

Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
Московский государственный университет имени М.В. Ломоносова  
Факультет вычислительной математики и кибернетики

**УТВЕРЖДАЮ**

Декан факультета ВМК МГУ

/И.А.Соколов/

2023 г.



## **РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)**

**Наименование дисциплины (модуля):**

**Практикум на ЭВМ (в поддержку курса «Операционные системы»)**

**Уровень высшего образования:**

**бакалавриат**

**Направление подготовки (специальность):**

**02.03.02 Фундаментальная информатика и информационные технологии**

**Направленность (профиль) ОПОП:**

**дисциплина относится к вариативной части профессионального цикла программы**

**Форма обучения:**

**очная**

Москва 2023

Рабочая программа дисциплины (модуля) разработана в соответствии с самостоятельно установленным МГУ образовательным стандартом (ОС МГУ) для реализуемых основных профессиональных образовательных программ бакалавриата, магистратуры, реализуемых последовательно по схеме интегрированной подготовки по направлениям 02.03.02, 02.04.02 «Фундаментальная информатика и информационные технологии» в редакции приказа МГУ от 30 декабря 2016г.

## **Место дисциплины (модуля) в структуре ОПОП ВО**

Дисциплина относится к вариативной части профессионального цикла. Содержание курса определяется образовательным стандартом МГУ высшего профессионального образования по направлению 02.03.02 «Фундаментальная информатика и информационные технологии»

## **Входные требования для освоения дисциплины (модуля), предварительные условия**

Курс «Практикум на ЭВМ» проводится в поддержку лекционного курса «Операционные системы», читаемого в 3 семестре. Изучаются язык программирования Си, особенности программирования на языке Си в среде операционной системы UNIX. Изучаются базовые средства организации и взаимодействия параллельных процессов. Изучение данного курса опирается на знания, полученные студентами в результате прослушивания курсов «Алгоритмы и алгоритмические языки», «Архитектура ЭВМ и язык ассемблера» и «Основы программирования». Данный курс является предшествующим для дальнейших модулей курса «Практикум на ЭВМ», курса «Системы программирования». Практические навыки, полученные при изучении дисциплины, могут быть использованы при выполнении выпускной квалификационной работы.

Освоение данной дисциплины (модуля) необходимо студентам (независимо от их дальнейшей специализации) для получения навыков разработки алгоритмов и практического программирования на языке Си, являющемся универсальным языком программирования, и, в частности, основным языком системного программирования. Навыки работы в среде операционной системы UNIX также являются весьма актуальными на сегодняшний день.

## **Результаты обучения по дисциплине (модулю), соотнесенные с требуемыми компетенциями выпускников.**

Процесс изучения дисциплины направлен на формирование элементов следующих компетенций в соответствии с ФГОС ВПО по направлению подготовки 02.03.02 «Фундаментальная информатика и информационные технологии»:

способность использовать базовые знания естественных наук, математики и информатики, основные факты, концепции, принципы теорий, связанных с фундаментальной информатикой и информационными технологиями (ОПК-1);

способность применять в профессиональной деятельности современные языки программирования, библиотеки и пакеты программ, современные профессиональные стандарты информационных технологий (ОПК-2);

способность к разработке алгоритмических и программных решений в области системного и прикладного программирования, созданию тестов и средств тестирования систем и средств на соответствие стандартам и исходным требованиям (ОПК-3);

способностью использовать современные инструментальные и вычислительные средства (ПК-3);

способность эффективно применять базовые математические знания и информационные технологии при решении проектно-технических и прикладных задач, связанных с развитием и использованием информационных технологий (ПК-6);

способность разрабатывать и реализовывать процессы жизненного цикла программного обеспечения (ПК-7);

способность составлять и контролировать план выполняемой работы, планировать необходимые для выполнения работы ресурсы, оценивать результаты собственной работы (ПК-11).

В результате освоения дисциплины студент должен:

### **знать**

- основные понятия, алгоритмы и методы организации управления процессами в операционной системе UNIX;
- основные понятия, алгоритмы и методы организации взаимодействия процессов в операционной системе UNIX;
- основные понятия, алгоритмы и методы организации работы с файлами;

### **уметь**

- разрабатывать алгоритмы для решения типовых задач, оценивать сложность полученных алгоритмов,
- реализовывать программы на языке Си с использованием системных вызовов ОС UNIX,
- тестировать написанные самостоятельно программы на соответствие исходным требованиям;
- находить, анализировать и контекстно обрабатывать научно-техническую информацию;

### **владеть**

- основами алгоритмизации, пониманием методов построения алгоритмов на основе разбиения задачи на подзадачи;
- навыками программирования на языке Си с использованием функций стандартной библиотеки языка Си, а также с использованием библиотеки системных вызовов ОС UNIX;
- навыками написания программ для работы с текстовыми и бинарными файлами;
- базовыми навыками разработки и реализации параллельных программ, организации взаимодействия процессов с использованием средств, предоставляемых ОС UNIX;
- навыками работы с пользовательским интерфейсом ОС UNIX.

