

Федеральное агентство связи
Федеральное государственное бюджетное образовательное учреждение
высшего образования
Поволжский государственный университет телекоммуникаций и информатики

**КАФЕДРА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ И
УПРАВЛЕНИЯ В ТЕХНИЧЕСКИХ СИСТЕМАХ**

Малахов С.В., Иванов М.С.

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ
ПО ВЫПОЛНЕНИЮ КУРСОВЫХ РАБОТ**

ПО УЧЕБНОЙ ДИСЦИПЛИНЕ
«ОПЕРАЦИОННЫЕ СИСТЕМЫ И ОБОЛОЧКИ»

по направлениям подготовки: 09.03.01 – Информатика и
вычислительная техника, 09.03.04 – Программная инженерия,
02.03.03 - «Математическое обеспечение и администрирование
информационных систем»

Самара 2020

Малахов С.В., Иванов М.С. Методические указания по выполнению курсовых работ по учебной дисциплине «Операционные системы и оболочки». – Самара: ПГУТИ, 2020. – 29 с., ил.

Методические указания содержат общие рекомендации для студентов по выполнению и оформлению курсовой работы по дисциплине «Операционные системы и оболочки» и требования для оформления работы. Цель курсовой работы является закрепление, углубление и контроль знаний, полученных в процессе изучения дисциплины «Операционные системы и оболочки». Задачи курсовой работы состоят в том, чтобы спроектировать, разработать и протестировать файловый менеджер, отражая, таким образом, полученные в пройденном курсе знания.

Методические указания по выполнению курсовых работ по учебной дисциплине «Операционные системы и оболочки» подготовлены на кафедре «ПОУТС», предназначены для студентов всех форм обучения направлений подготовки:

- 09.03.01 – «Информатика и вычислительная техника»,
- 09.03.04 – «Программная инженерия»,
- 02.03.03 - «Математическое обеспечение и администрирование информационных систем».

ВВЕДЕНИЕ

Начало XXI века ознаменовалось тотальным распространением электронных устройств и цифровой информации. Происходит крайне быстрое развитие как программной, так и аппаратной составляющей электронных устройств. Операционная система (ОС) является неотъемлемой программной составляющей современного компьютера и ряда других электронных устройств (смартфоны, планшетные компьютеры, ноутбуки, оборудование для томографии), которая предоставляет средства взаимодействия других компьютерных программ с аппаратным обеспечением и друг с другом, а также обеспечивает удобство пользователя при работе с компьютером. Поэтому, глубокое и всестороннее изучение современных операционных систем, способов управления ими и администрирования является важным звеном при подготовке специалистов в области информатики и вычислительной техники.

XXI век — век цифровой информации, и крупномасштабная обработка и хранение этой информации в настоящее время осуществляется сотнями тысяч и миллионами серверных компьютеров и суперкомпьютеров по всему миру.

Дисциплина «Операционные системы и оболочки» изучает непосредственно сами операционные системы и их компоненты, которые связаны между собой и работают как единый комплекс.

Данная дисциплина призвана сформировать у студента систематизированные представления о основополагающих принципах работы операционных систем, о структурах и механизмах, лежащих внутри системы, а также получить навыки работы с операционными системами.

1 ЦЕЛИ И ЗАДАЧИ КУРСОВОЙ РАБОТЫ

Целями курсовой работы (КР) по дисциплине «Операционные системы и оболочки» являются закрепление и углубление теоретических знаний в

области современных операционных систем, активное применение знаний, полученных в лекционном курсе и на лабораторных занятиях, а также овладение современными информационными технологиями для решения задач в своей профессиональной деятельности.

Задачами курсовой работы являются приобретение навыков проектирования и разработки компонентов операционных систем, организации структуры комплекса программных средств и распределения функциональных задач между аппаратными и программными средствами реализации.

Выполнение курсовой работы должно позволит получить опыт работы с операционными системами Windows и Linux. Исследование функций операционных систем Windows и Linux, сравнительный анализ одних и тех же функций этих систем, изучение специального программного обеспечения, включает в себя выполнение курсовой работы.

Работая над курсовой работой, студенты научатся пользоваться справочной литературой, материалами ГОСТов, а также применять современные технические средства для разработки системных программных продуктов.

