

**Федеральное государственное автономное образовательное  
учреждение высшего образования  
«Московский физико-технический институт  
(национальный исследовательский университет)»**

**УТВЕРЖДЕНО**  
**Проректор по учебной работе**

**А.А. Воронов**

	<b>Рабочая программа дисциплины (модуля)</b>
<b>по дисциплине:</b>	Операционные системы UNIX
<b>по направлению:</b>	Прикладные математика и физика
<b>профиль подготовки:</b>	Физика атомных ядер и элементарных частиц, физика высоких энергий Физтех-школа природоподобных, плазменных и ядерных технологий им. И.В. Курчатова кафедра информатики и вычислительной математики
<b>курс:</b>	2
<b>квалификация:</b>	бакалавр

Семестр, формы промежуточной аттестации: 3 (осенний) - Дифференцированный зачет

Аудиторных часов: 45 всего, в том числе:

лекции: 0 час.

семинары: 0 час.

лабораторные занятия: 45 час.

Самостоятельная работа: 45 час.

Всего часов: 90, всего зач. ед.: 2

Количество контрольных работ, заданий: 2

Программу составил: В.И. Голубев, д-р физ.-мат. наук, доцент

Программа обсуждена на заседании кафедры информатики и вычислительной математики 30.01.2024

## Аннотация

Рассматривается широкий круг вопросов функционирования операционной системы UNIX, в том числе базовые вопросы построения операционной системы, принципы организации файловой системы UNIX, учетные записи пользователей, установка, запуск и функционирование популярных операционных систем, взаимодействие пользователя с операционной системой и командные оболочки. С позиции пользователя изложены базовые концепции работы операционной системы UNIX в сетях TCP/IP, настройка Интернета и электронной почты. Рассмотрены вопросы разработки программного обеспечения, в том числе программ на языке Си, приложений на Java и командных сценариев на языке Perl. Приведены исходные коды целого ряда программ и примеров. В доступной форме даны принципы построения графических интерфейсов пользователя и работа с ними.

### 1. Цели и задачи

#### Цель дисциплины

- получение базовых знаний об организации операционных систем, разделении обязанностей между аппаратным обеспечением и ядром операционной системы. Рассмотрение концепций современных операционных систем производится на примере операционной системы Unix. Рассматриваются пользовательский интерфейс Unix, программирование на языке Unix Shell, использование системных вызовов для взаимодействия с ядром в программах на языке Си.

#### Задачи дисциплины

- изучение основных концепций и принципов проектирования операционных систем. Рассмотрение взаимодействия ядра операционной системы с аппаратным обеспечением современных компьютеров;
- рассмотрение реализации основных концепций современных ОС на примере Unix (понятия процесс, планировщик процессов файл и др.);
- знакомство с командной оболочкой Unix Shell на уровне пользователя и программиста. Выполнение лабораторных работ по написанию Shell-скриптов. Выполнение лабораторных работ на других скриптовых языках, в том числе, sed и AWK;
- изучение основных системных вызовов Unix. Программирование на языке Си с использованием системных вызовов.

### 2. Перечень формируемых компетенций

Освоение дисциплины направлено на формирование следующих компетенций:

Код и наименование компетенции	Индикаторы достижения компетенции
ОПК-1 Способен применять фундаментальные знания, полученные в области физико-математических и (или) естественных наук, и использовать их в профессиональной деятельности	ОПК-1.1 Способен анализировать поставленную задачу, намечать пути ее решения
	ОПК-1.2 Способен строить математические модели, производить количественные расчеты и оценки
	ОПК-1.3 Способен определять границы применимости полученных результатов

### 3. Перечень планируемых результатов обучения по дисциплине (модулю)

В результате освоения дисциплины обучающиеся должны

знать:

- основные компоненты ОС общего назначения, необходимые для её функционирования;
- основные команды, необходимые для уверенной работы в Unix Shell на уровне пользователя;
- управляющие операторы и управляющие конструкции Unix Shell, необходимые для написания shell-скриптов.

уметь:

- работать в командной оболочке Unix Shell, писать скрипты для Unix Shell, писать программы на языке Си с использованием системных вызовов ОС Unix.

владеть:

- приёмами программирования на скриптовых языках на примере Unix Shell, awk и sed.

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

#### 4.1. Разделы дисциплины (модуля) и трудоемкости по видам учебных занятий

№	Тема (раздел) дисциплины	Трудоемкость по видам учебных занятий, включая самостоятельную работу, час.			
		Лекции	Семинары	Лаборат. работы	Самост. работа
1	Организация ОС Unix, командная оболочка ОС Unix			45	45
Итого часов				45	45
Подготовка к экзамену		0 час.			
Общая трудоёмкость		90 час., 2 зач.ед.			