### **Формат обучения**

Семинары по данной дисциплине проводятся как в формате аудиторных занятий, так и в виде практических занятий в компьютерном классе.

Предусмотрено выполнение ряда практических работ в течение семестра. Задания небольшого объема выдаются и выполняются непосредственно в компьютерном классе. Задания, большего объема, требующие предварительного проектирования, выдаются заранее для домашней подготовки. Домашние задания также предлагается выполнять на компьютере.

Используются традиционные технологии проведения практических занятий. Широко применяются *интерактивные и активные методы обучения*, например, коллективное обсуждение путей решения предлагаемых задач, разбор ошибочных ситуаций, самостоятельное выполнение студентами ряда практических заданий.

Взаимодействие преподавателя и студентов осуществляется, как в аудитории, так и дистанционно с помощью современных средств связи (электронная почта, социальные сети и т.п.) для обсуждения проблем, возникающих у студентов при самостоятельной работе. Преподаватели, ведущие практические занятия, периодически проводят консультации по дисциплине.

Все методические материалы для прохождения дисциплины отражены на сайте в Интернете.

### **Объем дисциплины (модуля)**

Объем дисциплины (модуля) составляет 2 (две) з.е., в том числе 72 академических часов, отведенных на контактную работу обучающихся с преподавателем.

## Содержание дисциплины (модуля)

### Тематический план курса

№ п/п	Название темы	Семинар	Самостоятельная работа студента на выполнение задания
1	<i>Язык программирования Си</i> : Повторение. Аргументы командной строки, работа со строками, динамические структуры данных.	2	2
2	<i>Язык программирования Си</i> : Указатели на функции. Работа с файлами (стандартная библиотека языка Си).	4	4
3	<i>Интерфейс с системой UNIX</i> : Низкоуровневая работа с файлами.	8	8
4	<i>Интерфейс с системой UNIX</i> : Организация работы с процессами.	2	2
5	<i>Интерфейс с системой UNIX</i> : Взаимодействие процессов. Каналы. Сигналы.	14	14
6	<i>Интерфейс с системой UNIX</i> : Межпроцессное взаимодействие средствами IPC. Очереди сообщений, разделяемая память, семафоры.	6	6
	Итого:	<b>36</b>	<b>36</b>
	Всего:	<b>72</b>	

### Структура дисциплины по видам работ

№ п/п	Раздел Дисциплины	Виды учебной работы, включая самостоятельную работу студентов и трудоемкость (в часах)			Формы текущего контроля успеваемости (по неделям семестра) Форма промежуточной аттестации (по семестрам)
		ауд	практ (МЗ)	сам	
1	<i>Язык программирования Си</i> : Повторение. Аргументы командной строки, работа со строками, динамические структуры данных.	2	0	2	Самостоятельная работа, индивидуальный опрос, прием практического домашнего задания
2	<i>Язык программирования Си</i> : Указатели на функции. Работа с файлами (стандартная библиотека языка Си).	2	2	4	Индивидуальный опрос, прием практического задания, прием практического домашнего задания
3	<i>Интерфейс с системой UNIX</i> : Низкоуровневая работа с файлами.	4	4	8	Индивидуальный опрос, прием практического задания, прием практического домашнего задания
4.	<i>Интерфейс с системой UNIX</i> : Организация работы с процессами.	2	0	2	Индивидуальный опрос, прием практического домашнего задания

5	<i>Интерфейс с системой UNIX:</i> Взаимодействие процессов. Каналы. Сигналы.	8	6	14	Индивидуальный опрос, прием практического задания, прием практического домашнего задания. Контрольная работа(15 неделя)
6	<i>Интерфейс с системой UNIX:</i> Межпроцессное взаимодействие средствами IPC. Очереди сообщений, разделяемая память, семафоры.	2	4	6	Индивидуальный опрос, прием практического домашнего задания. Практическая контрольная работа(18 неделя). Зачет

### **План семинарских занятий.**

Номера задач с префиксом «[2]» даются по задачнику [2] основной литературы (Н.В.Вдовикина, И.В.Машечкин, А.Н.Терехин, В.В.Тюляева. Программирование в ОС UNIX на языке Си.), с префиксом «[3]» даются по задачнику [3] основной литературы (Л.Н.Кузина. Сборник задач по практикуму: язык Си, ОС UNIX.).