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

2 СТРУКТУРА И СОДЕРЖАНИЕ КУРСОВОЙ РАБОТЫ

Выполнение курсовой работы следует начать с изучения заданий. Чтобы найти оптимальное решение для своего варианта, необходимо по специальной литературе, сборникам алгоритмов и программ подобрать литературу, в которой описано решение подобных задач. Для получения отличной оценки за работу требуется составление оригинальной программы.

В курсовой работе необходимо отразить современный, индивидуальный и творческий подход к изучаемой проблеме. При этом студент должен использовать теоретические знания и практические навыки, получаемые в процессе обучения. Курсовая работа выполняется самостоятельно. Возникающие вопросы обсуждаются с руководителем на консультациях, установленных графиком.

Недопустимо механическое (без оформления в качестве цитаты) переписывание материала из первоисточников. Выявив нарушение этого требования, преподаватель возвращает представленный вариант работы для повторного написания.

Курсовая работа включает следующие элементы:

- титульный лист (Приложение А);
- рецензия (Приложение Б);
- содержание (Приложение В);
- задание;
- введение;
- основная часть;
- заключение;
- список использованных источников;
- приложения.

Основная часть, в свою очередь, структурно и тематически делится на:

1 Проектирование приложения

1.1

1.2

2 Разработка приложения

2.1

2.2

3 Пример работы разработанного приложения

3.1

3.2

Объем курсовой работы составляет 40-50 страниц печатного текста.

Рекомендуемый объем структурных частей курсовой работы:

- введение – 1-2 стр.;
- основная часть – 30-40 стр.;
- заключение – 1-2 стр.;
- список использованных источников – 1 стр.

Введение

Во введении представляется общая характеристика курсовой работы. Дается обоснование актуальности выбранной темы – современные научные исследования, проблемы и возможные их решения. Следует четко определить функции программного обеспечения, его достоинства по сравнению с существующими, а также пользователя с точки зрения его квалификации. Формулируется объект и предмет исследования, цель исследования и задачи, решаемые в курсовой работе. Раскрывается структура работы и дается сжатое изложение ее основных положений.

Актуальность выбранной темы – дается обоснование актуальности темы курсовой работы, ее значение для современных операционных систем, программного обеспечения и информационных технологий в целом, и раскрываются проблемные вопросы.

Формулировка цели курсовой работы – любая цель, как правило, начинается с глаголов: выяснить, выявить, сформировать, обосновать, проверить, определить, создать, построить и т.п.

Определение задач курсовой работы. Задачи – это, как правило, конкретизированные или более частные цели. Цель разворачивается в комплексе взаимосвязанных задач.

Объект исследования – это конкретная сфера применения операционных систем и программного обеспечения.

Предмет исследования – это конкретная характеристика, направление в рамках объекта исследования. Берется из задания к курсовой работе.

Необходимо отметить важное правило – введение, как и заключение, рекомендуется писать после полного завершения основной части. До того, как будет создана основная часть работы, сложно написать хорошее введение, так как автор еще не вполне овладел материалами по теме.

Основная часть

Включает последовательное выполнение варианта задания. В проектирование относится 1-3 задания, в разработку с 4-7 задания.

Раздел «Пример работы разработанного приложения» содержит процесс установки и запуска программного обеспечения, описание всех функциональных возможностей со скриншотами.

Одной из основных задач курсовой работы по курсу «Операционные системы и оболочки» является получение практических навыков по составлению эффективных программ. Программа, разработанная студентом, должна быть описана в соответствии с принятыми стандартами для описания программ.

В заключении в лаконичной форме подводятся итоги проделанной работы и делаются основные выводы. Также следует указать, чему студент научился, выполняя задание.

Список использованных источников

В список литературы включают источники, изученные при написании работы. Список включает библиографические описания всех использованных, цитированных или упоминаемых в работе документов, а также прочитанную литературу по теме, которая оказала существенное влияние на содержание работы. Источники указываются в порядке упоминания их в тексте отчета.

Приложения

В приложения при необходимости следует включить вспомогательный или дополнительный материал, необходимый для более полного раскрытия темы, который загромождает текст основной части работы, но необходим для полноты её восприятия и оценки практической значимости.

Программный код в приложения вставляется с комментариями. Для уменьшения количества страниц разрешается уменьшить размер шрифта программного, также можно разбить код на 2 колонки.

