

Министерство науки и высшего образования Российской Федерации  
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ (НИ ТГУ)

Научно-образовательный центр «Высшая ИТ школа»

УТВЕРЖДЕНО:  
Исполнительный директор НОЦ ВИТШ

Т.С.Кетова

Рабочая программа дисциплины

**Основы системного администрирования  
(ОСА)**

по направлению подготовки  
**09.03.04 (33.04) Программная инженерия**

Направленность подготовки:  
**«Программная инженерия»**

Форма обучения  
**Очная**

Квалификация  
**Программный инженер**

Год приема  
**2023**

СОГЛАСОВАНО:  
Руководитель ОП  
О.А.Змеев

Председатель УМК  
Д.О. Змеев

Томск – 2024

## 1. Цель и планируемые результаты освоения дисциплины (модуля)

Целью освоения дисциплины является формирование следующих компетенций:

БК-1	Способен применять общие и специализированные компьютерные программы при решении задач профессиональной деятельности	Знает: правила и принципы применения общих и специализированных компьютерных программ для решения задач профессиональной деятельности Умеет: применять современные IT-технологии для сбора, анализа и представления информации; использовать в профессиональной деятельности общие и специализированные компьютерные программы
БК-7	Способен создавать программное обеспечение на основе предоставленных требований к дизайну, функциональности, безопасности, с использованием стандартных подходов, библиотек, инструментов контроля версий	Знает: Принципы оценки характеристик программного обеспечения. Правила, языки и методы фиксации требований к программному обеспечению, архитектуре или ожидаемому поведению, на уровне чтения технической и проектной документации. Основные принципы безопасной разработки приложений. Теорию использования и работы с хранилищами данных, протоколами передачи данных, алгоритмы обработки и кодирования данных  Умеет: Модифицировать программный код приложения с целью исправления дефектов, расширения функциональности, или при изменении требований к приложению; разрабатывать программное обеспечение совместно с другими членами команды разработки; использовать инструменты для совместной разработки и развертывания приложений; проверять приложение на дефекты; проверять критерии готовности приложения для практического использования или передачи другой команде разработчиков; оценивать требования запущенного приложения к среде его функционирования и развертывания
ОПК-5	Способен использовать технологии, методы, инструменты и процессы для поддержки процессов введения программного обеспечения в эксплуатацию, а также внесение модификаций и обновления программного обеспечения уже внедренной в эксплуатацию системы.	Знает: Основные особенности влияния аппаратных систем, операционных систем, инфраструктурного окружения на работу и функционирование программного обеспечения. Основы процесса развертывания различных компонентов информационных систем, а также основные инструменты применяемые для этого процесса. Принципы функционирования программного обеспечения в рамках систем виртуализации и контейнеризации Умеет: Оказывать помощь в реализации процессов и планов обслуживания программного обеспечения и вносить изменения в программное обеспечение для реализации потребностей и запросов на обслуживание; следовать установленному процессу внесения изменений во внедренную и функционирующую систему; изучать данные мониторинга систем для выявления закономерностей, аномалий и потенциальных проблем;

сопоставлять данные из различных источников для диагностики проблем, проводить анализ первопричин и предлагать соответствующие решения или оптимизации;  
использовать технологии и инструменты контейнеризации и виртуализации для соблюдения процесса внедрения и обслуживания систем;  
определять предварительные оценки по сложности, стоимости, времени внесения предлагаемых изменений в функционирующую программную систему

## **2. Задачи освоения дисциплины**

- Получить общее представление о работе системного администратора (инженера) и специфики эксплуатации информационных систем;
- Освоить основы сетевых технологий и работу с \*nix-подобными операционными системами на примере Ubuntu;
- Овладеть практическими навыками автоматизации тестирования, сборки и развертывания программных компонент;
- Узнать основы контейнеризации и оркестрации приложений.

## **3. Место дисциплины (модуля) в структуре образовательной программы**

Дисциплина относится к Блоку 1 «Дисциплины (модули)» обязательной части образовательной программы. Для внесения оценок в зачетные книжки обучающихся принимается сокращенное название дисциплины «ОСА».

