

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

УТВЕРЖДАЮ

Проректор по учебной работе и
образовательным инновациям

«» О.М. Чуприс

Регистрационный № УД-5101 уч.



Операционные системы семейства UNIX

Учебная программа учреждения высшего образования
по учебной дисциплине для специальности второй ступени высшего
образования (магистратуры) с углубленной подготовкой специалиста

1-31 81 09 Алгоритмы и системы обработки больших объемов
информации

2018 г.

Учебная программа составлена на основе образовательного стандарта высшего образования ОСВО 1-31 81 09-2014 и учебного плана G31-254/уч. от 26.05.2017.

Составители:

С. А. Соболев – ассистент кафедры дискретной математики и алгоритмики факультета прикладной математики и информатики Белорусского государственного университета.

Рецензенты:

А. О. Сикорский – директор ООО «ЯндексБел»;

С. В. Вабищевич – заведующий кафедрой информатики и методики преподавания информатики Белорусского государственного педагогического университета им. М. Танка, кандидат педагогических наук, доцент.

Рекомендована к утверждению:

Кафедрой дискретной математики и алгоритмики Белорусского государственного университета (протокол № 8 от 18 января 2018 г.);

Научно-методическим Советом Белорусского государственного университета (протокол № 4 от 15 февраля 2018 г.).



ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Учебная программа по учебной дисциплине «Операционные системы семейства UNIX» относится к государственному компоненту цикла дисциплин специальной подготовки и разработана в соответствии с учебным планом и образовательным стандартом второй ступени высшего образования (магистратуры) с углубленной подготовкой специалиста по специальности 1-31 81 09 «Алгоритмы и системы обработки больших объемов информации».

Учебная дисциплина «Операционные системы семейства UNIX» знакомит магистрантов с принципами, заложенными в функционирование UNIX-подобных операционных систем, их архитектурой, основами межпроцессного взаимодействия, базовым администрированием компьютеров под управлением UNIX-систем.

Основой для изучения учебной дисциплины являются следующие учебные дисциплины первой ступени высшего образования: «Программирование», «Операционные системы», «Архитектура компьютеров», «Компьютерные сети». Знания, полученные в учебной дисциплине, используются при изучении учебных дисциплин: «Технологии проектирования и разработки высоконагруженных веб-систем» и «Системы хранения данных».

Цель преподавания учебной дисциплины «Операционные системы семейства UNIX»: создание базы для использования студентами вычислительных систем под управлением UNIX-подобных ОС (в частности, как инфраструктуры в задачах обработки больших объемов данных) и формирование у магистрантов умения использовать технологии и компьютерные системы, основанные на принципах UNIX, при решении задач, связанных с обработкой и анализом данных.

При изложении материала учебной дисциплины важно показать спектр применения UNIX-подобных операционных систем при решении прикладных задач обработки и анализа больших объемов информации, возникающих в различных областях науки, техники, экономики и др., в сравнении с альтернативами по таким критериям, как стоимость, удобство сопровождения, техническая поддержка и др.

Основные задачи, решаемые при изучении учебной дисциплины «Операционные системы семейства UNIX»:

- изучение доступных операционных систем семейства UNIX;
- изучение особенностей использования и администрирования UNIX-подобных систем;
- автоматизация прикладных задач с использованием средств операционной системы.

В результате изучения дисциплины магистрант должен:

знать:

- архитектуру операционной системы UNIX;
- организацию памяти процессов, способы взаимодействия процессов;
- концепции безопасности UNIX;

уметь:

- работать в терминале операционной системы;
- работать с файловой системой;
- реализовывать межпроцессное взаимодействие на языках C/C++;
- управлять пользователями и правами доступа.

владеть:

- компьютерными системами, основанными на принципах UNIX.