#### 4.2. Содержание дисциплины (модуля), структурированное по темам (разделам)

Семестр: 3 (Осенний)

##### 1. Организация ОС Unix, командная оболочка ОС Unix

Основные особенности ОС UNIX. История UNIX. Версии UNIX. Многозадачность. Режим разделения времени. Виртуальная память. Многопользовательская ОС. UID и GID. Права доступа к файлам. Процессы. Выполняемые файлы. Атрибуты процессов. Файлы /etc/passwd, /etc/group и /etc/master.passwd. Login Shell. Домашняя директория.

Авторизация пользователей. Виды терминалов. Программа putty. Man. Управление терминалом с помощью stty. Файл termcap. Горячие клавиши. csh и sh. История. Bash. Переменные. Операторы присваивания. Встроенные и внешние команды. Спец комментарий #!. PATH. Команда echo. Подстановки. Порядок подстановок. Экранирование символов. Запуск внешних команд shell-ом. Приглашение. Интерактивный shell. PS1 и PS2. Переменные окружения. Shell scripts. Файл .profile. Задача "hello".

Права доступа к файлам и директориям. Команды chown, chmod, chgrp. SUID -ные программы. Sticky bit. Файловые шаблоны. Перенаправление ввода/вывода. Условия в shell. Код возврата. Управляющие конструкции. Команда test. Группировка команд с помощью фигурных и круглых скобок. Чтение файлов с помощью команды read. Переменная IFS. Задача ФИО.

Файловые системы. Древоподобная структура оглавлений. Назначение основных директорий. Рабочая директория. Относительные и полные имена файлов. Команды работы с файлами. Имя переменной в фигурных скобках. Задание значения переменной по умолчанию. Редактирование значения переменной. Доступ к аргументам командной строки. Команда shift. Специальные переменные. Отладка программ на shell. Задача whichx.

Структура файловой системы. Inodes. Жесткие и мягкие ссылки. Типы файловых объектов. Оглавления . и .. . Корневая файловая система. Монтирование файловых систем. Фоновые процессы. Управление фоновыми процессами. Задача catslow.

Символьные и блочные устройства. Оглавление /dev. Создание файлов устройств. Разделы и слайсы. Подготовка жестких дисков для использования в ОС UNIX. Кэширование дисков. Команда sync. Сигналы. Причины возникновения сигналов. Дамп памяти. Обработка сигналов по умолчанию. Управляющий терминал. Сеансы. Задача exesbg.

Программы работы с файлами: cp, mv, cat, split, head, tail, find, locate, xargs. Функции в shell. Команда source. Задача addpath

Пакетная обработка заданий. Cron. Формат файла crontab. Пользовательские crontab. Организация взаимного исключения доступа к файлам. Задача «отдел кадров»

Pipes. Фильтры. Примеры программ-фильтров. Временные файлы. Mktmp. Оглавление /tmp. Изменение номеров inodes при редактировании, копировании и перемещении файлов. Задача overwrite.

Команды работы с процессами: ps, top. Оглавление /proc. Приоритеты процессов. Команда nice. Регулярные выражения. Диалекты регулярных выражений. Программы обработки текстов: ed, sed, grep, cut. Задача frame.

Организация вычислений в shell. Команда exes. Команда env. Использование exes для перенаправления ввода/вывода. Команда expr. Хэш таблицы. Создание хэш таблиц в shell. Задача workers.

Форматы файлов. Команда File. Magic file. Библиотеки. Команда tar. Резервное копирование. Команды dump и restore. Инкрементный backup. Файл dumpdates. Алгоритм Ханойской башни. Команда dd. Учет работы пользователей в ОС Unix. Файлы utmp и wtmp. Команды who, w, finger, last, ac. Задача ucounter.

Awk. Селекторы и действия. Управляющие конструкции. Поля. Специальные переменные. Встроенные функции. Индексы массивов в awk. Селекторы BEGIN и END. Задача «частотный словарь».

Дисковые квоты. Grace period. ulimits. Виды пользовательских лимитов.

Perl. Скаляры, массивы и хэштаблицы. Основные управляющие конструкции. Суффиксы команд. Ввод/вывод. Встроенные функции. Специальные переменные. Ключи командной строки. Обработка файла на месте.

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

- персональные компьютеры;
- учебная аудитория;
- мультимедийный проектор;
- экран.

