

Белорусский государственный университет

УТВЕРЖДАЮ

Проректор по учебной работе
и образовательным инновациям

О.Н. Здрок



2020 г.

Регистрационный № УД-8650/уч.

**РАЗРАБОТКА ВЫСОКОПРОИЗВОДИТЕЛЬНЫХ ПРИЛОЖЕНИЙ
НА С И С++**

**Учебная программа учреждения высшего образования
по учебной дисциплине для специальности:**

1-31 80 09 Прикладная математика и информатика
профилизация

Алгоритмы и системы обработки больших данных

2020 г.

Учебная программа составлена на основе ОСВО 1-31 80 09-2019 и учебного плана G31-072/уч. от 11.04.2019 г.

СОСТАВИТЕЛИ:

С. А. Соболев, старший преподаватель кафедры дискретной математики и алгоритмики Белорусского государственного университета;

Д. Ю. Косицин, старший преподаватель кафедры дискретной математики и алгоритмики Белорусского государственного университета;

РЕЦЕНЗЕНТЫ:

А.Ю. Рыжиков – программист ООО «Фитбит Бел».

РЕКОМЕНДОВАНА К УТВЕРЖДЕНИЮ:

Кафедрой дискретной математики и алгоритмики
(протокол № 14 от 19 марта 2020 года);

Научно-методическим Советом БГУ
(протокол № 4 от 25 марта 2020 года).

Заведующий кафедрой
дискретной математики и алгоритмики



В.М. Котов

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Цели и задачи учебной дисциплины

Учебная дисциплина «Разработка высокопроизводительных приложений на С и С++» дает представление о принципах реализации высокопроизводительных приложений для обработки больших объемов информации с использованием языков программирования С и С++.

Цель учебной дисциплины – формирование навыков написания быстрого и качественного программного кода, который может использовать все доступные аппаратные ресурсы для достижения оптимальной производительности.

Задачи учебной дисциплины:

1. Ознакомление с новыми возможностями современных стандартов.
2. Изучение методов оптимизации скорости выполнения программы с учетом конкретной архитектуры компьютера.
3. Изучение принципов параллелизма для эффективного использования нескольких ядер процессора.

Место учебной дисциплины в системе подготовки специалиста с высшим образованием (магистра).

Учебная дисциплина относится к компоненту учреждения высшего образования и входит **в модуль** «Избранные главы компьютерных наук».

Программа составлена с учетом **межпредметных связей** с учебными дисциплинами. Основой для изучения учебной дисциплины являются следующие учебные дисциплины первой ступени высшего образования: «Программирование», «Архитектура компьютеров», «Операционные системы». Также очень полезными будут знания, полученные при изучении дисциплины второй ступени высшего образования вузовского компонента «Внутреннее устройство ОС семейства UNIX» модуля «Избранные главы компьютерных наук». Знания, полученные в учебной дисциплине, используются при изучении учебной дисциплины вузовского компонента «Технологии проектирования и разработки высоконагруженных веб-систем» модуля «Избранные главы компьютерных наук».

Требования к компетенциям

Освоение учебной дисциплины «Разработка высокопроизводительных приложений на С и С++» должно обеспечить формирование следующих специализированных и углубленных профессиональных компетенций:

специализированные компетенции:

СК-24. Использовать современные научные и технические достижения в области разработки эффективных алгоритмов для решения конкретных.

СК-25. Развивать способность выбирать методологию и технологию проектирования компьютерных систем.

углубленные профессиональные компетенции:

УПК-5. Развивать способность выбирать методологию и технологию проектирования компьютерных систем

В результате изучения дисциплины студент магистратуры должен:

знать:

- основные положения стандартов языка;
- возможности стандартной библиотеки;
- особенности выполнения программного кода на архитектуре x86-64;
- способы организации параллельного выполнения программ;

уметь:

- писать модульный программный код на современном C++;
- выполнять сборку программ, статических и динамических библиотек под ОС Windows и GNU/Linux (как собственных, так и сторонних, которые распространяются в виде исходного кода);
- пользоваться средствами отладки, профилирования и диагностики, выявлять ошибки и узкие места;
- оптимизировать код с точки зрения скорости выполнения;
- проектировать многопоточные приложения;

владеть:

приемами программирования с использованием широкого спектра инструментов языков программирования C и C++.

