-14.6](#), которые иллюстрируют работу с сигналами. Опишите поведение программ — их реакцию на сигналы (команду `kill`, нажатие клавиш `<CTRL>` и `<4>`), а также заложенные в них действия.
3. Пройдите [Тест 8](#).

### 16.3. Порядок подготовки и сдачи письменного отчета

Отчет должен описывать содержание работ студента при выполнении заданий 2: поведение программ — их реакцию на сигналы (команду `kill`, нажатие клавиш `<CTRL>` и `<4>`), а также заложенные в них действия.

### **Структура архива отчета:**

- отчет о проделанной работе в формате **pdf**;
- **tar.gz**-архива с исходными кодами программ, созданных при выполнении задания 2 (один архив, файлы исходников называть по шаблону **13-14.\*.c**).
- скриншот, подтверждающий прохождения теста (должны быть видны имя и отчество студента);

## **ПРИМЕРНЫЙ ПЕРЕЧЕНЬ ВОПРОСОВ**

### **Примерный перечень вопросов к зачету (5 семестр)**

1. Структура вычислительной системы. Понятие ОС. Основные понятия, концепции ОС.
2. Архитектурные особенности ОС. Функции ОС. Подходы к построению ОС.
3. Понятие процесса. Состояния процесса.
4. Операции над процессами и связанные с ними понятия (набор операций; Process Control Block и контекст процесса; одноразовые операции; многократные операции; переключение контекста).
5. Уровни планирования. Критерии планирования и требования к алгоритмам.
6. Параметры планирования. Вытесняющее и невытесняющее планирование.
7. Алгоритмы планирования First-Come и First-Served.
8. Алгоритмы планирования Round Robin и Shortest-Job-First.
9. Алгоритмы гарантированного и приоритетного планирования.
10. Алгоритмы планирования Multilevel Queue и Multilevel Feedback Queue.
11. Категории средств обмена информацией. Логическая организация механизма передачи информации.
12. Взаимодействующие процессы. Interleaving, race condition и взаимное исключение. Критическая секция.
13. Программные алгоритмы организации взаимодействия процессов и требования, предъявляемые к алгоритмам.
14. Алгоритмы синхронизации «Запрет прерываний» и «Переменная-замок».
15. Алгоритмы синхронизации «Строгое чередование» и «Флаги готовности».
16. Алгоритмы синхронизации «Алгоритм Петерсона» и Bakery Algorithm.
17. Аппаратная поддержка взаимного исключения. Команда Test-and-Set.
18. Механизмы синхронизации. Семафоры (концепция семафоров; решение проблемы producer-consumer с помощью семафоров).
19. Механизмы синхронизации. Мониторы и сообщения.
20. Эквивалентность семафоров, мониторов и сообщений. Реализация мониторов и передачи сообщений с помощью семафоров.
21. Эквивалентность семафоров, мониторов и сообщений. Реализация семафоров и передачи сообщений с помощью мониторов.
22. Эквивалентность семафоров, мониторов и сообщений. Реализация семафоров и мониторов с помощью очередей сообщений.
23. Тупики. Условия возникновения тупиков. Основные направления борьбы с тупиками.

### **Примерный перечень вопросов к экзамену (6 семестр)**

1. Физическая организация памяти компьютера. Локальность. Логическая память. Связывание адресов.
2. Функции системы управления памятью.
3. Простейшие схемы управления памятью. (Схема с фиксированными разделами. Один процесс в памяти. Оверлейная структура. Динамическое распределение. Свопинг. Схема с переменными разделами).
4. Сегментная и сегментно-страничная организация памяти. Понятие виртуальной памяти.
5. Архитектурные средства поддержки виртуальной памяти. (Страничная виртуальная память. Сегментно-страничная организация виртуальной памяти. Структура таблицы страниц. Ассоциативная память. Инвертированная таблица страниц).

