Министерство транспорта Российской Федерации

Федеральное агентство железнодорожного транспорта

Федеральное государственное бюджетное   
образовательное учреждение высшего образования

«Дальневосточный государственный   
университет путей сообщения»

Кафедра «Высшая математика»

Е.В. Кадура

**Операционные системы**

Методические указания   
по выполнению курсовой работы

Хабаровск

Издательство ДВГУПС

2018

УДК 004.451 (075.8)

ББК 3 972.11я73

К 13

Рецензенты:

Кандидат педагогических наук, доцент кафедры   
«Высшая математика» Дальневосточного государственного   
университета путей сообщения

*Н.А. Шулика*

**Кадура, Е.В.**

К 13 Операционные системы : метод. указания по выполнению курсовой работы / Е.В. Кадура – Хабаровск : Изд-во ДВГУПС, 2018. – 22 с.

Методические указания разработаны в соответствии с рабочей программой дисциплины «Операционные системы».

В указаниях рассмотрены основные требования, предъявляемые к курсовым работам, методика сбора материалов, самостоятельной работы студента, а также даны советы по написанию курсовой работы и определены критерии оценки.

Предназначены студентам 2-го курса всех форм обучения по направлениям подготовки 010302 «Прикладная математика и информатика», 090303 «Прикладная информатика», 090301 «Информатика и вычислительная техника».

УДК 004.451 (075.8)

ББК 3 972.11я73

© ДВГУПС, 2018

Введение

На сегодняшний день одними из самых популярных языков программирования, которые используются для разработки программного обеспечения являются языки программирования С++ и C#. Область их применения включает создание операционных систем, разнообразных прикладных программ, драйверов устройств, приложений для встраиваемых систем, высокопроизводительных серверов и др.

Понимание принципов программирования на языках С++ и C# помогает программисту лучше освоить структурное построение программы, понять основы управления операционными системами и др.

Данные методические указания предназначены для студентов направления 010302 «Прикладная математика и информатика», 090303 «Прикладная информатика», 090301 «Информатика и вычислительная техника», определяет содержание и правила оформления курсовых работ по этой дисциплине.

В результате изучения курса «Операционные системы» студент должен ***знать*** основные понятия, функции, состав и принципы работы операционных систем; архитектуры современных операционных систем; особенности построения и функционирования семейств операционных систем Unix и Windows; принципы управления ресурсами в операционной системе; основные задачи администрирования и способы их выполнения в изучаемых операционных системах; ***уметь*** управлять параметрами загрузки операционной системы, выполнять конфигурирование аппаратных устройств, управлять учетными записями, настраивать параметры рабочей среды пользователя, управлять дисками и файловыми системами, настраивать сетевые параметры, управлять разделением ресурсов локальной сети; ***иметь навыки*** реализации типовых алгоритмов для разработки программы-диспетчера; отладки и тестирования программы на языке программирования С++, или С#.

Курсовая работа является комплексной самостоятельной работой студента, главной целью и содержанием которой является закрепление теоретических знаний и формирование у студентов навыков практической работы, связанной с разработкой программного обеспечения на языке программирования С++, или C#.

Работа может быть оценена:

**Оценка «отлично»** ставится тогда, когда работа выполнена самостоятельно, студент творчески подошел к реализации ПО с глубоким знанием содержания основных вопросов и литературы по теме.

**Оценка «хорошо»** ставится тогда, когда в работе, выполненной на высоком творческом уровне, нет должной самостоятельности и элементов творчества, есть недочеты в оформлении работы.

**Оценка «удовлетворительно»** ставится тогда, когда в работе в целом освещаются основные вопросы темы, но существуют недостатки в логическом изложении положений, содержатся отдельные ошибочные выводы, а также есть небрежность в оформлении работы.

**Оценка «неудовлетворительно»** ставится, если тема в работе не освещена.

Положительная итоговая оценка по дисциплине «Современные технологии программирования», изучение которой предусматривает выполнение курсовой работы, выставляется при условии сдачи курсовой работы на оценку не ниже, чем «удовлетворительно».

Студентам, получившим неудовлетворительную оценку по курсовой работе, по решению руководителя курсовой работы предоставляется право доработки прежней темы или выбора новой темы курсовой работы и определяется новый срок для её выполнения или доработки.

1. Общие положения курсового проектирования

Курсовая работа является важной и обязательной формой процесса изучения профессиональных дисциплин и представляет собой самостоятельное научно-практическое исследование студента в области определения дисциплины.

На основе изучения учебной и научной литературы, а также в соответствии с практическими навыками программирования студент должен самостоятельно разработать ПО для поставленной задачи, проанализировать, систематизировать и изложить материал избранной им темы, высказать свое мнение по поводу разработки ПО на языке программирования С++ , или С#.