## **4. Семестр(ы) освоения и форма(ы) промежуточной аттестации по дисциплине** Семестр 3, Зачет с оценкой

## **5. Входные требования для освоения дисциплины**

Для успешного освоения дисциплины требуются компетенции, сформированные в ходе освоения образовательных программ предшествующего уровня образования.

## **6. Язык реализации**

Русский

## **7. Объем дисциплины (модуля)**

Общая трудоемкость дисциплины составляет 4 з.е., 144 часов, из которых:

- лекции: 30.0 ч.;
- практические занятия: 16.0 ч.

Объем самостоятельной работы студента определен учебным планом.

## **8. Содержание дисциплины (модуля), структурированное по темам**

Тема 1. Введение в операционную систему Linux и сетевые технологии.

Тема 2. Основы контейнеризации с использованием технологии Docker.

Тема 3. Автоматизация сборки и развертывания программного кода (CI/CD).

Тема 4. Продвинутое топики системного администрирования.

## **9. Текущий контроль по дисциплине**

Текущий контроль по дисциплине проводится путем контроля посещаемости,

проведения контрольных работ, тестов по лекционному материалу, выполнения домашних заданий, и фиксируется в форме контрольной точки не менее одного раза в семестр.

### 10. Порядок проведения и критерии оценивания промежуточной аттестации

1. Основу дисциплины составляют три лабораторно-практические (самостоятельные) работы. За выполнение каждой работы выставляется оценка от 2 до 5 баллов (0 за списывание или иное нечестное поведение студента и 2 за несданную работу). Часть работ подразумевает автоматическую или полуавтоматическую проверку.
2. Итоговая оценка рассчитывается как среднее арифметическое от оценок за первые три модуля с округлением по данной шкале:

Диапазон оценки	Итоговая оценка	Результат аттестации
> 5.3	5+	отлично
5 - 5.3	5	отлично
4.7 - 5	5-	отлично
4.3 - 4.7	4+	хорошо
4 - 4.3	4	хорошо
3.7 - 4	4-	хорошо
3.3 - 3.7	3+	удовлетворительно
3 - 3.3	3	удовлетворительно
2.7 - 3	3-	удовлетворительно
2.3 - 2.7	2+	неудовлетворительно
2 - 2.3	2	неудовлетворительно
< 2	2-	неудовлетворительно

*Например, если студент набрал 3.3, 0 и 5 баллов за три модуля, то ему будет выставлена итоговая оценка 3- (удовлетворительно) по расчёту:  $(3.3 + 0 + 5) / 3 = 2.76$*

3. Если студент не успевает сдать работу в том модуле, в котором она опубликована, то при сдаче в каждом последующем модуле оценка будет выставлена сниженная оценка с коэффициентом из данной таблицы (но не ниже оценки 2):

Длительность просрочки сдачи работы	Коэффициент оценивания
0 модулей	1
1 модуль	0.9
2 модуля	0.75

3 модуля	0.5
----------	-----

*Например, если студент получил оценку 5- (эквивалент 4.7), но сдал работу на модуль позже её публикации, то за лабу он получит  $4.7 * 0.9 = 4.23$  баллов*

4. В четвертом модуле можно досдавать долги и/или получить дополнительные (бонусные) баллы за выполнение заданий повышенной сложности, публикуемых в начале данного модуля. За них можно получить до 1.5 бонусных баллов (будут прибавлены к текущей оценке за прошлые модули).
5. Студенты, которые пропускали занятия по уважительной причине, могут договориться с преподавателем об отсрочке сдачи лабораторных работ или альтернативном задании.