Приложения должны быть пронумерованы и иметь названия. Приложения оформляются как продолжение КР на последующих страницах и располагаются в порядке появления ссылок в тексте. Каждое приложение следует начинать с новой страницы с указанием наверху по середине страницы слова «Приложение». Если в работе более одного приложения, их нумеруют последовательно, например, Приложение А, Приложение Б и т.д.

3 ТРЕБОВАНИЯ К ОФОРМЛЕНИЮ КУРСОВОЙ РАБОТЫ

В пояснительной записке КР должны быть выдержаны единые обозначения и единые размерности для используемых параметров. Допускаются общепринятые сокращения слов, терминов, обозначений. Курсовой проект оформляется по следующим требованиям:

1. Общие требования к оформлению.

- 1.1. Титульный лист является первой страницей КР и заполняется по утвержденной форме. Обязательно расставляется подпись студента, дата, номер зачетной книжки и тема КР.

- 1.2. Лист рецензии остается пустым.

- 1.3. Содержание должно включать наименования заголовков КР с указанием номеров страниц, на которых размещается начало материала соответствующих частей.

- 1.4. В листе задания указывается номер зачетной книжки и задание соответствующее варианту.

- 1.5. КР должен быть оформлен на стандартных листах бумаги А4 (210 297 мм) с одной стороны. Текст работы печатается через полтора интервала шрифтом Times New Roman кеглем в 14 пунктов. Размеры полей: левое – 30 мм, правое – 15 мм, верхнее – 20 мм, нижнее

–20 мм., абзацный отступ - 1,25 см. Текст работы должен быть выровнен по ширине.

1.6. Текст основной части КР делится на разделы, подразделы. Объем раздела (подраздела) не должен быть меньше половины страницы.

1.7. Расстояние между заголовками разделов КР и последующим текстом, а также между предыдущим текстом и заголовками подразделов должно составлять одну строку. Расстояние между заголовками подразделов и последующим текстом устанавливается в полтора интервала. Если между заголовками отсутствует текст, то расстояние между ними устанавливается также в полтора интервала.

1.8. Наименования заголовков разделов КР печатаются строчными буквами (кроме первой прописной) шрифтом Times New Roman кеглем в 16 пунктов. Заголовки подразделов печатаются строчными буквами (кроме первой прописной), шрифтом Times New Roman кеглем в 14 пунктов. Точку в конце заголовка не ставят. Если заголовок состоит из двух или более предложений, их разделяют точкой. Для заголовков разделов и подразделов использоваться полужирный шрифт. Все заголовки располагаются с выравниванием по центру. (пример смотри выше «Пояснительная записка КР должна содержать следующие разделы»).

1.9. Если в нумерованном списке после номера стоит точка, то текст начинается с прописной буквы и заканчивается точкой. В остальных случаях ставится точка с запятой и текст начинается со строчной буквы, а в конце такого списка ставится точка.

2. Нумерация страниц и разделов.

2.1. Нумерация страниц, разделов, подразделов, рисунков, таблиц, формул и приложений дается арабскими цифрами без знака №.

2.2. Первой страницей КР является титульный лист. При нумерации учитываются все страницы, начиная с титульного листа. Номера страниц на содержании, титульном листе и рецензии не проставляются. На последующих листах номер страницы проставляется внизу страницы справа без точки.

2.3. Номер раздела ставится перед его заголовком, точка после номера не ставится. Слово «раздел» не используется. Подразделы нумеруются в пределах каждого раздела. Номер подраздела состоит из номера раздела и порядкового номера подраздела, разделенных точкой. В конце номера подраздела точка не ставится. Далее через пробел идет заголовок подраздела.

2.4. При наличии пунктов они нумеруются в пределах каждого подраздела. Номер пункта состоит из номера раздела и порядковых номеров подраздела, пункта, разделенных точками.

3. Иллюстрации.

3.1. Иллюстрации (фотографии, рисунки, чертежи, схемы, графики, карты) располагаются в КР непосредственно на странице с текстом после абзаца, в котором они упоминаются впервые, или отдельно на следующей странице. Располагаются по центру без абзаца.

3.2. Иллюстрации обозначаются словом «Рис.» и нумеруются последовательно в пределах раздела. Номер иллюстрации должен состоять из номера раздела и порядкового номера иллюстрации, разделенных точкой. Например, «Рис. 1.2 - ...» (второй рисунок первого раздела). Номер иллюстрации, её название и поясняющие подписи размещаются последовательно под иллюстрацией.