Курсовая работа представляет собой результат самостоятельной творческой деятельности студента. Творческая самостоятельность студента проявляется в умении самостоятельно разработать удобный интерфейс для своего ПО, в соответствии с заданием. Грамотно написать программный код для своего интерфейса и дать теоретическое обоснование для кода.

***Цели*** курсового проектирования по дисциплине «Операционные системы»:

* углубление знаний, полученных в результате изучения теорети­ческого курса дисциплины;
* развитие у студентов практических навыков технологии про­граммирования на языках: С++, или С#, и использования полученных теоретических зна­ний на практике;
* овладение навыками системного подхода к решению вопросов, связанных с разработкой и использованием приемов программи­рования при решении задач прикладного характера;
* овладение навыками использования ПЭВМ при разработке и от­ладке программ;
* применение методов отладки, сборки и тестирования программ­ных изделий;
* овладение методами автоматизации приложений.

Основными ***задачами*** курсового проектирования по дисциплине «Операционные системы» являются:

* обобщить знания полученные студентами по дисциплинам «Архитектура компьютеров», «Информатика»;
* подготовить студентов к самостоятельному составлению программы-диспетчера на языках С++, или С#, для более успешного изучения дисциплин, таких как «Системное программирование», «Администрирование локальных сетей», «Компьютерные сети».

Формулировка задания на курсовую работу связана с тематикой КР по дисциплине «Операционные системы». Тематика курсового проектирования отвечает задачам дисциплины «Операционные системы» и наряду с этим увязываться с практическими требованиями для этой дисциплины.

Темы курсовых работ, их содержание рассматривается и утверждаются на заседании кафедры и не могут быть изменены по желанию студента.

Для курсовых работ по дисциплине «Операционные системы» утвердили следующую тему: «СОЗДАНИЕ ПРОГРАММЫ – ДИСПЕТЧЕРА ОПЕРАЦИОННОЙ СИСТЕМЫ ДЛЯ ОС Windows».

Задание на курсовую работу: Разработать программу-диспетчер операционной системы для ОС Windows, который должен обеспечивать (выбрать задание в соответствии с вариантом).

Задания на курсовую работу индивидуальны и не повторяются:

1. Просмотр списка текущих процессов. Возможность завершения процессов и запуск новых процессов.
2. Просмотр потоков принадлежащих выбранному процессу. Возможность изменения приоритетов потоков. Завершение потоков.
3. Отображения списка установленных в системе устройств и информации о них.
4. Управление питанием компьютера – выключение и включение компьютера в указанное время.
5. Просмотр мониторинга потока событий.
6. Просмотр мониторинга изменений файлов выбранного каталога.
7. Просмотр содержимого текстовых файлов.
8. Просмотр журнала событий ОС.
9. Управление работой таймера.
10. Просмотр информации о состоянии работающих на локальном компьютере приложений.
11. Создание и просмотр событий.
12. Отслеживание режима работы ПО.
13. Отслеживание изменения буфера обмена.
14. Управление виртуальной памятью.
15. Моделирование обрабатывающего устройства и очереди к нему.
16. Управление переходом в спящий режим.
17. Просмотр мониторинга изменения файлов выбранного каталога.
18. Просмотр списка автоматически загружаемого программного обеспечения.
19. Обмен сообщениями посредствам динамически подключаемых библиотек (DLL).
20. Демонстрация генерации исключительных ситуаций и методов их обработки.
21. Просмотр содержимого графических файлов.
22. Сжатие и восстановление данных по алгоритму.

При оформлении курсовой работы на листе задание необходимо четко формулировать название темы работы и характеристики, определяющие его объем и содержание. Также в заданиях указываются дата выдачи, сроки представления работы. Задание подписывает руководитель проектирования.

Выбор программных средств производится студентом и зависит от решаемой задачи, подлежит согласованию с преподавателем (руководителем проекта).

При выполнении курсовой работы следует руководствоваться общими указаниями по организации и методике проведения курсового проектирова­ния и методическими материалами по курсовому проектированию про­граммных продуктов.

Курсовая работа должна быть выполнена студентом в сроки, указанные руководителем.

На процедуру защиты курсовой работы отводится до 30 мин., в том числе   
5–7 мин. отводится на доклад студента.

На защите курсовой работы студент должен кратко изложить содержание работы (возможно представление графического и раздаточного материала, фотографий и др.), дать исчерпывающие ответы на замечания руководителя и вопросы группы. Руководитель определяет уровень теоретических знаний и практических умений студента, а также степень соответствия курсовой работы предъявляемым к ней требованиям. Курсовая работа оценивается руководителем по пятибалльной системе: 5 («отлично»), 4 («хорошо»), 3 («удовлетворительно»), 2 («неудовлетворительно»).