#### **Семинар № 1. Язык программирования Си.**

Аргументы командной строки, работа со строками, динамическое выделение памяти, динамические структуры данных.

Задачи: [2] - 4.25; [3] - 1.1, 1.2, 1.3, 1.7

Домашнее задание: [3] – 1.8

#### **Семинар № 2. Язык программирования Си.**

Указатели на функции. Чтение сложных деклараций. Работа с файлами с использованием функций стандартной библиотеки языка Си: открытие и закрытие файлов, позиционирование в файле, чтение/запись данных. Функции `fopen()`, `fclose()`, `fseek()`, `ftell()`, `fgets()`, `fputs()`, `fread()`, `fwrite()` и т.д.

Задачи: [3] – 1.6, 1.4, 2.1, 2.2; [2] – 8.5

Домашнее задание: [3] – 2.4

#### **Семинар № 3. Практическое занятие в компьютерном классе.**

Выполнение задания на тему «Работа с файлами (стандартная библиотека языка Си)».

#### **Семинар № 4. Интерфейс с системой UNIX.**

Низкоуровневая работа с файлами. Работа с файлами с использованием системных вызовов ОС UNIX. Таблицы, используемые ОС UNIX для работы с файлами. Понятия индексного дескриптора и файлового дескриптора. Открытие и закрытие, создание и удаление файлов, позиционирование в файле, чтение/запись данных, права доступа к файлу, получение информации о файле. Функции `open()`, `close()`, `lseek()`, `read()`, `write()`, `stat()` и т.д.

Задачи: [3] – 2.5, 2.6, 2.7; [2] – 10.2

Домашнее задание: [3] – 2.16

#### **Семинар № 5. Практическое занятие в компьютерном классе.**

Выполнение задания на тему «Работа с файлами (системные функции)»

#### **Семинар № 6. Интерфейс с системой UNIX.**

Низкоуровневая работа с файлами. Решение задач.

Задачи: [3] – 2.14, 2.15

Домашнее задание: [3] – 2.17

#### **Семинар № 7. Практическое занятие в компьютерном классе.**

Сдача задания на тему «Реализация команд UNIX».

**Семинар № 8. Интерфейс с системой UNIX.**

Понятие процесса. Идентификация процессов. Создание и завершение процессов. Ожидание завершения процесса-потомка. Замена тела процесса. Системные вызовы: fork(), exit(), getpid(), getppid(), wait(), waitpid(), вызовы семейства exec().

Задачи: [2] – 9.1, [3] - 3.1

Домашнее задание: [3] – 3.2

**Семинар № 9. Интерфейс с системой UNIX.**

Взаимодействие процессов. Каналы. Перенаправление ввода-вывода. Организация конвейера. Системные вызовы: dup(), dup2(), pipe().

Задачи: [3] – 3.3, 3.5

Домашнее задание: [3] – 3.4, 3.6

**Семинар № 10.** Практическое занятие в компьютерном классе. Выполнение задания на тему «Взаимодействие процессов. Каналы».

**Семинар № 11. Интерфейс с системой UNIX.**

Взаимодействие процессов. Сигналы. Функции: signal(), kill(), alarm(), pause().

Задачи: [3] - 3.8, 3.9, 3.10

Домашнее задание: [3] – 3.15, 3.16

**Семинар № 12.** Практическое занятие в компьютерном классе.

Выполнение задания на тему «Взаимодействие процессов. Сигналы».

**Семинар № 13. Интерфейс с системой UNIX.**

Взаимодействие процессов. Каналы. Сигналы. Решение задач.

Задачи: [2] – 11.1, 11.5, [3] - 3.11, 3.13

Домашнее задание: [2] – 11.6

**Семинар № 14.** Практическое занятие в компьютерном классе.

Сдача задания на тему «Моделирование элементов работы командного интерпретатора Shell».

**Семинар № 15. Контрольная работа.**

**Семинар № 16. Интерфейс с системой UNIX.**

Межпроцессное взаимодействие средствами IPC. Именованные разделяемые объекты. Очереди сообщений, разделяемая память, семафоры.

Задачи: [3] - 4.1, 4.4

Домашнее задание: [3] – 4.3

**Семинар № 17.** Практическое занятие в компьютерном классе.

Сдача задания на тему «IPC».