Структура учебной дисциплины

Дисциплина изучается в 3-м семестре. Всего на изучение учебной дисциплины «Разработка высокопроизводительных приложений на C и C++» отведено:

- для очной формы получения высшего образования – 90 часов, в том числе 40 аудиторных часов, из них: лекции – 20 часов, практические занятия – 20 часов (в том числе – 10 часов дистанционного обучения).

Трудоемкость учебной дисциплины составляет 3 зачетные единицы.

Форма текущей аттестации по учебной дисциплине – зачет.

СОДЕРЖАНИЕ УЧЕБНОГО МАТЕРИАЛА

Раздел I. Избранные сведения о языке C++

Тема 1.1. Возможности современного C++

Стандарты языка. Нововведения синтаксиса.

Списки инициализации и универсальная инициализация. Лямбда-функции и выражения. Семантика перемещения и rvalue-ссылки. Оптимизация возвращаемого значения. Шаблоны с переменным числом аргументов. Обобщенные константные выражения.

Тема 1.2. Исключения в C++

Обработка исключительных ситуаций. Уровни безопасности исключений. Идиома RAII. Умные указатели. Борьба с утечками памяти.

Тема 1.3. Типичные ошибки при программировании

Понятие неопределенного поведения. Неуточняемое поведение.

Трудности создания переносимого кода.

Средства диагностики ошибок и отладки. Работа с отладчиком GDB.

Тема 1.4. Сборка библиотек и программ

Компиляция и компоновка. Статические и динамические библиотеки.

Понятия API и ABI. Идиома Pimpl.

Системы сборки Microsoft Visual Studio, GNU make и CMake.

Раздел II. Ускорение выполнения программы

Тема 2.1. Механизм вызова функций

Регистры процессора x86-64. Соглашения вызова. Организация системного стека. Реализация виртуальных вызовов и обработки исключений. Inline-функции.

Тема 2.2. Оптимизация на уровне машинного кода

Модули исполнения в современном процессоре x86-64. Значения latency и throughput отдельных инструкций.

Приемы оптимизации, реализованные в современных компиляторах. Эффективность конструкций языка.

Оптимизация доступа к памяти. Кэши. Выравнивание.

Тема 2.3. Решение ресурсоемких вычислительных задач

Стандарт IEEE 754. Реализация вычислений с плавающей точкой в процессоре x86-64.

Использование векторных операций. SIMD. Современные наборы инструкций. Ручная и автоматическая векторизация.

Тема 2.4. Диагностика и профилирование кода

Корректное измерение времени выполнения.

Средства профилирования: gperftools, Linux perf, Callgrind (из состава Valgrind).

Раздел III. Параллельное программирование

Тема 3.1. Многопоточность в современном C++

Модель памяти в C++11. Атомарные типы.

Потоки. Блокировки и синхронизация. Асинхронные функции.

Тема 3.2. Приемы конкурентного программирования

Пулы потоков. Организация работы производителей и потребителей.

Механизм блокировки чтения-записи (Readers–Writer Lock). Механизм чтения-модификации-записи (Read-Copy-Update).

Неблокирующие (lock-free) алгоритмы и структуры данных. Проблема АВА.

Тема 3.3. Кооперативная многозадачность

Механика переключения контекста. Сопрограммы (coroutines). Нововведения стандарта C++20.

УЧЕБНО-МЕТОДИЧЕСКАЯ КАРТА УЧЕБНОЙ ДИСЦИПЛИНЫ

Дневная форма получения образования с применением дистанционных образовательных технологий

№ п/п	Название раздела, темы	Количество аудиторных часов				Количество часов УСП	Форма контроля знаний
		Лекции	Семинарские занятия	Практические занятия	Иное		
1	Избранные сведения о языке C++	6		3 3(ДО)			
1.1	Возможности современного C++	2					Устный опрос
1.2	Исключения в C++	2		1 1(ДО)			Устный опрос. Отчеты по домашним практическим заданиям с их устной защитой.
1.3	Типичные ошибки при программировании	1		1 1(ДО)			Устный опрос. Отчеты по домашним практическим заданиям с их устной защитой.
1.4	Сборка библиотек и программ	1		1 1(ДО)			Устный опрос. Отчеты по домашним практическим заданиям с их устной защитой.
2	Ускорение выполнения программы	7		3 3(ДО)			
2.1	Механизм вызова функций	2		1 1(ДО)			Устный опрос. Отчеты по домашним практическим заданиям с их устной защитой.
2.2	Оптимизация на уровне машинного кода	2		1 1(ДО)			Устный опрос. Отчеты по домашним практическим заданиям с их устной защитой.
2.3	Решение ресурсоемких вычислительных задач	2		1 1(ДО)			Устный опрос. Отчеты по домашним практическим заданиям с их устной защитой.
2.4	Диагностика и профилирование кода	1					Контрольная работа 1