6. Исключительные ситуации при работе с памятью. Стратегии управления страничной памятью.
7. Алгоритмы замещения страниц. Алгоритм FIFO. Выталкивание первой пришедшей страницы. Аномалия Билэди (Belady).
8. Алгоритмы замещения страниц. Оптимальный алгоритм (OPT).
9. Алгоритмы замещения страниц. Выталкивание дольше всего не использовавшейся страницы. Алгоритм LRU.
10. Алгоритмы замещения страниц. Выталкивание редко используемой страницы. Алгоритм NFU.
11. Алгоритмы замещения страниц. Другие алгоритмы. Управление количеством страниц, выделенных процессу.
12. Модель рабочего множества. Трешинг (Thrashing).
13. Страничные демоны.
14. Общие сведения о файлах. Организация файлов и доступ к ним.
15. Операции над файлами. Директории. Логическая структура файлового архива.
16. Разделы диска. Организация доступа к архиву файлов. Операции над директориями.
17. Защита файлов. Контроль доступа к файлам. Списки прав доступа.
18. Общая структура файловой системы.
19. Управление внешней памятью. (Методы выделения дискового пространства. Управление свободным и занятым дисковым пространством. Размер блока. Структура файловой системы на диске).
20. Реализация директорий. Примеры реализации директорий в некоторых ОС.
21. Поиск в директории. Линейный поиск. Хеш-таблица.
22. Монтирование файловых систем. Связывание файлов.
23. Кооперация процессов при работе с файлами. Примеры разрешения коллизий и тупиковых ситуаций.
24. Надежность файловой системы. Целостность файловой системы. Порядок выполнения операций. Журнализация.
25. Надежность файловой системы. Управление «плохими» блоками.
26. Производительность файловой системы. Кэширование. Оптимальное размещение информации на диске.
27. Реализация некоторых операций над файлами.
28. Современные архитектуры файловых систем. Физические принципы организации ввода-вывода.
29. Общие сведения об архитектуре компьютера. Структура контроллера устройства.
30. Опрос устройств и прерывания. Исключительные ситуации и системные вызовы.
31. Прямой доступ к памяти (Direct Memory Access – DMA).
32. Логические принципы организации ввода-вывода. Структура системы ввода-вывода.
33. Систематизация внешних устройств и интерфейс между базовой подсистемой ввода-вывода и драйверами.
34. Функции базовой подсистемы ввода-вывода. (Блокирующиеся, неблокирующиеся и асинхронные системные вызовы. Буферизация и кэширование. Spooling и захват устройств. Обработка прерываний и ошибок. Планирование запросов).
35. Алгоритмы планирования запросов к жесткому диску. Строение жесткого диска и параметры планирования.

36. Алгоритм First Come First Served (FCFS). Алгоритм Short Seek Time First (SSTF).
37. Алгоритмы сканирования (SCAN, C-SCAN, LOOK, C-LOOK).
38. Понятие протокола. Многоуровневая модель построения сетевых вычислительных систем.
39. Проблемы адресации в сети. Одноуровневые адреса. Двухуровневые адреса.
40. Проблемы адресации в сети. Удаленная адресация и разрешение адресов. Локальная адресация. Понятие порта. Полные адреса. Понятие сокета (socket).
41. Проблемы маршрутизации в сетях. Связь с установлением логического соединения и передача данных с помощью сообщений.
42. Синхронизация удаленных процессов.
43. Угрозы безопасности. Формализация подхода к обеспечению информационной безопасности.
44. Криптография как одна из базовых технологий безопасности ОС.
45. Шифрование с использованием алгоритма RSA.
46. Идентификация и аутентификация. Пароли, уязвимость паролей. Шифрование пароля.
47. Авторизация. Разграничение доступа к объектам ОС. Домены безопасности. Матрица доступа.
48. Недопустимость повторного использования объектов. Выявление вторжений. Аудит системы защиты.

### *Пример экзаменационного билета*

---



ФГБОУ ВПО «Самарский государственный  
технический университет»  
Кафедра «Прикладная математика  
и информатика»

Экзаменационный билет № 1  
по дисциплине «Операционные  
системы», 5-й семестр

---

1. Надежность файловой системы. Управление «плохими» блоками.
2. Идентификация и аутентификация. Пароли, уязвимость паролей. Шифрование пароля.
3. Простейшие схемы управления памятью. Схема с переменными разделами.

Составил: доц. каф. ПМИИ  
Утвердил: зав. каф. ПМИИ

М. Н. Саушкин  
В. П. Радченко

### **ЗАДАНИЕ** *по курсовому проектированию*

Студенту

1. Тема проекта Разработка программного обеспечения прикладного уровня с использованием системных интерфейсов набора стандартов POSIX
2. Сроки сдачи студентом законченного проекта 22.05.20 г.
3. Исходные данные к проекту

---

Приложение должно быть разработано для работы в ОС Linux. Программа должна быть написана на языке C (стандарты ANSI C, C99, C11; компилятор gcc) с использованием системных вызовов переносимого интерфейса операционных систем Unix (POSIX, стандарт

---

ISO/IEC 9945). Необходимо использовать системные вызовы для работы с процессами и нитями; работы с потоками ввода-вывода, файлами и файловой системой; работы с механизмами коммуникации и синхронизации процессов и нитей. Использование библиотечных вызовов языка C вместо соответствующих им системных вызовов запрещено, если иное не оговорено в варианте задания.