**Семинар № 18.** Практическое занятие в компьютерном классе. Контрольная работа.

**Фонд оценочных средств (ФОС) для оценивания результатов обучения по дисциплине**

В конце семестра по дисциплине выставляется зачет с оценкой (дифференцированный зачет).

В формировании оценки учитываются промежуточные результаты работы студента в течение семестра: своевременность и качество выполнения практических заданий, домашних заданий, итоги самостоятельных и контрольных работ.

**Типовые контрольные задания или иные материалы для проведения текущего контроля успеваемости**

Приводятся примеры заданий, выполняемых студентами на практических занятиях.

### Пример задания № 1. Работа с файлами. Стандартная библиотека

Программа. Дан файл и строка. Все вхождения строки в файл (в том числе и в качестве подстроки) удалить. Имя файла и строка задаются в командной строке. Строка с пробелами при передаче ее в командной строке должна быть заключена в двойные кавычки.

### Пример задания № 2. Работа с файлами. Системные функции

Написать программу. В командной строке передается имя файла и число **n**.

- Вывести на экран **n** первых строк файла, если строк в файле меньше **n**, вывести их все.
- Если файла с заданным именем еще не существует, создать его и заполнить **n** строками, введенными со стандартного ввода.

Для работы с файлом использовать только системные функции.

### Пример задания № 3. Реализация команд UNIX.

Сортировка файлов `sort [-r +n -m -o] файлы`

Сортировать строки каждого файла в лексикографическом порядке.

`sort -r файлы` - сортировка в обратном порядке.

`sort +n файлы` – сортировать файл, начиная с **n**-ой строки.

`sort -m файлы` – слияние исходных (предварительно отсортированных ) файлов.

`sort -o выходной файл` - результат направляется не на стандартный вывод (как происходит по умолчанию), а в **выходной файл**, который может совпадать с одним из исходных.

Порядок опций при задании команды может быть любым, и любая из них (в том числе и все) при запуске программы может отсутствовать.

### Пример задания № 4. Взаимодействие процессов. Каналы

В командной строке передается имя файла.

Процесс-отец создает сына. Отец считывает со стандартного ввода строку (длина не больше 20 символов) и передает эту строку процессу-сыну. Сын выводит в заданный файл полученную строку. После этого отец выводит в файл строку из 10 '=' и снова переходит к вводу. Работа продолжается до ввода команды «exit». Синхронизировать работу с помощью аппарата каналов.

*Подсказка.* После каждого вывода рекомендуется сбрасывать буфер с помощью вызова функции `fflush()`. Завершение процессов также должно быть аккуратным – с закрытием файла.

### Пример задания № 5. Взаимодействие процессов. Сигналы

Написать программу. Работают два процесса: отец и сын.

Отец выводит на экран “Father”, а сын - “Son”. Вывод строго чередуется и должен происходить многократно (Father Son Father Son Father Son и т.д.) в течение 2 секунд.

Через 2 секунды сын завершает работу, а отец продолжает выводить “Father”, но с интервалом в 1 секунду. Завершается отец по Ctrl-C. Не должно остаться процессов –зомби.

### Пример задания № 6. Моделирование элементов работы командного интерпретатора Shell

#### Реализация конвейера.

Написать программу, которая осуществляет запуск конвейера из произвольного количества процессов. Команда считывается со стандартного ввода. Правильная команда описывается следующими правилами:

<Команда> → <Конвейер> {<перенаправление ввода/вывода>}

<перенаправление ввода/вывода> →

{<перенаправление ввода > } <перенаправление вывода> |

{<перенаправление вывода>} <перенаправление ввода >

<перенаправление ввода > → '<' файл

<перенаправление вывода> → '>' файл | '>>' файл



<Конвейер>→ <Простая команда> {'|' <Конвейер>}

<Простая команда>→ <имя команды><список аргументов>

Примеры команд: **ls -a -l |wc| wc**

**cat | sort -r <file1 >file2**

Пробелы между отдельными элементами команды (аргументы, имена файлов, '|', '>', '>>', '<') допустимы в произвольном количестве.

**pr1 | ...| prN** – конвейер: стандартный вывод всех команд, кроме последней, направляется на стандартный ввод следующей команды конвейера. Каждая команда выполняется как самостоятельный процесс (т.е. все  $pr_i$  выполняются параллельно). Управляющий процесс ожидает завершения последней команды. Не должно оставаться процессов «зомби».

### Пример задания № 7. IPC

Сервер – 2 клиента.

Сервер в командной строке получает имя файла. Длина строк в файле ограничена MAXLEN.

