

# C++ и UNIX системы

**Лекторы:**

Иван Маслов



**Язык:**

Русский

**Трудоемкость:**

4 з.е.

**Форма контроля:**

Экзамен

**Образовательная программа:**

Теоретическая и экспериментальная физика

6 семестр

**Пререквизиты:**

Информатика

| Лекции<br>(ак.час)*               | Практические занятия<br>(ак.час) | Лабораторные занятия<br>(ак.час) |
|-----------------------------------|----------------------------------|----------------------------------|
| 32                                | 32                               |                                  |
| *1 академический час = 45 минутам |                                  |                                  |

Данный курс посвящен программированию на языках C и C++, а также работе в UNIX подобных системах. Особое внимание уделяется формированию базовых навыков разработки и проектирования ПО. Слушатели получат возможность взглянуть на процесс разработки изнутри, стать настоящим C++ разработчиком, используя современные технологии и средства. Также слушатели смогут прочувствовать всю строгость дедлайнов и нагрузку в виде лабораторных занятий.

## Содержание курса

### 6 семестр

#### C++ и UNIX системы

#### Структура курса

| Разделы   | Лекции<br>(ак. ч.) | Лаб.<br>(ак.ч.) |
|---|--------------------|-----------------|
| <b>1. UNIX HISTORY / ARCH / SSTRUCT</b>   |                    |                 |
| История UNIX, архитектура ЭВМ, структура linux  | 2                  |                 |
| <b>2. UNIX USER / FILE / PERMISSIONS</b>  |                    |                 |
| Пользователи, группы, права доступа   | 2                  |                 |
| <b>3. UNIX VIRT / CONT / GIT</b>  |                    |                 |
| Виртуализация и контейнеризация (Virtual Box & Docker). GIT, CI/CD(лекция)  | 2                  |                 |
| <b>4. C++ HISTORY</b>   |                    |                 |
| История C, C++, перспективы развития. Плюсы и минусы C++ в сравнении с Python. Виды интерфейсов (CLI vs GUI vs API). Архитектуры проектов (монолитная, микросервисная, сервисная)   | 2                  | 4               |
| <b>5. C++ BUILD / CODE STYLE</b>  |                    |                 |
| Этапы сборки проекта (make, gcc). Стиль программирования (camel case, snake case и т.д.)  | 2                  | 4               |
| <b>6. C++ TYPES / IF / LOOP</b>   |                    |                 |
| Типы данных, алгоритмы, логические выражения, условия, циклы, процедуры, функции, рекурсия  | 2                  |                 |
| <b>7. C++ PRACTICE LINK / POINTER / VAR / ARGS</b>  |                    |                 |
| Практическое занятие (без презентации, 2 пары) по использованию ссылок, указателей, переменных. Переопределение типов. Операторы и операнды. * и &. А также разбор входных аргументов. Заголовочные файлы и исходные файлы. Структуры struct. Списки, массивы. Объектный код. Линковка. | 2                  | 4               |
| <b>8. C++ CONST / GLOBAL / ENUM</b>   |                    |                 |
| Константы, переменные (глобальные, локальные), пространства имен, перечисления enum.  | 2                  |                 |
| <b>9. C++ AUTODOC / LOGGING / PREPROCESS</b>  |                    |                 |
| Система автодокументирования кода (doxygen). Логирование. #ifndef #define #endif  | 2                  |                 |
| <b>10. C++ MULTIPROCESSING / PARALLEL</b>   |                    |                 |
| Многопоточное, многопроцессное/многопроцессорное программирование. Передача данных: PIPE, очередь, шина, JSON, XML. Устройства ввода/вывода (экран, клавиатура, мышь, файл)   | 2                  | 4               |
| <b>11. C++ STL</b>  |                    |                 |
| Стандартная библиотека шаблонов STL   | 2                  |                 |
| <b>12. C++ BOOST / ROOT</b>   |                    |                 |
| Библиотеки решения задач линейной алгебры (Boost.uBLAS , ROOT)  | 2                  |                 |
| <b>13. C++ ENCODE / DECODE</b>  |                    |                 |
| Енкодирование/декодирование текста  | 2                  |                 |
| <b>14. C++ OBJECT / CLASS</b>   |                    |                 |
| Объектно-ориентированное программирование (ООП): история, классы, объекты, наследование, защищенность, виртуальность  | 2                  | 4               |
| <b>15. C++ IMPLEMENT / POLYMORPHISM</b>   |                    |                 |
| ООП: Перегрузка, полиморфизм, сложный полиморфизм, операторы  | 2                  |                 |
| <b>16. C++ TEMPLATES</b>  |                    |                 |