Освоение образовательной программы магистратуры должно обеспечить формирование следующих групп компетенций:

академических компетенций – углубленных научно-теоретических, методологических знаний и исследовательских умений, обеспечивающих разработку научно-исследовательских, инновационной деятельности, непрерывного самообразования (АК-2. Методологические знания и исследовательские умения, обеспечивающие решение прикладных задач и инновационной деятельности);

социально-личностных компетенций – личностных качеств и умений следовать социально-культурным и нравственным ценностям; способностей к социальному, межкультурному взаимодействию, критическому мышлению; социальной ответственности, позволяющих решать социально-профессиональные, организационно-управленческие, воспитательные задачи (магистр должен: СЛК-1. Учитывать социальные и нравственно-этические нормы в социально-профессиональной деятельности. СЛК-2. Быть способным к сотрудничеству и работе в команде. СЛК-3. Владеть коммуникативными способностями для работы в междисциплинарной и международной среде. СЛК-4. Совершенствовать и развивать свой интеллектуальный и общекультурный уровень, добиваться нравственного и физического совершенствования своей личности. СЛК-6. Логично, аргументированно и ясно строить устную и письменную речь, использовать навыки публичной речи, ведения дискуссии и полемики. СЛК-7. Проявлять инициативу и креативность, в том числе в нестандартных ситуациях);

профессиональных компетенций – углубленных знаний по специальным дисциплинам и способностей решать сложные профессиональные задачи, задачи научно-исследовательской и научно-педагогической деятельности, разрабатывать и внедрять инновационные проекты, осуществлять непрерывное профессиональное самосовершенствование (магистр должен быть способен:

ПК-1. Квалифицированно использовать современные достижения по разработке и анализу алгоритмов и современные информационные технологии. ПК-3. Разрабатывать эффективные численные алгоритмы и интегрировать их в компьютерные системы. ПК-4. Обосновывать выбор методов и инструментов для решения прикладных задач).

Учебная программа рассчитана на 168 часов, из них 56 аудиторных часов, в том числе 20 лекционных часов, 18 часов лабораторных занятий и 18 часов семинарских занятий.

Форма текущей аттестации – зачет, экзамен.

СОДЕРЖАНИЕ УЧЕБНОГО МАТЕРИАЛА

Раздел I. Введение. Базовые навыки использования UNIX-систем

Тема 1.1. Интерфейс командной строки.

Разновидности командных оболочек. Работа с командной оболочкой в интерактивном режиме. Получение справки по командам. Примеры команд и опций.

Тема 1.2. Стандарт иерархии файловой системы.

Дерево каталогов и монтирование. Специальные типы файлов. Символические и жёсткие ссылки. Стандартные утилиты для работы с файлами и каталогами. Названия и назначение каталогов верхнего уровня.

Раздел II. Командная оболочка

Тема 2.1. Перенаправление ввода-вывода.

Стандартные дескрипторы 0, 1, 2. Именованные каналы. Конвейер. Применение утилиты tee.

Тема 2.2. Командные сценарии.

Базовый синтаксис сценариев: shebang, переменная окружения PATH, запуск процессов, коды выхода. Использование переменных. Получение аргументов командной строки. Строковые литералы, экранирование. Обработка ошибок.

Тема 2.3. Программирование в командной оболочке.

Подстановка параметров (parameter substitution). Проблема разбиения на слова (word splitting). Подстановка вывода команд (command substitution). Условия, циклы, функции, целочисленная математика. Использование source. Управление заданиями (job control). Подстановка процесса (process substitution). Использование шаблонов (globbing). Утилиты xargs и parallel.

Тема 2.4. Обработка текстовой информации.

Работа с текстом при помощи стандартных утилит UNIX. Регулярные выражения. Команды семейства grep. Команды cut, paste, sed, awk и др.

Раздел III. Средства разработки

Тема 3.1. Компиляция и компоновка кода на языке C.

Семейство компиляторов GCC. Символы. Программы, статические и динамические библиотеки. Формат ELF.

Тема 3.2. Средства сборки программ.

Программа make. Классический подход к сборке ПО с открытым исходным кодом.

Тема 3.3. Средства отладки и анализа программ.