Сервер читает строки из файла и передает их клиентам через разделяемую память.

Клиент считывает из разделяемой памяти строку, дописывает в начало строки свои идентификационные данные (например, свой PID) и выводит строку в файл-результат (имя определено заранее).

После того, как оба клиента считали строку, сервер помещает в разделяемую память следующую строку файла и т.д. Работа продолжается до конца исходного файла.

Вывод в файл-результат осуществляется клиентами строго поочередно.

Синхронизацию доступа к файлу реализовать с помощью аппарата семафоров.

Разделяемая память и семафор(ы) должны быть корректно удалены по окончании работы.

*Типовые контрольные задания или иные материалы для проведения промежуточной аттестации.*

### Пример контрольной работы (выполняется в аудитории)

1, Дан однонаправленный список, элемент данных типа **int**.

Написать функцию (от 2 параметров) для удаления из списка всех элементов, значения которых равны заданному числу.

2. Написать функцию, параметр – имя файла. Функция должна в данном файле поменять местами первую и последнюю строки. Использовать функции стандартной библиотеки. Длина строк в файле неограниченна.

3. В командной строке передаются имена нескольких исполняемых файлов. Если их меньше двух, ничего не делать. Иначе запустить эти файлы сначала на параллельное исполнение. Когда они все выполнятся, запустить еще раз последние 2 из них, но уже последовательно. Не должно остаться процессов «зомби».

4. Написать программу. Работают 2 процесса. Отец читает со стандартного ввода по строке (длина строки не больше MAXLEN), передает строку сыну. Сын выводит на экран длину строки. Только **после вывода** длины отец **считывает** очередную строку. Работа заканчивается по второму нажатию ^C. **Синхронизировать работу с помощью сигналов**. Не должно остаться процессов «зомби».

### Пример практической контрольной работы (выполняется в компьютерном классе)

Написать программу. Работают 2 процесса. Отец читает со стандартного ввода имя исполняемого файла, передает его сыну. Сын запускает этот файл на исполнение. Только после того, как исполняемый файл завершит работу, сын сообщает об этом отцу, и отец считывает следующее имя. Работа заканчивается по второму нажатию ^C. Синхронизировать работу с помощью сигналов. Не должно остаться процессов «зомби».

**Шкала и критерии оценивания результатов обучения (РО) по дисциплине (модулю)**

Оценка				
РО и соответствующие виды оценочных средств	2	3	4	5
<b>Знания</b> (устные и письменные опросы, практические и контрольные задания, зачет)	Отсутствие знаний	Фрагментарные знания	Общие, но не структурированные знания	Сформированные систематические знания
<b>Умения</b> (практические и контрольные задания, зачет)	Отсутствие умений	В целом успешное, но не систематическое умение	В целом успешное, но содержащее отдельные пробелы умение (допускает неточности не принципиального характера)	Успешное и систематическое умение
<b>Навыки (владения, опыт деятельности)</b> (практические и контрольные задания, зачет)	Отсутствие навыков (владений, опыта)	Наличие отдельных навыков (наличие фрагментарного опыта)	В целом, сформированные навыки (владения), но используемые не в активной форме	Сформированные навыки (владения), применяемые при решении задач

**Соответствие результатов обучения и компетенций, в развитии которых участвует дисциплина (модуль)**

Результаты обучения	Компетенция, с формированием которой связано достижение результата обучения
<p><b><u>знать</u></b></p> <ul style="list-style-type: none"> <li>- основные понятия, алгоритмы и методы организации управления процессами в операционной системе UNIX;</li> <li>- основные понятия, алгоритмы и методы организации взаимодействия процессов в операционной системе UNIX;</li> <li>- основные понятия, алгоритмы и методы организации работы с файлами;</li> </ul>	<p>ОПК-1 ОПК-2</p>
<p><b><u>уметь</u></b></p> <ul style="list-style-type: none"> <li>- разрабатывать алгоритмы для решения типовых задач, оценивать сложность полученных алгоритмов,</li> <li>- реализовывать программы на языке Си с использованием системных вызовов ОС UNIX,</li> <li>- тестировать написанные самостоятельно программы на соответствие исходным требованиям;</li> <li>- находить, анализировать и контекстно обрабатывать научно-техническую информацию;</li> </ul>	<p>ОПК-3 ПК-3 ПК-6 ПК-7 ПК-11</p>
<p><b><u>владеть</u></b></p> <ul style="list-style-type: none"> <li>- основами алгоритмизации, пониманием методов</li> </ul>	