3	Параллельное программирование	7		4 4(ДО)			
3.1	Многопоточность в современном C++	2		1 1(ДО)			Устный опрос. Отчеты по домашним практическим заданиям с их устной защитой.
3.2	Приемы конкурентного программирования	3		2 2(ДО)			Устный опрос. Отчеты по домашним практическим заданиям с их устной защитой. Тестирование.
3.3	Кооперативная многозадачность	2		1 1(ДО)			Контрольная работа 2. Отчеты по домашним практическим заданиям с их устной защитой.

ИНФОРМАЦИОННО-МЕТОДИЧЕСКАЯ ЧАСТЬ

Перечень основной литературы

1. Керниган Б., Ритчи Д. Язык программирования С = The C programming language. — 2-е изд. — М.: Вильямс, 2007. — 304 с.
2. Мейерс С. Эффективный и современный С++: 42 рекомендации по использованию С++11 и С++14. — М.: Вильямс, 2016. — 304 с.
3. Agner Fog. Optimizing software in C++: An optimization guide for Windows, Linux and Mac platforms: <http://www.agner.org/optimize/>
4. Herlihy M. The Art of Multiprocessor Programming / Maurice Herlihy, Nir Shavit. — San Francisco: Morgan Kaufmann Publishers Inc., 2012. — 536 p.
5. Sehr V., Andrist B. C++ High Performance: Boost and optimize the performance of your C++ 17 code. — Packt Publishing, 2018.
6. Roth S. Clean C++: Sustainable Software Development Patterns and Best Practices with C++ 17. — Apress, 2017.

Перечень дополнительной литературы

1. W. Richard Stevens, Stephen A. Rago. Advanced Programming in the UNIX Environment. — Addison-Wesley, 2013.
2. Jeffrey Richter. Windows via C/C++. — Microsoft Press, 2011.
3. Michael Kerrisk. The Linux Programming Interface: A Linux and UNIX System Programming Handbook. — No Starch Press, 2010.
4. Eli Bendersky's website: <https://eli.thegreenplace.net/tag/c-c>

Перечень рекомендуемых средств диагностики и методика формирования итоговой оценки

Для диагностики компетенции в рамках учебной дисциплины рекомендуется использовать следующие формы:

1. Устная форма: устный опрос.
2. Письменная форма: контрольные работы, тестирование.
3. Устно-письменная форма: отчеты по домашним практическим заданиям с их устной защитой, оценивание на основе проектного метода.

Формой текущей аттестации по дисциплине «Разработка высокопроизводительных приложений на С и С++» учебным планом предусмотрен зачет.

При формировании итоговой оценки используется рейтинговая оценка знаний студента, дающая возможность проследить и оценить динамику про-

цесса достижения целей обучения. Рейтинговая оценка предусматривает использование весовых коэффициентов для текущего контроля знаний студентов по дисциплине.

Примерные весовые коэффициенты, определяющие вклад текущего контроля знаний и текущей аттестации в рейтинговую оценку (формирование оценки за текущую успеваемость):

- отчеты по домашним практическим заданиям с их устной защитой – 30%;
- контрольные работы – 30 %;
- устный опрос – 20%;
- тестирование – 20%.

Рейтинговая оценка по дисциплине рассчитывается на основе оценки текущей успеваемости и зачетной оценки с учетом их весовых коэффициентов. Вес оценка по текущей успеваемости составляет 40 %, зачетная оценка – 60%.

Примерная тематика практических занятий

Занятие № 1. «Вводный тест».

Занятие № 2. «Принцип работы умных указателей».

Занятие № 3. «Неопределенное поведение и кроссплатформенность».