2. Основное содержание курсовой работы

Пояснительная записка к курсовой работе по дисциплине «Операционные системы» должна давать достаточно полное представление о принципе решения задачи с обоснованием пра­вильности решения задачи на ЭВМ. Записка иллюстрируется схемами и программными кодами, выполняемыми с соблюдением всех требований ЕСПД. Эти схемы и программные коды входят в общий объем пояснительной записки и нуме­руются.

Курсовая работа должна включать в указанной последовательности следующие разделы:

1. титульный лист;
2. содержание (оглавление) с указанием страниц;
3. пояснительная записка;
4. руководство пользователя;
5. заключение;
6. список литературы;
7. приложения.

Содержание основных разделов следующее.

1. ТИТУЛЬНЫЙ ЛИСТ должен соответствовать установленному об­разцу (прил. 1).
2. СОДЕРЖАНИЕ включает наименование всех разделов курсовой работы (проекта), а также подразделов и пунктов, если они имеют наименова­ние, с указанием номеров страниц, на которых размещаются начала разде­лов, подразделов и пунктов.
3. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА должна содержать всю информацию, необходимую для сопровождения и модификации программного обеспечения: сведения о его структуре и конкретных компонентах, общее описание алгоритмов и их схемы, а также обоснование принятых технических решений.

**Содержание пояснительной записки** должно включать следующие разделы:

1. Введение:

• актуальность разрабатываемого приложения;

• обоснование в выборе платформы программирования (например .NET Framework) и языка программирования.

2. Глава 1. Теоретические сведения: делается обзор литературных источников, который должен содержать краткую оценку изложенных материалов и принципов программирования.

3. Глава 2. Реализация программы-диспетчера на языке программирования C# (или C++):

* назначение разработки;
* описание интерфейса создаваемого приложения;
* описание системы меню;
* описание вспомогательных форм.

4. Глава 3. Руководство пользователя:

* общие сведения о программном продукте;
* описание установки;
* описание запуска;
* инструкции по работе (или описание пользовательского интерфейса);
* сообщения пользователю.

5. Заключение: краткий вывод по результатам выполнения курсовой работы, отметить достоинства выбранного способа.

6. Список литературы.

В зависимости от особенностей программы или программного изделия допускается уточнять содержание разделов, вводить новые разделы или объединять/исключать отдельные из них. При необходимости допускается включать приложения.

ГЛАВА 1. Теоретические сведения.

Теоретический раздел курсовой работы пишут в соответствии с вариантом.

ГЛАВА 2. Реализация программы на языке программирования C# (или C++).

1. В разделе «Назначение разработки» указать, какие задачи будут реализованы и цели реализации. Сделать описание системы меню приложения. Указать типы объектов, которые могут использоваться при загрузке файла созданного приложения.

2. В разделе «Описание интерфейса создаваемого приложения» необходимо провести теоретический анализ реализуемой задачи и выбрать для практической реализации приемлемый вариант. Указать какие средства программирования облегчат решение задачи (например, использование библиотеки .NET, использование классов и др.). Указать способы взаимодействия с пользователем, которые использовались при создании приложения. Привести скриншоты основных окон создаваемого приложения и описание к ним.

3. В разделе «Описание системы меню» привести скелетную структуру меню приложения. Пояснить логику создания такого меню.

4. В разделе «Описание вспомогательных форм» привести скриншоты вспомогательных форм и дать подробное описание этих форм до и после осуществления какого-либо действия.

ГЛАВА 3. Руководство пользователя.

Составление документации для пользователей имеет свои особенности, связанные с тем, что пользователь, как правило, не является профессионалом в области разработки программного обеспечения.

**Рекомендации по написанию подобной программной документации:**

• учитывайте интересы пользователей – руководство должно содержать все инструкции, необходимые пользователю;

• излагайте ясно, используйте короткие предложения;

• избегайте технического жаргона и узкоспециальной терминологии, если все же необходимо использовать некоторые термины, то их следует пояснить;

• будьте точны и рациональны – длинные и запутанные руководства обычно никто не читает, например, лучше привести рисунок формы, чем долго ее описывать.

**Руководство пользователя содержит следующие разделы:**

1. Раздел «Общие сведения о программе» обычно содержит наименование программного продукта, краткое описание его функций, реализованных методов и возможных областей применения.

2. Раздел «Описание установки» обычно содержит подробное описание действий по установке программного продукта и сообщений, которые при этом могут быть получены.

3. В разделе «Описание запуска», как правило, описаны действия по запуску программного продукта и сообщений, которые при этом могут быть получены.