3.3. Иллюстрации должны иметь наименование, которое располагается после номера рисунка и тире. После наименования рисунка точка не ставится.

4. Таблицы.

4.1. Каждая таблица должна иметь заголовок, который располагается над таблицей и печатается в середине строки. Надпись «Таблица» с указанием её номера размещается в правом верхнем углу над заголовком таблицы. Заголовок и слово «Таблица» пишется без подчеркивания с прописной буквы. Точка в конце заголовка не ставится. Заголовки в графах таблицы начинаются с прописных букв.

4.2. Таблицы нумеруются последовательно (за исключением таблиц, приведенных в приложении) в пределах раздела. Номер таблицы должен состоять из номера раздела и порядкового номера таблицы, разделенных точкой, например, «Таблица 1.2» (вторая таблица первого раздела). Если в КР одна таблица, её не нумеруют и слово «Таблица» не пишут.

4.3. Таблица размещается после первого упоминания о ней в тексте таким образом, чтобы ее можно было читать без поворота работы или с поворотом по часовой стрелке.

4.4. Таблицу с большим количеством строк допускается переносить на другой лист. При переносе части таблицы на другой лист (страницу) слово «Таблица» и номер её указываются один раз справа над первой частью таблицы, над другими частями пишут слово «Продолжение табл.». Если в работе несколько таблиц, то после слова «Продолжение табл.» указывается номер таблицы, например, «Продолжение табл. 1.2».

4.5. При наличии таблиц большого формата с большим количеством ячеек допускается применять шрифт и межстрочный интервал меньшего размера.

5. Ссылки.

5.1. Ссылки на иллюстрации КР указываются порядковым номером иллюстрации, например, «На рис. 1.2 ...» или «(рис. 1.2)».

5.2. Ссылки на формулы КР указывают порядковым номером формулы в скобках, например, «... в формуле (2.1)».

5.3. На все таблицы КР должны быть ссылки в тексте, при этом слово «Таблица» в тексте пишется полностью, если она не имеет номера, и сокращенно – если имеет номер, например, «... в табл. 1.2» или «(табл. 1.2)». В случае повторных ссылок на таблицы и иллюстрации в тексте пишется: «см. табл. 1.3».

6. Правила оформления программного кода

6.1. Гарнитура Courier New, кегль (размер шрифта) – 11; междустрочный интервал – одинарный; выравнивание – по левому краю; цвет шрифта – черный.

6.2. При оформлении программного кода следует использовать структурный отступ в два или четыре пробела. Другие размеры отступа использовать не рекомендуется.

6.3. Разделители и пробелы. Перед знаками препинания (запятые, точки с запятой и двоеточия) пробелы не ставятся, а после них – ставятся. Пробелы с внутренних сторон скобок ставить не рекомендуется.

Пример:

```
MyProcedure(a, b[idx + 5], c);
```

При обращении к процедурам и функциям пробел между именем вызываемой подпрограммы и открывающей скобкой не ставят, так же как и пробел между именем массива и открывающей квадратной скобкой операции индексирования.

В заголовке сложного оператора между его названием (ключевым словом if, for, switch и др.) и открывающей скобкой, обрамляющей условное выражение, рекомендуется ставить пробел.

Пример:

```
if (!cond(i)) {
```

Пробелы перед скобками после имен функций ставить не рекомендуется.

Пример:

```

for (i=0; f(i) < g(); i++) {
    if (!cond(i)) {
        proceed(i);
        break;
    }
}

```

7. Пример оформления списка использованных источников.

7.1. Гудвин, Г.К. Проектирование программного обеспечения [Текст] / Г.К. Гудвин, С.Ф. Гребе, М.Э. Сальдаго. – пер. с англ. – М.: БИНОМ, Лаборатория знаний, 2004. – 911 с.

7.2. Брюханов, В.Н. Теория управления программным обеспечением [Текст] : Учеб. для машиностроит. спец. вузов / В.Н. Брюханов, М.Г. Косов, С.П. Протопопов и др.; под ред. Ю.М. Соломенцева – 3-е изд., стер. – М.: Высш. шк., 2000. – 268 с.

7.3. Руководство по Fibre Channel [Электронный ресурс] // Мир компьютерной автоматизации. -2017. -Режим доступа: <http://www.mka.ru/?p=40030>.