Функциональные требования к программе (согласно варианту задания №       ):

- 1) сразу после запуска должны порождаться несколько процессов, формируя определённое вариантом генеалогическое дерево процессов;
- 2) каждый из процессов должен вывести свой идентификатор и идентификатор родительского процесса; выполнить некоторые действия согласно варианту; вывести сообщение, что процесс с таким-то идентификатором и таким-то идентификатором родительского процесса завершает работу;
- 3) отправку и получение любых данных каждым процессом необходимо сопровождать выводом сообщения, что процесс с таким-то идентификатором произвёл такие-то действия;
- 4) приложение должно быть протестировано в различных вариантах запуска в терминале ОС Linux, в том числе в конвейерах с другими командами, с разными вариантами перенаправления потоков ввода-вывода.

---

#### 4. Содержание расчетно-пояснительной записки (перечень разрабатываемых вопросов)

---

Введение

1. Анализ постановки задачи
2. Проектирование структуры приложения
3. Разработка алгоритмов функционирования и структур данных
4. Реализация приложения
5. Результаты испытаний

Заключение

Список использованных источников

Приложения

---

---

#### 5. Перечень графического материала (с точным указанием обязательных чертежей и графиков)

---

---

7. Дата выдачи задания 22.01.20 г.

---

8. Календарный график работы над проектом на весь период проектирования (с указанием сроков выполнения и трудоемкости отдельных этапов)

---

1. Анализ постановки задачи	22.09 – 22.10:	20%
2. Проектирование структуры приложения	23.10 – 14.11:	20%
3. Разработка алгоритмов функционирования и структур данных	15.11 – 29.12:	10%
4. Реализация дополнительного функционала приложения	01.02 – 21.02:	30%
5. Результаты испытаний	22.02 – 13.03:	10%
6. Оформление пояснительной записки и сдача на проверку	14.04 – 22.04:	10%
7. Защита работы	23.04 – 29.05	

Руководитель

\_\_\_\_\_  
(подпись)

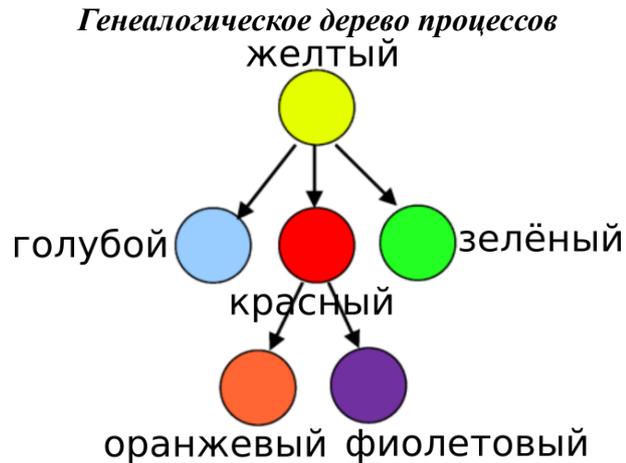
Задание принял к исполнению (дата)

---

(подпись студента)

---

## Пример варианта индивидуального задания по курсовому проектированию



### *Дополнительные действия процессов*

Все процессы должны выводить сообщения (см. функциональные требования), и, кроме того, производить дополнительные действия. Передачу и получение информации каждым из процессов необходимо сопровождать выводом на экран информации типа «процесс такой-то передал/получил такую-то информацию таким-то образом (через конвейер, FIFO и т.д.)». Для верной работы программ их необходимо запускать в конвейерах с какими-либо командами ОС (например, для получения на стандартном потоке ввода вашей программы расширенного списка файлов запустите `ls -l | ./myprog`).

*Для всех вариантов.* Каждый процесс, запустившись и сообщив об этом выводом строки в поток ошибки, должен отправить самому первому процессу сигнал SIGUSR1, а перед завершением работы сигнал SIGUSR2. При помощи этих сигналов первый процесс должен вести учет количества ваших процессов, функционирующих в данный момент в системе. По получении этих сигналов он должен вывести в поток ошибки сообщение с указанием этого количества.