4. Раздел «Инструкции по работе» обычно содержит описание режимов работы, форматов ввода-вывода информации и возможных настроек.

5. Раздел «Сообщения пользователю» должен содержать перечень возможных сообщений, описание их содержания и действий, которые необходимо предпринять по этим сообщениям.

3. Общие требования к оформлению   
курсового проекта (работы)

Общие характеристики составления курсовой работы изложены в требованиях ГОСТов 7.32-2001, 7.1-2003, 7.12-93, 7.82-2001. Страницы курсовой, а также иллюстрации и таблицы обязаны соответствовать формату А4 (ГОСТ 9327).

Пояснительная записка является основным документом, предъявляемым студентом при защите курсовой работы. В ней отражаются этапы работы и результаты, полученные при выполнении курсовой работы.

Пояснительная записка к курсовому проекту по дисциплине «Операционные системы» должна быть оформлена в печатном виде на стандартных листах А4.

Для печати должен использоваться шрифт – Times New Roman, с размером – 14, межстрочным интервалом – 1,0. Отступ первой строки в абзаце должен составлять 1,25 см. Расстояние от края листа до текста (поля) должны составлять: левый – 3,5 см (оставляется немного места для прошивки); верхний, нижний – 2,0 см; правый – 1,0 см.

Наименование разделов должно выполняться заглавными бук­вами, а остального текста (включая наименования подразделов) – про­писными буквами. Между текстом и наименованием разделов (подраз­делов) должен быть отступ в одну строку. Допускается выполнять заго­ловки разделов и подразделов жирным шрифтом. Не допускается при­менение курсивного шрифта.

Наименование разделов и подразделов начинается с номера с точкой после цифры или между цифрами и располагается от границы левого поля (за исключением наименований «Содержание» и «Библиографический список», которые выполняются строго по центру), а текст выполняется по ширине между полями без примене­ния ручного или автопереноса. Каждый раздел начинают с новой стра­ницы, а подразделы пишутся независимо от места окончания предыду­щего подраздела. Заголовок раздела или подраздела записывают с но­вой строки, причем перед нумеруемыми частями пояснительной запис­ки ставят их номер. Нумерация страниц пояснительной записки начина­ется с содержания, которому присваивается – № 1.

Ссылки на источники информации нумеруются арабскими циф­рами и располагаются в квадратных скобках по окончанию текста взя­того с источника (количество ссылок не должно быть менее пяти). Ссылки на рисунки, формулы, таблицы должны состоять из двух цифр, разделенных точкой, где первая цифра – номер раздела, а вторая – но­мер по порядку в пределах раздела (пример –   
Рис. 5.4; Табл. 2.3; (2.5)).

При выполнении рисунка заголовок и номер выполняется снизу. При выполнении таблицы – номер таблицы располагается у правого поля и пишется полностью, а заголовок ниже на строчку и по центру над таблицей. При выполнении формулы – нумеруется только расчет­ная формула с буквенными индексами, причем номер располагается в той же строке, что и формула, но только у правого поля в круглых скобках, а сама формула пишется по центру. В случае большой длины – формула разбивается на две и больше частей, но только после знака.

Листы приложений подписываются полностью заглавными бук­вами (например – ПРИЛОЖЕНИЕ 1), а ссылки выполняются сокра­щенно – прил. 1, прил. 2. Текст пояснительной записки и приложений разделяются листом в центре, которого – надпись «ПРИЛОЖЕНИЯ», выполненная заглавными буквами (допускается жирным шрифтом).

4. Рекомендации по организации курсовой работы

При разработке программы традиционно выделяют следующие эта­пы: формализация постановки (перевод задачи на язык математической символики), выбор либо разработка методов решения, разработка алгорит­ма, кодирование (перевод алгоритма на алгоритмический язык в соответст­вии с его синтаксисом), тестирование, отладка, защита.

При отладке программы необходимо выявить, локализовать и устра­нить ошибки в алгоритме и реализующей его программе. Отдельно разра­ботать набор тестов, которые позволяли бы убедиться в корректности алго­ритма, уверенности в том, что программа выдает результаты, соответст­вующие задаче и исходным данным. Тесты необходимо разрабатывать до, а не во время или после разработки программы, чтобы избежать влияния сте­реотипов алгоритма на тестирование. Необходимо готовить не один тест, а их совокупность – набор тестов, призванный охватить максимум ситуаций. Набор тестов называется полным, если он позволяет активизировать все ветви алгоритма. Задача – разработать полный набор тестов. При разработ­ке набора тестов необходимо учесть следующие группы тестов для провер­ки программы:

* при корректных, нормальных исходных данных самого просто­го вида;
* на границе области определения, в ситуациях, которые могут произойти и на которые нужно корректно реагировать;
* за границей области определения (а возможно и здравого смыс­ла) – ситуации, бессмысленные с точки зрения постановки за­дачи, но которые могут произойти из-за ошибок пользователя.