4 ТРЕБОВАНИЯ К СОДЕРЖАНИЮ И СТИЛЮ НАПИСАНИЯ КУРСОВОЙ РАБОТЫ

Работа должна быть написана хорошим литературным языком, без грамматических и синтаксических ошибок от 3 лица. Сокращения допускаются только общепринятые. Следует избегать канцелярского стиля, но и не следует подражать стилю учебника или публицистики. Изложение должно быть ясным и понятным, четко выражать мысль автора. Рекомендуется обрести собственный стиль.

Основными целями и задачами написания курсовой работы является не только расширение и углубление знаний студента, но и формирование умения анализировать теоретический и практический материал логично, последовательно.

Курсовая работа должна представлять собой целостную, однородную и завершенную студенческую научную работу, в которой должны быть четко сформулированы проблема и их решение, обоснована их актуальность, изложены степень изученности проблемы и состояние ее исследования.

Курсовая работа должна быть написана хорошим научным языком, т.е. с соблюдением общих норм литературного языка, правил грамматики и с учетом особенностей научной речи – точности и однозначности, терминологии и стиля.

В современной научной литературе личная манера изложения уступила место безличной. Не употребляются личные местоимения «я» и «мы».

Например, вместо фразы «я предполагаю...» можно сказать «предполагается, что...», вместо фразы «я ...» можно сказать «автором...» и т.д.

5 ПОРЯДОК ПОДГОТОВКИ И ЗАЩИТЫ КУРСОВОЙ РАБОТЫ

В установленный учебным графиком срок каждому студенту выдается для ознакомления с заданием на выполнение методическое руководство по выполнению курсовой работы.

Для этого перед непосредственной проверкой руководителем КР на предмет выполнения задания КР и соответствия требованиям, предъявляемым к КР, каждый студент передает преподавателю архив с электронной версией КР, который включает: отчет (файл в формате doc, название файла: дисциплина_группа_год выполнения КР_Фамилия_Имя_Отчество) и файлы проекта. Если проверка пройдена, то далее студент сдает руководителю КР на проверку бумажную версию КР.

Каждая проверенная КР содержит в рецензии результаты проверки преподавателем КР. Если рецензия содержит допуск к защите КР, то далее студент готовится защищать полученные результаты и теоретические знания по предметной области КР. Иначе доработать КР по представленным

замечаниям и сдать повторно на проверку до тех пор, пока не будет получен допуск.

Результаты защиты курсовой работы оцениваются по пятибалльной системе оценки («отлично», «хорошо», «удовлетворительно», «неудовлетворительно»). Во время защиты курсовой работы студенту задаются вопросы преподавателем.

Итоговая оценка за защиту курсовой работы определяется:

«отлично» - если разработанная программа полностью функционирует и включает все задания КР, составлен отчет согласно требованиям к оформлению и дан ответ 5 вопросов;

«хорошо» - если разработанная программа оценена не ниже «хорошо», составлен отчет согласно требованиям к оформлению и дан ответ на 4 вопроса;

«удовлетворительно» - если разработанная программа оценена на «удовлетворительно», в оформлении отчета имеются недочеты и дан ответ на 3 вопроса;

«неудовлетворительно» - если не выполнены условия получения положительной оценки.

6 ЗАДАНИЕ И ВЫБОР ВАРИАНТА КУРСОВЫХ РАБОТ

Задание курсовой работы состоит из следующих основных пунктов:

1. Проектирование программного обеспечения.
2. Разработка графического интерфейса.
3. Разработка программного обеспечения.
4. Обеспечение минимального функционала.
5. Добавление основного функционала.
6. Выбор стратегии тестирования и разработка тестов.
7. Формирование отчета по курсовой работе.

Варианты заданий:

Программное обеспечение «Файловый менеджер» разрабатывается для ОС Windows - четные номера зачетных книжек и Linux - нечетные.

Способ межпроцессного взаимодействия «Файлового менеджера» и всплывающего окна: Отображение файлов – для последней цифры [1 и 2] зачетной книжки, Сокеты — [3 и 4], Каналы — [5 и 6], Семафор — [7 и 8] и Разделяемая память — [9 и 0].

Отслеживание всех процессов ОС для 1-5 номеров зачетных книжек и процессы внутри «Файлового менеджера» 6-0 номера.