Занятие № 4. «Методы сборки программ под Linux и Windows».

Занятие № 5. «Профилирование и оптимизация кода».

Занятие № 6. «Особенности стандарта IEEE 754».

Занятие № 7. «Модель future/promise».

Занятие № 8. «Барьеры памяти и когерентность кешей».

Занятие № 9. «Реализация обновления глобального ресурса».

Занятие № 10. «Потоки и сопрограммы».

Описание инновационных подходов и методов к преподаванию учебной дисциплины

При организации образовательного процесса большинства практических занятий используется *практико-ориентированный подход*, который предполагает освоение содержания учебного материала через решение практических задач, а также приобретение навыков эффективного выполнения разных видов профессиональной деятельности.

Кроме этого, при организации образовательного процесса используется комбинация *методов группового обучения, проектного обучения и учебной дискуссии*. Комбинация методов предполагает: ориентацию на генерирование

идей, приобретение навыков для решения исследовательских, творческих и коммуникационных задач, появление нового уровня понимания изучаемой темы, применение знаний (теорий, концепций) при решении проблем, определение способов их решения.

Методические рекомендации по организации самостоятельной работы обучающихся, подготовка к экзамену

Для организации самостоятельной работы студентов магистратуры по учебной дисциплине следует использовать информационно коммуникационные технологии:

Образовательный портал БГУ (<https://edufpmi.bsu.by>);

Образовательный портал InsightRunner (<https://acm.bsu.by>);

AnyTask (<https://anytask.org/school/bsu>).

Рекомендуется разместить в сетевом доступе комплекс учебных и учебно-методических материалов (учебно-программные материалы, учебное издание для теоретического изучения дисциплины, презентации лекций, методические указания к практическим занятиям, электронные версии домашних заданий, материалы текущего контроля и текущей аттестации, позволяющие определить соответствие учебной деятельности обучающихся требованиям образовательных стандартов высшего образования и учебно-программной документации, в том числе вопросы для подготовки к экзамену, задания, вопросы для самоконтроля, список рекомендуемой литературы, информационных ресурсов и др.).

Примерный перечень заданий для домашних работ

Modernize. Дается программа на чистом C. Требуется переписать ее на C++ с использованием всех возможностей из современного стандарта. Сделать вывод, стал ли при этом код проще и понятнее.

UB. В указанных фрагментах кода обнаружить случаи неопределенного поведения и предложить способы исправления этих ошибок.

Portability. В заданных фрагментах кода найти места, которые потенциально вызовут трудности при переносе программы на другую архитектуру.

Linking. Вам предоставляется набор а- и о-файлов (без исходного кода). Нужно скомпоновать программу при помощи команды gcc с нужным набором аргументов, запустить ее на Linux-системе и получить ответ.

Digest. Требуется написать программу для вычисления хеш-функции от содержимого файла (MD5, SHA-1, ...), состоящую из нескольких файлов,

скомпоновать ее с внешней библиотекой двумя способами (статически и динамически), организовать сборку при помощи GNU Make.

Debug. Предоставляется код, содержащий серьезные ошибки, приводящие к аварийному завершению программы. С помощью отладчика GDB требуется эти ошибки обнаружить и исправить.

I'm a Hacker. Эксплуатируя уязвимость, связанную с переполнением буфера, требуется обойти «защиту» и узнать секретный ключ.

Float. На входе задано вещественное число. Требуется распечатать его битовое представление в формате float согласно стандарту IEEE 754.

Speedup. Предоставляется исходный код программы, решающей определенную задачу. С помощью профилирования нужно выявить узкие места с точки зрения производительности и ускорить программу, не меняя сути алгоритма. Задание можно организовать в виде соревнования среди студентов (у кого получится более быстрый код).

Py module. Для конкретной вычислительной задачи требуется написать на языке C модуль расширения для языка Python. Сравнить производительность такого решения с кодом на чистом Python.

Рекомендуемая тематика контрольных работ

Контрольная работа 1. «Разработка быстрых однопоточных программ».

Контрольная работа 2. «Эффективное использование ресурсов многоядерных систем».

ПРОТОКОЛ СОГЛАСОВАНИЯ УЧЕБНОЙ ПРОГРАММЫ УВО

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	Вносить изменения не требуется (протокол № 14 от 19 марта 2020 года)

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)