Дампы памяти. Использование инструментов GDB и valgrind.

Тема 3.4. Методы развёртывания программ.

Системы управления пакетами. Debian-пакеты, утилиты apt и dpkg.

Раздел IV. Пользователи и права доступа

Тема 4.1. Пользователи и группы.

Конфигурационные файлы с данными о пользователях, хранение паролей. Суперпользователь (root). Различия между su и sudo.

Тема 4.2. Система разграничения прав доступа.

Команды редактирования прав доступа к файлам и каталогам. Биты setuid и setgid, sticky-бит для каталогов. Маска umask.

Раздел V. Процессы

Тема 5.1. Понятие процесса.

Идентификатор процесса. Просмотр списка запущенных процессов. Файловая система procfs. Статусы процессов. Приоритеты процессов и планирование. Назначение процессов на ядра. Процесс init. Уровни выполнения (runlevels). Современные системные менеджеры.

Тема 5.2. Сигналы.

Средства командной строки для посылки сигналов процессам. Реализация обработчика сигнала на языке C.

Тема 5.3. Порождение процессов.

Системный вызов fork. Процессы-зомби и процессы-сироты. Техника fork — exec. Запуск произвольной программы в UNIX. Неименованные каналы (pipes). Проблема обнаружения ошибки вызова exec при создании нового процесса и метод её решения.

Тема 5.4. Потoki.

Стандарт POSIX Threads. Ручной запуск потока. Основы pthreads. Создание и завершение потока. Мьютексы и условные переменные. Устройство примитивов синхронизации. Атомарные операции. Механизм futex.

Раздел VI. Управление памятью

Тема 6.1. Понятие виртуальной памяти.

Механизм копирования при записи (copy-on-write). Ситуации отказа страниц (page faults).

Тема 6.2. Классификация виртуальной памяти.

Память private и shared, anonymous и file-backed. Анализ показаний программы htop. Ввод-вывод посредством отображения файлов в память. Разделяемая память: способы создания, использование для межпроцессного взаимодействия.

Тема 6.3. Доступная память в системе.

Учёт свободной памяти командой free. Кеширование дисковых операций, сброс кешей. Ситуация нехватки памяти в системе. Swap-раздел.

Раздел VII. Сетевые возможности

Тема 7.1. Использование сетей.

Прикладные программы для работы с сетью: nc, telnet, wget, curl, ...

Тема 7.2. Администрирование сетей.

Настройка сети через командную строку. UNIX-система на сервере в локальной сети: DNS, DHCP, веб-сервер, интернет-шлюз.

Тема 7.3. Программирование сетевых приложений.

Понятие сокета. Поточковые и датаграммные сокеты. Реализация клиента и сервера для взаимодействия через сокеты. Протоколы IPv4 и IPv6. Сокеты домена UNIX и их особенности. Дополнительные возможности (неблокирующий ввод-вывод, системный вызов select).

Раздел VIII. Хранение данных

Тема 8.1. Жёсткие диски.

Система адресации, разбиение на разделы, MBR и GPT. Технологии хранения данных RAID и LVM.

Тема 8.2. Устройство файловых систем.

Индексные дескрипторы (inodes). Примеры файловых систем. Сетевые диски. Диски в оперативной памяти. Механизм FUSE.

Раздел IX. Внутреннее устройство ОС

Тема 9.1. Ядро Linux.

Общая характеристика и структура. Нумерация версий. Модули ядра. Создание простейшего модуля.

Тема 9.2. Загрузка ОС.

Этапы загрузки компьютера. Загрузчики.

Тема 9.3. Сборка ядра.

Конфигурирование, сборка и установка «ванильного» ядра Linux.