## **6. Перечень рекомендуемой литературы**

### Основная литература

1. Операционная система UNIX [Текст] : учеб. пособие для вузов / А. М. Робачевский, С. А. Немнюгин, О. Л. Стесик .— 2-е изд., перераб. и доп. — СПб. : БХВ-Петербург, 2005, 2007, 2010 .— 656 с.

### Фонд литературы кафедры

2. Б. Керниган, Р. Пайк. Unix. Программное окружение. ISBN 5-93286-029-4, 0-13-937681-X; 2003 г.

### Дополнительная литература

### Фонд литературы кафедры

1. А. Роббинс. Unix. Справочник. ISBN 5-91136-031-4; 2007 г.
2. Л. Торвальдс, Э. Олман, Д. М. Ритчи, Э. Немец, Г. Снайдер, С. Сибасс, Т. Р. Хейн. UNIX. Руководство системного администратора. Для профессионалов. ISBN 978-5-318-00754-5, 5-318-00754-6; 2007 г.

## **7. Перечень ресурсов информационно-телекоммуникационной сети "Интернет", необходимых для освоения дисциплины (модуля)**

1. <http://www.citforum.ru/>
2. <http://www.opennet.ru/man.shtml> (оригиналы и переводы справочных руководств Unix).
3. <https://www.freebsd.org/cgi/man.cgi> (оригинальные MAN-страницы ОС FreeBSD)

## **8. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень необходимого программного обеспечения и информационных справочных систем (при необходимости)**

Во время занятий демонстрируются презентации с помощью мультимедийных технологий. Разбор примеров производится на удалённом Unix-сервере с помощью подключения по протоколу SSH.

## **9. Методические указания для обучающихся по освоению дисциплины (модуля)**

Освоение курса требует напряжённой работы студента. Для этого студента необходимо:

- чтение и конспектирование рекомендованной литературы;
- проработка учебного материала (по учебной и научной литературе), подготовка ответов на вопросы, предназначенных для самостоятельного изучения;
- доказательство отдельных утверждений, свойств;
- решение предлагаемых задач.

## ПРИЛОЖЕНИЕ

### ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

<b>по направлению:</b>	Прикладные математика и физика
<b>профиль подготовки:</b>	Физика атомных ядер и элементарных частиц, физика высоких энергий Физтех-школа природоподобных, плазменных и ядерных технологий им. И.В. Курчатова кафедра информатики и вычислительной математики
<b>курс:</b>	<u>2</u>
<b>квалификация:</b>	бакалавр
Семестр, формы промежуточной аттестации: 3 (осенний) - Дифференцированный зачет	
<b>Разработчик:</b>	В.И. Голубев, д-р физ.-мат. наук, доцент

## 1. Компетенции, формируемые в процессе изучения дисциплины

Код и наименование компетенции	Индикаторы достижения компетенции
ОПК-1 Способен применять фундаментальные знания, полученные в области физико-математических и (или) естественных наук, и использовать их в профессиональной деятельности	ОПК-1.1 Способен анализировать поставленную задачу, намечать пути ее решения
	ОПК-1.2 Способен строить математические модели, производить количественные расчеты и оценки
	ОПК-1.3 Способен определять границы применимости полученных результатов

## 2. Показатели оценивания компетенций

В результате изучения дисциплины «Операционные системы UNIX» обучающийся должен:

### знать:

- основные компоненты ОС общего назначения, необходимые для её функционирования;
- основные команды, необходимые для уверенной работы в Unix Shell на уровне пользователя;
- управляющие операторы и управляющие конструкции Unix Shell, необходимые для написания shell-скриптов.

### уметь:

- работать в командной оболочке Unix Shell, писать скрипты для Unix Shell, писать программы на языке Си с использованием системных вызовов ОС Unix.

### владеть:

- приёмами программирования на скриптовых языках на примере Unix Shell, awk и sed.

## 3. Перечень типовых (примерных) вопросов, заданий, тем для подготовки к текущему контролю

В целях текущего контроля успеваемости предусмотрен краткий опрос по темам предыдущих занятий по теме прошлой лекции или в конце занятия по пройденной теме.

## 4. Перечень типовых (примерных) вопросов и тем для проведения промежуточной аттестации обучающихся

Промежуточный контроль знаний, умений, навыков по дисциплине «Операционные системы UNIX» проводится при защите лабораторных работ

Примерный перечень тем лабораторных работ

Hello, world

Shell скрипт печатающий:

- Строку "Hello world!!!"
- Строку "I'm ваш login"  
login берется из переменной USER
- Строку "Today: сегодняшняя дата/время"  
Дата берется из выдачи команды "date"

ФИО

Напечатать Фамилию Имя Отчество студентов своего потока используя команду read.