Фиолетовый процесс выгружается последним. Прежде, чем завершиться, он должен создать новую директорию, в которую перенести все созданные в процессе работы файлы, а на их старом месте создать символические ссылки на эти файлы. Таким образом, после каждого из запусков должна оставаться новая директория с набором созданных файлов.

Механизмы синхронизации и коммуникации между фиолетовым и остальными процессами выбрать по своему усмотрению.

**Вариант 1.** Оранжевый получает со стандартного потока ввода список файлов каталога, и выводит их на экран, добавляя перед каждым именем порядковый номер; после этого общее количество полученных файлов оранжевый передает желтому (через fifo). Желтый полученное количество возводит в квадрат и передает зеленому через pipe. Зеленый выводит это число на экран и передает красному, используя разделяемую память. Тот полученную информацию отображает на экране и сохраняет в файле.

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

При проведении экзамена могут быть учтены результаты освоения дисциплины за семестр.

**Минимальное и максимальное (общее) количество баллов за 5 семестр**

Вид работы (контрольные точки)		Минимальное количество баллов	Максимальное количество баллов	Вес, %
1.	Устный опрос по 1 разделу, лабораторные работы 1, 2	9	15	30
2.	Лабораторные работы 3, 4	6	10	20
3.	Устный опрос по 2 разделу, лабораторные работы 5, 6	9	15	30
4.	Лабораторные работы 7, 8	6	10	20
<b>ИТОГО</b>		<b>30</b>	<b>50</b>	<b>100</b>

**Минимальное и максимальное (общее) количество баллов за 6 семестр**

Вид работы (контрольные точки)		Минимальное количество баллов	Максимальное количество баллов	Вес, %
1.	Устный опрос по 3 разделу, лабораторные работы 9, 10	6	10	25
2.	Устный опрос по 4 разделу, лабораторные работы 11, 12	6	10	25
3.	Устный опрос по 5 разделу, лабораторные работы 13, 14	6	10	25
4.	Устный опрос по 6 разделу, лабораторные работы 15, 16	6	10	25
<b>ИТОГО</b>		<b>24</b>	<b>40</b>	<b>100</b>

**Критерии оценивания устного опроса**

Систематически проводятся устные опросы. Правильный ответ на вопрос преподавателя оценивается в один балл. Максимальное количество баллов за одну КТ за устный опрос - 5 баллов.

**Критерии оценивания лабораторных работ**

Каждая лабораторная работа состоит из нескольких пунктов, которые предстоит выполнить студенту. Баллы за КТ выставляются в зависимости от качества проделанной студентом работы: получение верного результата, понимание проделанной работы, оформление, устный ответ. Максимальное количество баллов за одну КТ по лабораторным работам - 10 баллов (максимальное количество баллов за одну лабораторную работу - 5 баллов).

**Критерии и шкала оценивания результатов  
изучения дисциплины на промежуточной аттестации**

**Шкала оценивания**

«Отлично» – выставляется, если сформированность заявленных индикаторов компетенций 95% более (в соответствии с картами компетенций ОП): студент показал прочные знания основных положений фактического материала, умение самостоятельно решать конкретные практические задачи повышенной сложности, свободно использовать справочную литературу, делать обоснованные выводы из результатов анализа конкретных ситуаций;

«Хорошо» – выставляется, если сформированность заявленных индикаторов компетенций на 75% и более (в соответствии с картами компетенций ОП): обучающийся показал прочные знания основных положений фактического материала, умение самостоятельно решать конкретные практические задачи, предусмотренные рабочей программой, ориентироваться в рекомендованной справочной литературе, умеет правильно оценить полученные результаты анализа конкретных ситуаций;

«Удовлетворительно» – выставляется, если сформированность заявленных индикаторов компетенций 60% и более (в соответствии с картами компетенций ОП): обучающийся показал знание основных положений фактического материала, умение получить с помощью преподавателя правильное решение

конкретной практической задачи из числа предусмотренных рабочей программой, знакомство с рекомендованной справочной литературой;

«Неудовлетворительно» – выставляется, если сформированность заявленных индикаторов компетенций менее чем 59% (в соответствии с картами компетенций ОП): при ответе обучающегося выявились существенные пробелы в знаниях основных положений фактического материала, неумение с помощью преподавателя получить правильное решение конкретной практической задачи из числа предусмотренных рабочей программой учебной дисциплины.