Требование надежности программирования: принимать данные, если они корректны, и получать для них правильные результаты либо отвергать их как некорректные, по возможности с анализом некорректности. Одним из обязательных пунктов при защите курсовой работы является демонстра­ция работоспособности программы при использовании совокупности разра­ботанных тестов, проверке алгоритма и программы.

Эффективность разработанного алгоритма необходимо оценивать по­треб­ляемыми ресурсами компьютера, а именно быстродействием (по ко­личеству выполняемых операций, с учетом трудоемкости каждой из них) и общим объемом оперативной памяти, выделяемой (запрашиваемой под данные). Эти показатели являются противоречивыми. Повышение быстро­действия может потребовать дополнительных расходов памяти, либо на­оборот. Если можно улучшить один показатель без ущерба для другого, следует этого добиваться. Если этого сделать нельзя – следует отдавать предпочтение экономии памяти в ущерб производительности, так как так­товая частота компьютеров растет опережающими темпами в сравнении с объемом оперативной памяти.

Приведем ряд мер, которые можно рекомендовать для повышения эффективности разрабатываемых программ:

1. Не использовать рабочие массивы того же порядка размерно­сти, что и обрабатываемый или создаваемый, если это возможно. При обра­ботке двумерного массива допустимо выделение одномерного рабочего массива для временного хранения строки или столбца.
2. Выбирать где это возможно, короткие типы данных.
3. Использовать поименованные константы вместо неоднократно­го повторения констант-близнецов.
4. При обращении к процедурам (особенно рекурсивным) пара­метры передавать преимущественно по адресу, а не по значению. Перемен­ные, используемые в процедурах как рабочие, объявлять локальными, а не глобальными.
5. Выбирать алгоритмы, эффективные по числу операций, оценив предварительно порядок зависимости трудоемкости от размерности (лога­рифмическая, линейная, полиномиальная, факториальная и т. д.).
6. Избегать вычислений в циклах выражений, не зависящих от па­раметра цикла.
7. Прекращать вычисления, когда результат достигнут, либо оче­видно, что он не может быть достигнут за приемлемое время.

5. Оформление списка используемой литературы   
для курсовой работы

Список использованных источников и литературы – список содержит перечень литературных источников (монографии, статьи, отчеты, материалы из сети internet и т.п.), которые были использованы при выполнении курсовой работы и на которые сделаны ссылки в тексте работы. Список использованной литературы должен оформляться в соответствии с требованиями ГОСТа 7.1-2003. Не менее 15–20 наименований.

Допускается одно из следующих расположений библиографических описаний источников в списке использованной литературы:

– алфавитное;

– хронологическое (по годам издания);

– тематическое (в списке выделяются самостоятельные рубрики, а внутри каждой рубрики расположение литературы проводится в алфавитном порядке);

– расположение нумерованных записей (литература располагается в порядке первых упоминаний в основном тексте).

Нумерация источников в списке использованной литературы сквозная.

При указании места издания допускаются следующие сокращения: для Моск­вы – М., для Ростова-на-Дону – Ростов н/Д., для Нижнего Новгорода – Н. Новгород, для Санкт-Петербурга – СПб. Названия остальных городов должны указываться полностью.

Список использованной литературы в разделе «Монографии, учебные пособия, научные статьи, справочная литература» составляется в алфавитном порядке фамилий авторов с обязательной нумерацией работ. Например:

1. Скит, Д.С.C# для профессионалов: тонкости программирования / Д.С. Скит. – 3-е изд. – М. : Вильямс, 2014. – 608 с.

2. Таненбаум, Э. Архитектура компьютера / Э. Таненбаум, Т. Остин. – 6-е изд. – СПб. : Питер, 2015. – 816 с.

При использовании Интернет-источников указывается электронный адрес страницы полностью:

1. Российская государственная библиотека [Электронный удаленного доступа ресурс] / Центр информ. технологий РГБ; ред. Т.В. Властенко; Web-мастер Н.В. Козлова. – Электрон, дан. – М. : Рос. гос. б-ка, 2013. – Режим доступа: http//www.rsl.ru, свободный. – Загл. С экрана. – Яз. рус., англ.

6. Порядок выполнения курсовой работы

Курсовая работа должна быть выполнена студентом в сроки, указанные ниже.

1–2-я недели. Получение от руководителя курсового проектирования задания для курсовой работы.

3–5-я недели. Подбор литературы, составление предварительного перечня литературы, необходимой для выполнения курсовой работы.