Основной функционал для «Файлового менеджера» распределяет преподаватель каждому отдельно:

- 1) имя компьютера;
- 2) имя пользователя;
- 3) версию операционной системы;
- 4) имя процесса;
- 5) использованное время ЦП;
- 6) процент используемой физической памяти;
- 7) процент используемой виртуальной памяти;
- 8) PID процесса;
- 9) приоритет процесса;
- 10) пользователь процесса;
- 11) текущее местное время;
- 12) продолжительность текущего сеанса работы;
- 13) загрузка каждого ядра ЦП в %;
- 14) общая загрузка процессора в %;
- 15) размер файла подкачки в байтах;
- 16) количество свободных байтов файла подкачки;
- 17) ширину и высоту рамки окна;
- 18) ширину и высоту экрана;
- 19) значение загрузки процессора данным процессом;
- 20) количество модулей, используемых процессом;

- 21) размер рабочего множества страниц;
- 22) завершение прикладного процесса, выбранного из списка;
- 23) счетчик количества созданных описателей (дескрипторов);
- 24) маска привязки процесса к процессорам;
- 25) количество страничных ошибок;
- 26) изменение приоритета выбранного процесса;
- 27) время старта процесса (час: мин: сек);
- 28) количество потоков процесса;
- 29) значение размера используемой оперативной памяти;
- 30) пиковое значение размера используемой виртуальной памяти;
- 31) обнаружение фактов создания файлов;
- 32) обнаружение фактов создания каталогов;
- 33) обнаружение фактов переименования файлов;
- 34) обнаружение фактов переименования каталогов;
- 35) для процесса, выбранного с помощью ввода его PID, вывести список его дочерних процессов (имя процесса, PID, время работы в системе);
- 36) вывод в текстовый файл списка процессов, завершивших выполнение в период работы монитора;
- 37) для прикладного процесса, выбранного в списке выполняющихся процессов, вывести список его потоков с указанием свойств каждого потока.

Пояснение

Задание 1. Проектирование «Файлового менеджера». Срок выполнения 5 неделя.

Включает в себя: выбор средств разработки и среды разработки, сделать сравнительную таблицу по ним. Подключаемые внешние библиотеки, их описание.

Задание 2. Разработка графического интерфейса. Срок выполнения 6 неделя.

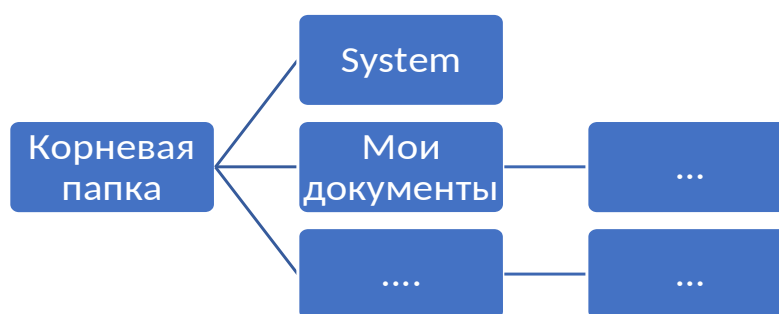
Сделать эскиз графического интерфейса. Дать описание каждой форме и объектам ПО. Для создания интерфейса допускается только специализированное ПО или MS Visio.

Задание 3. Разработка ПО. Срок выполнения 7 неделя.

Разрабатывается приложение «Файловый менеджер» для одной из двух операционных систем, Windows или Linux.

Создается корневая папка как среда работы «Файлового менеджера» (в менеджере отображаются только вложенные папки). Создать папку «Корзина» аналогичная ОС. Обеспечить удаление в корзину и безвозвратно. Создается папка «System», содержит все необходимые файлы (исполняемые) для работы файлового менеджера (ее удалить, переименовать или переместить нельзя). Так же создаются рабочие папки (произвольные). Защитить папку «System» от удаления/переименования. Сделать основное и контекстное меню (аналогичные вашей операционной системы). В основное меню добавить пункт «О программе» - Операционная система, Язык программирования, ФИО и группа разработчика.

Скриншоты из среды разработки обязательны, так же сделать к ним пояснения.



Задание 4. Обеспечение минимального функционала. Срок выполнения 9 неделя.

Обеспечить взаимодействие со съемными носителями, подключаемые к ОС.

Позволить через приложение запуск встроенных системных утилит ОС.