Пример лабораторной работы:

*Перед вами стоит задача реализовать CI/CD пайплайн для вашего собственного приложения (если у вас такого нет, то можете выбрать одно из представленных в задании), используя технологию GitLab CI или GitHub Actions.*

*Данный пайплайн должен включать несколько шагов:*

1. Автоматизированная сборка на выделенном/общем раннере
2. Запуск линтера и unit-тестов для исходного кода
3. Развертывание сервиса на виртуальной машине

*Каждый выполненный шаг добавляет по одному баллу к вашей оценке за работу.*

*При выполнении оценивается качество кода и документации, техническая сложность подходов к решению задачи, а также полнота понимания использованных для этого инструментов.*

## **11. Учебно-методическое обеспечение**

- а) Оценочные материалы текущего контроля и промежуточной аттестации.
- б) План лекционных, семинарских и практических занятий по дисциплине.
- в) Правила дисциплины, включающие методические указания по проведению групповых лабораторных работ и организации самостоятельной работы студентов.

## **12. Перечень учебной литературы и ресурсов сети Интернет**

а) основная литература:

- Замятина, О. М. Вычислительные системы, сети и телекоммуникации. Моделирование сетей : учебное пособие для магистратуры / О. М. Замятина. — Москва : Издательство Юрайт, 2018. — 159 с. — (Университеты России). — ISBN 978-5-534-00335-2. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/414064> (дата обращения: 14.02.2023).
- Операционная система Linux: Курс лекций. Учебное пособие / Г. В .Курячий, К. А. Маслинский — М. : ALT Linux; Издательство ДМК Пресс, 2010. — 348 с. : ил. ; 2-е изд., исправленное.— (Библиотека ALT Linux). — URL: <https://docs.altlinux.org/books/altlibrary-linuxintro2.pdf> (дата обращения: 14.02.2023).
- Kocher, Parminder Singh. Microservices and Containers / Parminder Singh Kocher. — Addison-Wesley Professional, 2018. — 304с.

### 13. Перечень информационных технологий

а) лицензионное и свободно распространяемое программное обеспечение:

– Microsoft Office Standart 2013 Russian: пакет программ. Включает приложения: MS Office Word, MS Office Excel, MS Office PowerPoint, MS Office OneNote, MS Office Publisher, MS Outlook, MS Office Web Apps (Word Excel MS PowerPoint Outlook);

– программа для виртуализации VirtualBox – <https://www.virtualbox.org/>

– платформа для контейнеризации Docker Engine для операционной системы Ubuntu – <https://docs.docker.com/engine/install/ubuntu/>

– платформы для хранения, версионирования и распространения исходного кода GitLab и GitHub – <https://gitlab.com/> и <https://github.com/>

– публично доступные облачные технологии (Google Docs, Яндекс диск и т.п.).

б) информационные справочные системы:

– Электронный каталог Научной библиотеки ТГУ – <http://chamo.lib.tsu.ru/search/query?locale=ru>  
"http://chamo.lib.tsu.ru/search/query?locale=ru&theme=system"&

"http://chamo.lib.tsu.ru/search/query?locale=ru&theme=system"theme=system

– Электронная библиотека (репозиторий) ТГУ – <http://vital.lib.tsu.ru/vital/access/manager/Index>

– ЭБС Лань – <http://e.lanbook.com/>

– ЭБС Консультант студента – <http://www.studentlibrary.ru/>

– Образовательная платформа Юрайт – <https://urait.ru/>

– ЭБС ZNANIUM.com – <https://znanium.com/>

– ЭБС IPRbooks – <http://www.iprbookshop.ru/>

### 14. Материально-техническое обеспечение

Аудитории для проведения занятий лекционного типа.

Аудитории для проведения занятий семинарского типа, индивидуальных и групповых консультаций, текущего контроля и промежуточной аттестации.

Помещения для самостоятельной работы, оснащенные компьютерной техникой и доступом к сети Интернет, в электронную информационно-образовательную среду и к информационным справочным системам.

### 15. Информация о разработчиках

Зоркин Александр Сергеевич,  
старший системный инженер  
ООО «Тинькофф Центр Разработки»

Куприянов Александр Андреевич,  
ассистент учебного офиса НОЦ ВИТШ