

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

УТВЕРЖДЕНО

**Директор института нано-, био-,
информационных, когнитивных
и социогуманитарных наук и
технологий**

П.А. Форш

Рабочая программа дисциплины (модуля)

по дисциплине:	Программные интерфейсы операционной системы UNIX
по направлению:	Прикладные математика и физика
профиль подготовки:	Термоядерная энергетика и плазменные технологии Физтех-школа природоподобных, плазменных и ядерных технологий им. И.В. Курчатова кафедра информатики и вычислительных сетей
курс:	2
квалификация:	бакалавр

Семестр, формы промежуточной аттестации: 4 (весенний) - Дифференцированный зачет

Аудиторных часов: 60 всего, в том числе:

лекции: 0 час.

семинары: 0 час.

лабораторные занятия: 60 час.

Самостоятельная работа: 75 час.

Всего часов: 135, всего зач. ед.: 3

Количество контрольных работ, заданий: 4

Программу составил: В.А. Федотов, ассистент

Программа обсуждена на заседании кафедры информатики и вычислительных сетей 19.03.2020

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	Системные вызовы ОС Unix, системное программирование на Си в ОС Unix.			60	75
Итого часов				60	75
Подготовка к экзамену		0 час.			
Общая трудоёмкость		135 час., 3 зач.ед.			

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

Семестр: 4 (Весенний)

1. Системные вызовы ОС Unix, системное программирование на Си в ОС Unix.

Ядро операционной системы. Основные компоненты ядра. Отличие системных вызовов от обычных функций. Системные вызовы open, creat, close. Команда umask. Системные вызовы read и write.

Этапы жизни процессов. Процессы-зомби. Системный вызов fork. Копирование при записи. Завершение процесса. Exit и _exit. Системные вызовы wait и waitpid. Макросы для обработки статуса завершения процесса. Задача fork.

Формат двоичных выполняемых файлов. Динамические библиотеки. Sbrk. Системные вызовы для загрузки выполняемых файлов. Задача echo hello.

Стандартные потоки ввода/вывода. Системные вызовы dup и dup2. Перенаправление ввода/вывода с помощью dup2. Задача echo hello > file.

Организация pipe-ов из C. Сигнал SIGPIPE. Именованные каналы. Задача echo hello | cat.

Обработка сигналов из языка C. Прерывания и исключения. Реентерабельные функции. Использование стандартной библиотеки в обработчике сигнала. Задача timeout.

Чтение оглавлений из C. Обработка Inodes Программа myls.

Синхронизации процессов. Взаимная блокировка процессов с помощью flock. Задача myshell.

BSD Сокеты. Модель клиент/сервер. UNIX и Internet сокеты. Последовательный сервер и клиент на UNIX сокетах. Задача «дай файл».

Протоколы. Уровни протоколов. MAC адрес. ARP запросы. IP адрес. Маршрутизация. Протоколы TCP и UDP. Команда telnet Сервер на IP сокетах. HTTP протокол. Задача Упрощенный http сервер и упрощенный IE.

Процес init. Процессы демоны. Начальная раскрутка в BSD и System V. Syslog сервер. Конфигурация, вращение логов. Задача «протокол работы в сервере»

Суперсервер. Конфигурация inetd. Getsockname и getpeername. Задача клиент под inetd.

DNS. Домены общего назначения и географические домены. Итеративные и рекурсивные запросы. Команда nslookup. Типы записей в зоне. Прямая и обратная зоны. Функции gethostbyname и gethostbyaddr. Задача

Параллельный сервер. Обработка сигнала SIGCHLD. Задача

System V IPC. Типы объектов. Команды просмотра и удаления объектов. Задача клиент и сервер на очередях сообщений

Сопрограммы. Потоки. Поддержка потоков в ядре. Библиотека pthreads. Создание и завершение потоков. Использование общих переменных. Синхронизация потоков. Задача «параллельный сервер на потоках».

Доступ к файлам с использованием системного вызова mmap. Отображение памяти на устройство /dev/zero. Использование tmpfs для организации общего доступа к памяти. Использование устройства /dev/mem для отображения физической памяти. Когерентность памяти. POSIX семафоры. Использование двоичных семафоров для доступа к общим переменным. Использование семафоров для совместного доступа к буферному пулу. Дeadлоки при использовании с двух семафоров.

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. Оллман, 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)

Обеспечение самостоятельной работы – предполагается наличие у студентов персональных компьютеров с доступом в интернет.

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;
- подготовка к коллоквиумам, дифференцированному зачёту.

ПРИЛОЖЕНИЕ

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

по направлению:	Прикладные математика и физика
профиль подготовки:	Термоядерная энергетика и плазменные технологии Физтех-школа природоподобных, плазменных и ядерных технологий им. И.В. Курчатова кафедра информатики и вычислительных сетей
курс:	2
квалификация:	бакалавр
Семестр, формы промежуточной аттестации: 4 (весенний) - Дифференцированный зачет	
Разработчик:	В.А. Федотов, ассистент

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 «Программные интерфейсы операционной 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.

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

Промежуточный контроль знаний, умений, навыков по дисциплине «Программные интерфейсы операционной системы UNIX» проводится при защите заданий и контрольных работ

Примерный перечень тем заданий и контрольных работ

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”, дожидаться 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 дожидаться 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 «Программные интерфейсы операционной системы UNIX»