УЧЕБНО-МЕТОДИЧЕСКАЯ КАРТА УЧЕБНОЙ ДИСЦИПЛИНЫ

Номер раздела, темы	Название раздела, темы	Количество аудиторных часов			УСР	Форма контроля знаний
		Лекции	Практические и семинарские занятия	Лабораторные занятия		
1	2	3	4	5	6	7
1	Базовые навыки использования UNIX-систем	2	2	2		
1.1	Интерфейс командной строки		1			Устный опрос
	<i>Лабораторная работа 1. Установка Linux</i>			1		Защита лабораторной работы
1.2	Стандарт иерархии файловой системы	1	1			Устный опрос
	<i>Лабораторная работа 2. «Матрешка»</i>			1		Защита лабораторной работы
2	Командная оболочка	2	6	4		
2.1	Перенаправление ввода-вывода	1	1			Устный опрос
2.2	Командные сценарии	1	1			Устный опрос
2.3	Программирование в командной оболочке		2			Устный опрос
	<i>Лабораторная работа 3. Сценарии командной оболочки</i>			2		Защита лабораторной работы
2.4	Обработка текстовой информации		2			Контрольная работа 1.
	<i>Лабораторная работа 4. Работа с текстом</i>			2		Защита лабораторной работы
3	Средства разработки	2	2	2		
3.1	Компиляция и компоновка кода на языке C	1	1			Устный опрос
	<i>Лабораторная работа 5. Сборка статических и динамических библиотек</i>			2		Защита лабораторной работы
3.2	Средства сборки программ	1				Устный опрос
3.3	Средства отладки и анализа программ		1			Контрольная работа 2.
4	Пользователи и права доступа	2				
4.1	Пользователи и группы	1				Устный опрос
4.2	Система разграничения прав доступа	1				Устный опрос

5	Процессы	4	2	5		
5.1	Понятие процесса	1				Устный опрос
5.2	Сигналы	1				Устный опрос
5.3	Порождение процессов	1	1			Устный опрос
	<i>Лабораторная работа 6. Запуск процессов</i>			1		Защита лабораторной работы
	<i>Лабораторная работа 7. Реализация конвейера</i>			3		Защита лабораторной работы
5.4	Потоки	1	1			Контрольная работа 3.
	<i>Лабораторная работа 8. Использование библиотеки pthreads</i>			1		Защита лабораторной работы
6	Управление памятью	2	2	1		
6.1	Понятие виртуальной памяти	2				Устный опрос
6.2	Классификация виртуальной памяти		1			Устный опрос
	<i>Лабораторная работа 9. Выделение памяти</i>			1		Защита лабораторной работы
6.3	Доступная память в системе		1			Устный опрос
7	Сетевые возможности	2	2	2		
7.1	Использование сетей		1			Устный опрос
7.2	Администрирование сетей		1			Устный опрос
7.3	Программирование сетевых приложений	2				Контрольная работа 4.
	<i>Лабораторная работа 10. Разработка клиент-серверного приложения</i>			2		Защита лабораторной работы
8	Хранение данных	2				
8.1	Жёсткие диски	1				Устный опрос
8.2	Устройство файловых систем	1				Устный опрос
9	Внутреннее устройство ОС	2	2	2		Коллоквиум
9.1	Ядро Linux	1	1			Устный опрос
	<i>Лабораторная работа 11. Разработка модуля ядра</i>			2		Защита лабораторной работы
9.2	Загрузка ОС	1				Устный опрос
9.3	Сборка ядра		1			Итоговый тест
ИТОГО		20	18	18		

ИНФОРМАЦИОННО-МЕТОДИЧЕСКАЯ ЧАСТЬ

Рекомендуемая литература

Основная

1. *Далхаймер М.* Запускаем Linux / М. Далхаймер, М. Уэлш. – М.: Символ-Плюс, 2012. – 992 с.
2. *Керниган Б.* UNIX. Программное окружение / Б. Керниган, Р. Пайк. – М.: Символ-Плюс, 2012. – 416 с.