|  |   |   |
|--|---|---|
| ООП: Шаблоны, исключения   | 2 |   |
| <b>17. C++ STATIC LIB / DYNAMIC LIB</b>  |   |   |
| Статические и динамические библиотеки. Вызов C из python   | 2 | 4 |
| <b>18. C++ API / REST / SOAP</b>   |   |   |
| API: HTTP, HTTPS, REST, SOAP, RPC (  | 2 |   |
| <b>19.C++ 11 AUTO / LYAMBDA</b>  |   |   |
| C++ 11: auto, nullptr, range-based циклы, override и final, интеллектуальные указатели, лямбда, non-member begin() и end() | 2 |   |
| <b>20.C++ 14 OPTIONAL / DYNARRAY</b>   |   |   |
| C++ 14: лямбда-выражения, шаблоны переменных, exchange, make_unique, обособленные строки, optional, dynarray               | 2 | 4 |
| <b>21.C++ 17 TEMPLATE&lt;AUTO&gt;</b>  |   |   |
| C++ 17: свертка, template<auto>, constexpr if (лекция)C++ 17: свертка, template<auto>, constexpr if                        | 2 |   |
| <b>22. C++ MODULES</b>   |   |   |
| C++ 20: модули   | 2 |   |

Перечень лабораторных работ:

1. UNIX знакомство: useradd, nano, chmod, Docker, GIT, CI, CD
2. C++ и Python сравнение производительности: Компиляция, функции, аргументы, строки, сравнение скоростных характеристик C++ и Python
3. C++ алгоритмы: CLI Калькулятор вещественных чисел +, -, ^, . Реализация с использованием только функций, условий, циклов, + и -. Вид команд в консоли: calc plus / minus / power; Ханойская башня, результат корректной последовательности записать в двусвязный список. Использовать new и delete.
4. C++ многопоточность: Распределенные математические вычисления (Boost.uBLAS, JSON, XML, PIPE)
5. C++ ООП: Объектное моделирование математических задач
6. C++ web socket: Сетевое взаимодействие по HTTP, REST, SOAP
7. C++ и Python: Интеграция C++ вычислений в Python

## Рекомендуемые ресурсы

1. Страуструп. Программирование. Принципы и практика с использованием C++ 2-е изд. : Пер. с англ. — М.: ООО И.Д. Вильямс, 2016.— ISBN: 978-5-8459-1949-6. <https://b-ok.org/book/2656798/357f29>
2. Стивен Прата. Язык программирования C++. Лекции и упражнения 6-е изд. : Пер. с англ. — М.: ООО И.Д. Вильямс, 2015.— ISBN: 978-5-8459-1950-2. <https://b-ok.org/book/3297781/58aa94>
3. Яцек Галовиц. C++17 STL. Стандартная библиотека шаблонов 1-е изд. : Пер. с англ. — М.: Питер, 2018.— ISBN: 978-5-4461-0680-6. <https://ru.b-ok.global/book/4987315/bf144e>

## Политика оценивания

**Оценочные средства дисциплины: лабораторная работа, письменный зачет.**

В течение семестра необходимо выполнить и защитить 7 лабораторных работ.

В конце семестра проводится письменный экзамен с оценкой (студенту необходимо ответить на два вопроса, при желании повысить итоговую оценку необходимо ответить на доп. вопросы).

Оценку «5» автомат можно получить тому студенту, который защитит все ЛБ работы перед окончанием последней лекции семестра.

Оценку «3» автомат можно получить том студенту, который защитит все ЛБ работы перед экзаменом.

**Оценка формируется по пятибалльной шкале, где:**

**Оценка 5 - «Отлично»** – обучающийся глубоко и прочно усвоил весь программный материал, исчерпывающе, последовательно, грамотно и логически стройно его излагает, не затрудняется с ответом при видеоизменении задания, свободно справляется с задачами и практическими заданиями, правильно обосновывает принятые решения, умеет самостоятельно обобщать и излагать материал, не допуская ошибок.

**Оценка 4 - «Хорошо»** – обучающийся твердо знает программный материал, грамотно и по существу излагает его, не допускает существенных неточностей в ответе на вопрос, может правильно применять теоретические положения и владеет необходимыми умениями и навыками при выполнении практических заданий.

**Оценка 3 - «Удовлетворительно»** – обучающийся усвоил только основной материал, но не знает отдельных деталей, допускает неточности, недостаточно правильные формулировки, нарушает последовательность в изложении программного материала и испытывает затруднения в выполнении практических заданий.

**Оценка 2 - «Неудовлетворительно»** – обучающийся не знает значительной части программного материала, допускает существенные ошибки, с большими затруднениями выполняет практические задания, задачи.