6–7-я недели. Разработка плана (содержания) курсовой работы и графика работ, согласование их с руководителем работы.

8–10-я недели. Изучение теоретического материала. Работа над «Основной частью».

11-я неделя. Представление руководителю проектирования «Основной части» курсовой работы. Исправление «Основной части» в соответствии с замечаниями руководителя проектирования.

12–13-я недели. Работа над практической частью курсовой работы. Демонстрация разработанного программного обеспечения руководителю работы.

14-я неделя. Работа над введением и заключением курсовой работы, оформление курсовой работы в соответствии с ГОСТом 7.32-2001, 7.1-2003, 7.12-93, 7.82-2001. Сдача оформленной работы руководителю проектирования.

15-я неделя. Защита курсовой проекта работы.

7. Порядок сдачи курсовой работы

С момента выбора темы курсовой работы студент начинает собирать материал: подбирать необходимую для написания работы литературу, нормативные источники, составляет план работы, который согласовывается с руководителем. В ходе подготовки курсовой работы студент обязательно консультируется с руководителем.

Студент обязан сдать подготовленную курсовую работу на кафедру в сроки, указанные во введении. Сдача курсовой работы на кафедру осуществляется в печатном и электронном виде.

Руководитель в течение двух недель осуществляет проверку курсовой работы и вносит свои поправки, коррективы, изменения, замечания и дополнения.

По истечении двух недель, студент забирает с кафедры курсовую работу с отзывом руководителя, содержащим допуск к защите либо с указанием замечаний, составленных в письменной форме, для их исправления.

Студент в течение двух недель исправляет курсовую работу в соответствии с замечаниями руководителя и сдает итоговый и предыдущий варианты курсовой работы на кафедру в бумажном и электронном виде.

Через неделю студент получает на кафедре курсовую работу с отзывом руководителя, в котором указываются вопросы, на которые студенту необходимо ответить во время защиты.

В назначенный кафедрой день студент приносит на защиту курсовую работу с отзывом руководителя.

8. Порядок защиты курсовой работы

Выполненная студентом курсовая работа проверяется в срок до 15 дней руководителем работы, который даёт письменный отзыв. На основании письменного отзыва решается вопрос о допуске курсовой работы к защите. Заключительный этап курсовой работы – это ее защита. В ходе защиты студент должен показать знания, полученные при изучении выбранной темы. Поэтому ему следует быть готовым к ответу на любой вопрос, как по тексту работы, так и по проблеме в целом.

Защита курсовой работы проводится на открытом практическом, или лекционном занятии. Рекомендуемая процедура защиты включает в себя:

* чтение отзыва руководителя курсовой работы;
* доклад студента;
* вопросы слушателей и руководителя;
* ответы студента на вопросы.

На процедуру защиты курсовой работы отводится до 30 мин., в том числе   
5–7 мин. отводится на доклад студента.

На защите курсовой работы студент должен кратко изложить содержание работы (возможно представление графического и раздаточного материала, фотографий и др.), дать исчерпывающие ответы на замечания руководителя и вопросы присутствующих слушателей. Руководитель определяет уровень теоретических знаний и практических умений студента, а также степень соответствия курсовой работы предъявляемым к ней требованиям. Курсовая работа оценивается руководителем (с учётом отзыва, доклада студента, ответов студента на вопросы, качества выполнения работы) по пятибалльной системе: 5 («отлично»), 4 («хорошо»), 3 («удовлетворительно»), 2 («неудовлетворительно»).

Заключение

Курсовое проектирование по дисциплине «Операционные системы» является неотъемлемой частью подготовки бакалавров направления 010302 «Прикладная математика и информатика», 090303 «Прикладная информатика», 090301 «Информатика и вычислительная техника» в высшем образовании.

Курсовое проектирование играет большую роль в развитии навыков самостоятельной творческой работы студентов, так как оно позволяет путем решения конкретных производственных задач приобщать студентов к будущей деятельности, воспитывает их в духе ответственности за выполненную работу, прививает им элементы научно-исследовательской работы, а так же своевременно подготавливать студентов к итоговой государственной аттестации.

Задачей курсового проектирования по дисциплине «Операционные системы» является, во-первых, закрепление знаний, полученных при теоретическом изучении данного курса и приобретение навыков создания приложений на языке программирования C# (или С++).

Курсовое проектирование должно способствовать закреплению, углублению и обобщению знаний, полученных студентами по изучаемым дисциплинам, и применению этих знаний к комплексному решению конкретной информационной задачи.

Одной из целей курсового проектирования является овладение современными информационными технологиями для решения задач в своей профессиональной деятельности.