Дополнительная

1. *Таненбаум Э.* Операционные системы. Разработка и реализация / Э. Таненбаум, А. Вудхалл – 3-е изд. – СПб.: Питер, 2007. – 704 с.
2. *Kerrisk M.* The Linux Programming Interface / Michael Kerrisk. – San Francisco: No Starch Press, Inc., 2010. – 1556 p.
3. *Моли Б.* Unix/Linux. Теория и практика программирования. – М.: КУДИЦ-Образ, 2004. – 576 с.
4. *Стивенс Р.* UNIX. Профессиональное программирование / Р. Стивенс, С. Раго. – 2-е изд. – СПб.: Символ-Плюс, 2007. – 1040 с.
5. *Керниган Б.* Язык программирования C = The C programming language / Б. Керниган, Д. Ритчи. – 2-е изд. – М.: Вильямс, 2009. – 304 с.

Перечень рекомендуемых средств диагностики

Для аттестации обучающихся на соответствие их персональных достижений поэтапным и конечным требованиям образовательной программы создаются фонды оценочных средств, включающие типовые задания, контрольные работы и тесты. Оценочными средствами предусматривается оценка способности обучающихся к творческой деятельности, их готовность вести поиск решения новых задач.

На лекционных занятиях возможно использование элементов проблемного обучения: проблемное изложение некоторых аспектов, использование частично-поискового метода.

Для текущего контроля качества усвоения знаний магистрантами используется следующий диагностический инструментарий:

1. Устная форма: устные опросы, устная защита заданий при выполнении магистрантами лабораторных работ, проведение коллоквиума «Операционные системы семейства UNIX».
2. Письменная форма: отчеты по лабораторным работам, итоговое тестирование, письменные контрольные работы по отдельным темам курса.

Рекомендуемая тематика контрольных работ

- Контрольная работа №1. Сценарии командной оболочки
 - Контрольная работа №2. Составление сценария сборки программы на С.
 - Контрольная работа №3. Разработка многопоточной программы.
 - Контрольная работа №4. Сетевое взаимодействие.
3. Техническая форма: система автоматического тестирования InsightRunner (www.acm.bsu.by) и система AnyTask (<https://anytask.org/school/bsu>) – инструменты с эффективной функциональностью контроля, тренинга и самостоятельной работы.

Текущая аттестация предусматривает проведение зачета и экзамена. При этом рекомендуется использовать оценивание успеваемости на основе модульно-рейтинговой системы, дающей возможность проследить и оценить динамику процесса достижения целей обучения. Рейтинговая оценка предусматривает использование весовых коэффициентов для текущего контроля знаний и текущей аттестации студентов по дисциплине. Рейтинговая оценка по дисциплине рассчитывается на основе оценки текущей успеваемости и экзаменационной оценки с учетом их весовых коэффициентов:

- 1) оценка по текущей успеваемости – 0.3,
- 2) экзаменационная оценка – 0.7.

Итоговая оценка формируется на основе:

1. Правил проведения аттестации студентов (Постановление Министерства образования Республики Беларусь № 53 от 29 мая 2012 г.);
2. Положения о рейтинговой системе оценки знаний по дисциплине в БГУ (Приказ ректора БГУ от 18.08.2015 № 382-ОД);
3. Критериев оценки знаний студентов (письмо Министерства образования от 22.12.2003г.).

Примерный перечень заданий к зачёту

1. Сценарии командной оболочки (часть А)

Необходимо реализовать простой сценарий для решения поставленной задачи (по вариантам). Это индивидуальное задание базового уровня по работе с командной оболочкой (bash). В задание входит работа с параметрами командной строки, ввод и вывод текста, печать справки, вызов стандартных утилит.

2. Сценарии командной оболочки (часть В)

Необходимо запрограммировать сложный сценарий с использованием ветвлений, циклов, функций. Это индивидуальное задание повышенного уровня сложности.

Задание затрагивает следующие темы: работа с утилитами `awk`, `sed`, GNU `Make`; модификация конфигурационного файла `.bashrc`, настройка приглашения командной оболочки.

3. *fork u vfork*

Необходимо сравнить производительность системных вызовов `fork()` и `vfork()`. Требуется выполнить многократный запуск утилиты `/bin/true` (или какой-либо другой на усмотрение студента) с ожиданием её завершения, измерить общее время. Определить, сколько времени занимает один запуск программы в обоих случаях.