Утилиты - программы, служащие для выполнения вспомогательных операций обработки данных или обслуживания компьютеров (диагностики,

тестирования аппаратных и программных средств, оптимизации использования дискового пространства, восстановления разрушенной на магнитном диске информации и т.п.). В Windows это Управление компьютером, командная строка, Монитор ресурсов и т.д. В Linux можно выделить терминал, Быстрая справка (или поиск чего-либо), калькулятор и прочие.

Сохранение протокола запущенных в период работы «Файлового менеджера» процессов в текстовом файле (имя процесса и время старта). Имя файла задается пользователем.

Скриншоты из среды разработки обязательны, так же сделать к ним пояснения. Сделать подробное описание всех обработчиков событий.

Задание 5. Добавление основного функционала. Срок выполнения 12 неделя.

С помощью приложения обеспечить возможность создания нового процесса (всплывающее окно), которое будет в дальнейшем отображать информацию основного функционала (или части его). Главное окно получает необходимую информацию и передает ее, по средствам межпроцессного взаимодействия всплывающему окну. Всплывающее окно зависит от главного окна.

Добавить новый функционал к разработанному Файловому менеджеру согласно варианта. Вывод полученных данных обеспечить с возможностью обновления в реальном времени (если данные не статические). Возможность сохранения полученных данных в файл (лог-файл).

Лог-файл – это текстовый файл, в котором хранится информация о посещениях и параметрах посещений Файлового менеджера, которые возникали на нем.

Скриншоты из среды разработки обязательны, так же сделать к ним пояснения. Сделать подробное описание всех обработчиков событий. Составить функциональную блок-схему и UML-диаграмму. Для задания алгоритмов можно использовать их графическое представление с помощью

схем, которые должны выполняться в соответствии со стандартами. Для увеличения наглядности схема не должна содержать более 50 блоков.

Задание 6. Выбор стратегии тестирования и разработка тестов. Срок выполнения 15 неделя.

Стратегия тестирования - это план проведения работ по тестированию системы или её модуля, учитывающий специфику функциональности и зависимости с другими компонентами системы и платформы. Стратегия определяет типы тестов, которые нужно выполнять для данного функционала системы, включает описание необходимых подходов с точки зрения целей тестирования.

Стратегия тестирования отвечает на вопросы:

- как, каким образом тестирование даст ответ, что данный функционал работает;
- что нужно сделать и чем пользоваться из инструментальных средств, для достижения целей тестирования;
- когда определённый функционал будет тестироваться и когда ожидать получения результатов.

Тестирование программного обеспечения (ПО) - это процесс выполнения программ на некотором наборе данных, для которого заранее известен результат применения или известны правила поведения этих программ. Указанный набор данных называется тестовым или просто тестом.

Для оптимизации набора тестов, т.е. для подготовки такого набора тестов, который позволял бы при заданном их числе (или при заданном интервале времени, отведенном на тестирование) обнаруживать большее число ошибок в ПО, необходимо, во-первых, заранее планировать этот набор и, во-вторых, использовать рациональную стратегию планирования тестов. Проектирование тестов можно начинать сразу же после завершения этапа внешнего описания ПО.

Оптимальная стратегия проектирования тестов расположена внутри интервала между этими крайними подходами, но ближе к левому краю. Она

включает проектирование значительной части тестов по спецификациям, но она требует также проектирования некоторых тестов и по текстам программ. При этом в первом случае эта стратегия базируется на принципах:

- на каждую используемую функцию или возможность - хотя бы один тест;
- на каждую область и на каждую границу изменения какой-либо входной величины - хотя бы один тест;
- на каждую особую (исключительную) ситуацию, указанную в спецификациях, - хотя бы один тест.

Рассмотрим подробнее методики тестирования «чёрный ящик» и метод тестирования «белый ящик».

Метод тестирования «чёрный ящик». Долгое время основным способом тестирования было тестирование методом «черного ящика» - программе подавались некоторые данные на вход и проверялись результаты, в надежде найти несоответствия. При этом, как именно работает программа, считается несущественным. Отметим, что даже при таком подходе необходимо иметь спецификацию программы для того, чтобы было с чем сравнивать результаты.

Этот подход до сих пор является самым распространенным в повседневной практике, но у него есть целый ряд недостатков. Во-первых, таким способом невозможно найти взаимоуничтожающихся ошибок, во-вторых, некоторые ошибки возникают достаточно редко (ошибки работы с памятью) и потому их трудно найти и воспроизвести.