Работая над курсовым проектом, студенты должны научиться пользоваться справочной литературой, а также научиться применять современные технические средства для разработки программного продукта.

Курсовое проектирование является завершающим этапом в изучении дисциплины «Операционные системы», в ходе, которого осуществляется обучение применению полученных знаний и умений при решении комплексных задач, связанных со сферой профессиональной деятельности будущих специалистов.

Библиографический список

1. Павловская, Т.А. C#. Программирование на языке высокого уровня : учеб. для вузов / Т.А. Павловская. – СПб. : Питер, 2014. – 423 с.
2. Павловская, Т. А. C/C++. Программирование на языке высокого уровня : учеб. для вузов / Т.А. Павловская. – СПб. : Питер, 2011. – 464 с.
3. Таненбаум, Э. Архитектура компьютера / Э. Таненбаум, Т. Остин. – 6-е изд. – СПб. : Питер, 2015. – 816 с.
4. Язык программирования C#. Классика Computers Science / А. Хейлсберг [и др.] – 4-е изд. – СПб. : Питер, 2011. – 784 с.
5. Троелсен, Э. Язык программирования C# 2010 и платформа .NET 4.0   
   / Э. Троелсен; пер. с англ. – 5-е изд. – М. : ООО «И.Д Вильямс», 2011. – 1392 с.
6. Фленов, М.Е. Библия C# / М.Е. Фленов – 2-е изд., перераб. и доп. – СПб. : БВХ-Петербург, 2011. – 560 с.
7. Шилдс, Г. C# 4.0: полное руководство / Г. Шилдс; пер. с англ. – М. : ООО «И.Д. Вильямс», 2011 – 1056 с.
8. Стиллмен, Э. Изучаем C# / Э. Стиллмен, Д. Грин. – 3-е изд. – СПб. :   
   Питер, 2014. – 816 с.
9. Дейтел, П. Как программировать на Visual C# 2012. / П. Дейтел, Х. Дейтел. – 5-е изд. – СПб. : Питер, 2014. – 864 с.
10. Рихтер, Д. CLR via C#. Программирование на платформе Microsoft .NET Framework 4.5 на языке C#. / Рихтер Д. – 4-е изд. – Спб. : Питер, 2015. – 896 с.

Приложение 1

Образец оформления титульного листа

Министерство транспорта Российской Федерации

Федеральное агентство железнодорожного транспорта

Федеральное государственное бюджетное образовательное   
учреждение высшего образования

«Дальневосточный государственный университет путей сообщения»

Кафедра «Высшая математика»

Курсовая работа по дисциплине

**«**Операционные системы**»**

Вариант 1

**СОЗДАНИЕ ПРОГРАММЫ – ДИСПЕТЧЕРА   
ОПЕРАЦИОННОЙ СИСТЕМЫ ДЛЯ ОС Windows**

**Выполнил:**

Студент группы 922 (2 курс, 3 семестр)

Направления 01.03.02 «Прикладная математика и информатика»

***Васечкин Василий Васильевич***

**Проверил:**

Доцент кафедры «Высшая математика», ДВГУПС

***Кадура Елена Вячеславовна***

**Рейтинговый балл за курсовУЮ РАБОТУ** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Оценка за курсовУЮ РАБОТУ**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Хабаровск

201\_

Приложение 2

Образец описания кода

Во время выполнения курсовой работы я воспользовался следующими библиотеками:

* #include <w32api.h> – для работы с WinApi;
* #include <windows.h> – для работы с системой;
* #include <QDialog> – для создания GUI.
* #include <QFileDialog> – для создания кнопки “обзор папок”

И использовал в них следующие функции и структуры:

BOOL CreateProcess(

LPCTSTR lpApplicationName, // имя исполняемого модуля

LPTSTR lpCoramandLine, // командная строка

LPSECURITY\_ATTRIBUTES lpProcessAttributes, // защита процесса

LPSECURITY\_ATTRIBUTES lpThreadAttributes, // защита потока

BOOL blnheritHandles, // признак наследования дескриптора

DWORD dwCreationFlags, // флаги создания процесса

LPVOID IpEnvironment, // блок новой среды окружения

LPCTSTR lpCurrentDirectory, // текущий каталог

LPSTARTUPINFO IpStartUpInfo, // вид главного окна

LPPROCESS\_INFORMATION lpProcessInformation); // информация о процессе

VOID ZeroMemory (

PVOID Destination, // адрес блока памяти

SIZE\_T Length); // длина блока памяти

STARTUPINFO cif; // информация о системе

HANDLE hproc; // дескриптор процесса

* 1. Исходный код
     1. launchapp.h

#ifndef LAUNCHAPP\_H

#define LAUNCHAPP\_H

**Продолжение прил. 2**

#include <w32api.h>

#include <windows.h>

#include <iostream>

#include <cstring>

#include <QFileDialog>

#include <QWidget>

using namespace std;

namespace Ui {

class LaunchApp;