4. *012345*

Дается некорректный код на языке C, в котором реализован вывод чисел из разных потоков. При помощи примитивов синхронизации требуется организовать попеременную работу двух потоков так, чтобы в результате получилась последовательность 0, 1, 2, 3, 4, 5, ...

5. *Pipeline*

Необходимо реализовать программу, которая выполняла бы роль конвейера (`pipeline`) из командной оболочки. Программа должна запускать новые дочерние процессы с указанными аргументами и связывать их между собой, осуществляя перенаправление ввода-вывода: то, что выводит на поток стандартного вывода предыдущий процесс, попадает в поток стандартного ввода следующего процесса.

6. *Guess the Number*

Нужно написать программу (клиент), которая будет играть с предоставленной программой (сервером) в игру «Угадай число». Межпроцессное взаимодействие осуществляется посредством сокета домена UNIX. В качестве дополнительного задания предлагается также реализовать аналогичную программу, работающую по протоколу TCP.

7. */dev/nulll*

Требуется реализовать внешний модуль для ядра Linux, который создаёт новое символьное устройство под названием `/dev/nulll` (три латинские буквы L — от «`/dev/null Limited Edition`»). По смыслу оно должно работать так же, как стандартное нулевое устройство. Но с двумя дополнительными возможностями: должен осуществляться учёт размера записанных данных и должно поддерживаться ограничение «места на диске».

Методические рекомендации по организации самостоятельной работы обучающихся

Для организации самостоятельной работы магистрантов по учебной дисциплине следует использовать современные информационные технологии: разместить в сетевом доступе комплекс учебных и учебно-методических материалов (учебно-программные материалы, учебное издание для теоретического изучения дисциплины, методические указания к лабораторным занятиям, материалы текущего контроля и текущей аттестации, позволяющие определить соответствие учебной деятельности обучающихся требованиям образователь-

ных стандартов высшего образования и учебно-программной документации, в т.ч. вопросы для подготовки к зачету, задания, тесты, вопросы для самоконтроля, список рекомендуемой литературы, информационных ресурсов и др.).

Условия для самостоятельной работы магистрантов, в частности для развития навыков самоконтроля, способствующих интенсификации учебного процесса, обеспечиваются наличием легко разворачиваемой виртуальной UNIX-подобной среды.

Эффективность самостоятельной работы студентов проверяется в ходе текущего и итогового контроля знаний. Для общей оценки качества усвоения студентами учебного материала рекомендуется использование рейтинговой системы.

ПРОТОКОЛ СОГЛАСОВАНИЯ УЧЕБНОЙ ПРОГРАММЫ

Название учебной дисциплины, с которой требуется согласование	Название кафедры	Предложения об изменениях в содержании учебной программы учреждения высшего образования по учебной дисциплине	Решение, принятое кафедрой, разработавшей учебную программу (с указанием даты и номера протокола)
Технологии проектирования и разработки высоконагруженных веб-систем	Кафедра ДМА	Нет	Оставить содержание учебной дисциплины без изменения, протокол № 8 от 18.01.2018 г.
Системы хранения данных	Кафедра ДМА	Нет	Оставить содержание учебной дисциплины без изменения, протокол № 8 от 18.01.2018 г.

ДОПОЛНЕНИЯ И ИЗМЕНЕНИЯ К УЧЕБНОЙ ПРОГРАММЕ

на ____ / ____ учебный год

№№ ПП	Дополнения и изменения	Основание

Учебная программа пересмотрена и одобрена на заседании кафедры дискретной математики и алгоритмики (протокол № ____ от _____ 201_ г.)

Заведующий кафедрой

(ученая степень, звание)

(подпись)

(И. О. Фамилия)

УТВЕРЖДАЮ
Декан факультета

(ученая степень, звание)

(подпись)

(И. О. Фамилия)