Фамилию Имя Отчество брать из файла /etc/passwd. Учесть что все студенты одного потока имеют одинаковый GID.

Exec

Программа должна запустить новый процесс, запустить в нем исполняемый код программы echo с помощью execsr и передать программе echo два параметра "Hello," и "world", дождаться пока программа echo закончит работу, и напечатать ее код возврата.

Dup

Перенаправить вывод команды echo предыдущей программы в файл с помощью dup2.

Pipe

Программа должна создать два процесса. В одном из них вызвать программу echo, как и в третьей задаче, а в другом команду cat и соединить эти две команды в конвейер с помощью системного вызова pipe, перенаправив вывод команды echo в качестве стандартного ввода команды cat. Программа должна дождаться завершения обоих процессов и напечатать их pids в порядке завершения процессов.

#### Myls

Программа должна печатать содержимое каталога, указанного параметром команды. Для каждого объекта каталога она должна печатать его номер инода, число жестких ссылок, тип, время последнего изменения файла, размер и имя файлового объекта.

#### Timeout

Программа должна запустить команду, указанную в командной строке начиная со второго параметра. Первый параметр командной строки должен указывать на максимальное время выполнения команды. Если это время будет превышено, команда должна быть принудительно завершена. Программа должна печатать произошло ли завершение команды по таймауту или команда завершилась нормально.

#### Unix sockets

Написать клиент и сервер обмена файлами через unix socket. Сервер должен в бесконечном цикле обрабатывать запросы от клиентов. Каждый запрос представляет собой имя файла. Обработка запроса должна заключаться в передаче содержимого файла обратно клиенту. Клиент должен послать серверу запрос с именем файла, указанного в командной строке, а затем выводить данные, полученные от сервера на стандартный вывод. По окончании вывода он должен завершать работу. Программа должна правильно обрабатывать двоичные данные. Сервер должен допускать повторный запуск.

### Критерии оценивания

Оценка «отлично (10)» выставляется студенту, показавшему всесторонние, систематизированные, глубокие знания учебной программы дисциплины, проявляющему интерес к данной предметной области, продемонстрировавшему умение уверенно и творчески применять их на практике при решении конкретных задач, свободное и правильное обоснование принятых решений.

Оценка «отлично (9)» Выставляется студенту, показавшему всесторонние, систематизированные, глубокие знания учебной программы дисциплины и умение уверенно применять их на практике при решении конкретных задач, свободное и правильное обоснование принятых решений.

Оценка «отлично (8)» Выставляется студенту, показавшему всесторонние, систематизированные, глубокие знания учебной программы дисциплины и умение уверенно применять их на практике при решении конкретных задач, правильное обоснование принятых решений, с некоторыми недочетами.

Оценка «хорошо (7)» Выставляется студенту, если он твердо знает материал, грамотно и по существу излагает его, умеет применять полученные знания на практике, но недостаточно грамотно обосновывает полученные результаты.

Оценка «хорошо (6)» Выставляется студенту, если он твердо знает материал, грамотно и по существу излагает его, умеет применять полученные знания на практике, но допускает в ответе или в решении задач некоторые неточности.

Оценка «хорошо (5)» Выставляется студенту, если он в основном знает материал, грамотно и по существу излагает его, умеет применять полученные знания на практике, но допускает в ответе или в решении задач достаточно большое количество неточностей.

Оценка «удовлетворительно (4)» выставляется студенту, показавшему фрагментарный, разрозненный характер знаний, недостаточно правильные формулировки базовых понятий, нарушения логической последовательности в изложении программного материала, но при этом он освоил основные разделы учебной программы, необходимые для дальнейшего обучения, и может применять полученные знания по образцу в стандартной ситуации.

Оценка «удовлетворительно (3)» выставляется студенту, показавшему фрагментарный, разрозненный характер знаний, допускающему ошибки в формулировках базовых понятий, нарушения логической последовательности в изложении программного материала, слабо владеет основными разделами учебной программы, необходимыми для дальнейшего обучения и с трудом применяет полученные знания даже в стандартной ситуации.



Оценка «неудовлетворительно (2)» выставляется студенту, который не знает большей части основного содержания учебной программы дисциплины, допускает грубые ошибки в формулировках основных принципов и не умеет использовать полученные знания при решении типовых задач.

Оценка «неудовлетворительно (1)» выставляется студенту, который не знает основного содержания учебной программы дисциплины, допускает грубейшие ошибки в формулировках базовых понятий дисциплины и вообще не имеет навыков решения типовых практических задач.

## **5. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности**

Дифференцированный зачет проводится по итогам сдачи задания и контрольных работ, предусмотренных программой дисциплины «Введение в информатику»