}

class LaunchApp : public QWidget

{

Q\_OBJECT

public:

explicit LaunchApp(QWidget \*parent = 0);

~*LaunchApp*();

private slots:

void on\_pushButton\_clicked(); //событие для кнопки «Создать процесс»

void on\_pushButton\_2\_clicked(); //событие для кнопки «Обзор»

void on\_pushButton\_3\_clicked(); //событие для кнопки «Уничтожить процесс»

private:

Ui::LaunchApp \*ui;

HANDLE hproc; // дескриптор

uint ext;

string \*sinfo;

};

#endif // LAUNCHAPP\_H

* + 1. launchapp.cpp

#include "launchapp.h"

#include "ui\_launchapp.h"

using namespace std;

LaunchApp::LaunchApp(QWidget \*parent) :

QWidget(parent),

ui(new Ui::LaunchApp)

**Окончание прил. 2**

{

ui->setupUi(this); //создание интерфейса

}

LaunchApp::~*LaunchApp*()

{

delete ui;

}

void LaunchApp::on\_pushButton\_clicked()

{

QString str = ui->lineEdit->text();

wstring s = str.toStdWString();

LPCWSTR sw = s.c\_str();//приведение типа из QString в тип Win32API LPCWSTR

STARTUPINFO cif; //структура для хранения системной информации

ZeroMemory (&cif, sizeof(STARTUPINFO)); //обнуляем поля

PROCESS\_INFORMATION pif; //структура для хранения информации о процессе

if(!CreateProcess(sw,NULL,NULL,NULL,FALSE,NULL,NULL,NULL,&cif,&pif))//если не получилось создать процесс – вывести в журнал ошибку

{

ui->listWidget->addItem("I can't create it");

}

Else //иначе запись информации о процессе в переменную и вывод соответствующего сообщения

{

hproc = pif.hProcess;

ui->listWidget->addItem(ui->lineEdit->text()+" !CREATED! ");

}

}

void LaunchApp::on\_pushButton\_2\_clicked()

{

QString str = QFileDialog::getOpenFileName(0,"Select a Directory",ui->lineEdit->text());

if(!str.isEmpty()) //если путь содержит символы – заносим их в linedit

{

ui->lineEdit->setText(str);

}

}

void LaunchApp::on\_pushButton\_3\_clicked()

{ if(TerminateProcess(hproc,1)) //если удалось уничтожить процесс – вывести соответствующее сообщение ui->listWidget->addItem("Process finished");}

Приложение 3

Образец описания интерфейса

Интерфейс приложения-диспетчера

Путь  
к приложению

Уничтожить процесс

Создать процесс

Обзор

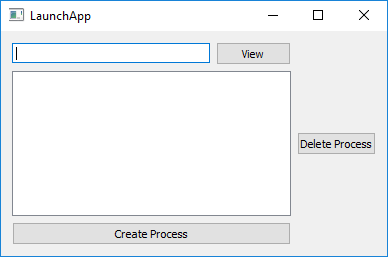


Рис. 1. Интерфейс приложения

**оглавление**

Введение 3

1. Общие положения курсового проектирования 5

2. Основное содержание курсовой работы 7

3. Общие требования к оформлению   
курсового проекта (работы) 9

4. Рекомендации по организации курсовой работы 11

5. Оформление списка используемой литературы   
для курсовой работы 12

6. Порядок выполнения курсовой работы 13

7. Порядок сдачи курсовой работы 13

8. Порядок защиты курсовой работы 14

Заключение 15

Библиографический список 16

Приложение 1. Образец оформления титульного листа 17

Приложение 2. Образец описания кода 18

Приложение 3. Образец описания интерфейса 21

Учебное издание

**Кадура**Елена Вячеславовна

**Операционные системы**

Методические рекомендации   
по выполнению курсовой работы

***Отпечатано методом прямого репродуцирования***

Технический редактор *Н.В. Ларионова*

————————————————————————————

План 2018 г. Поз. 9.7. Подписано в печать 24.01.2018.

Гарнитура Times New Roman. Печать RISO.

Усл. печ. л. 1,3. Уч.-изд. л. 1,5.

Зак. 60. Тираж 35 экз. Цена 75 р.

————————————————————————————

Издательство ДВГУПС

680021, г. Хабаровск, ул. Серышева, 47.

Кафедра «Высшая математика»

Е.В. Кадура

Операционные   
системы

Методические указания   
по выполнению курсовой работы

Хабаровск

2018