Метод тестирования «белый ящик». Метод тестирования, которые изучают не только внешнее поведение программы, но и ее внутреннее устройство (исходные тексты). Такие методики обобщенно называют тестированием «белого ящика». Назовем некоторых представителей этого класса методик: чтение программ, формальные просмотры программ, инспекции и т.п.). Основной трудностью подобных методов является сложность отслеживания вычислений времени выполнения.

При тестировании программы методом «белый ящик» происходит проверка логики программы. Полным тестированием в этом случае будет такое, которое приведет к перебору всех возможных путей. Даже для средних по сложности программ числом таких путей может достигать десятков тысяч.

Критерии тестирования. Поскольку исчерпывающее структурное тестирование невозможно, необходимо выбрать такие критерии его полноты, которые допускали бы их простую проверку и облегчали бы целенаправленный подбор тестов. Наиболее слабым из критериев полноты структурного тестирования является требование хотя бы однократного выполнения каждого оператора программы. Более сильным критерием является критерий: каждая ветвь алгоритма (каждый переход) должна быть пройдена (выполнена) хотя бы один раз.

Изучение этих методов тестирования показывает, что они дополняют друг друга, то есть различные методы находят разные ошибки. Поэтому наиболее эффективные процессы разработки программного обеспечения используют некоторую комбинацию методик "черного ящика" и "белого ящика".

Теперь проведём тестирование созданного программного продукта «Файловый менеджер». Для тестирования приложения была выбрана комбинация методик «черного ящика» и «белого ящика».

Протестируем основные функции программы, а это: открытие файлов, копирование файлов, переименование файлов, удаление файлов, создание новых каталогов. Необходимо для начала создать сами тест-требования, которые будут содержать: требование, ожидаемый результат, конечный результат, описание ошибок (если имеются). Образец для заполнения приведен в табл. 1.

Результаты тестирования приложения видим в табл. 2. Ниже приведены критерии тестирования программного продукта.

1. Значения исходных данных (во всех функциях ПС исходные данные-объекты файловой системы компьютера) - исходные данные отображаются в программе «Файловый менеджер» верно.

2. Ожидаемый результат - выполнение соответствующих функций ПС над объектами файловой системы компьютера.

3. Реакция программы - предупреждение об ошибках, совершаемых пользователем при выполнении функции программы. Если ошибок не обнаружено, то приступает к выполнению соответствующих функции приложения.

4. Проверка правильности работы функций приложения с помощью проводника Windows.

5. Возникаемые ошибки и их описание.

Таблица 1

Образец для заполнения тест-требований

№	Входные значения	Ожидаемый результат	Конечный результат	Примечание (возникающие ошибки и их описание)

Таблица 2

Результаты тестирования приложения

Функции прилож.	Открытие файлов	Копирование файлов	Перемещение файлов	Удаление файлов	Создание новых каталогов	
№						
1.	+	+	+	+	+	
2.	+	+	+	+	+	

3.	+	+	+	+	+	
4.	+	+	+	+	+	
5.	-	-	+	+	-	
			(ошибка возникает, если вы хотите перемести ть файл за пределы текущей директори и)	(ошибка возникает , если пользо- ват ель пытается удалить элементы корневой папки System, которую нельзя изменить)		

Следует отметить, что, тестирование не может доказать правильность работы ПО, в лучшем случае оно может продемонстрировать наличие в нем ошибки. Другими словами, нельзя гарантировать, что тестированием ПО практически выполнимым набором тестов можно установить наличие каждой имеющейся в программе ошибки.

По итогам проведённого тестирования приложения комбинацией методик черного и белого ящика все тесты были пройдены успешно. Следовательно, приложение не причиняет вреда объектам файловой системы компьютера и может быть использовано пользователями.

В отчете привести составленные тест-требования и ожидаемые результаты.

Задание 7. Формирование отчета по курсовой работе. Срок выполнения 16 неделя.

Оформление отчета по требованиям и отправка его на проверку.
Назначается дата и время защиты.

Основная часть включает последовательное выполнение варианта задания. В проектирование относится 1-2 задания, в разработку с 3-6 задания. Раздел содержит процесс установки и запуска программного обеспечения, описание всех функциональных возможностей и обработчиков событий со скриншотами.

В отчете должен присутствовать пункт «Руководство пользователя», в котором будет изложена подробная инструкция использования разработанного вами Файлового менеджера.

Руководство пользователя - документ, назначение которого, предоставить людям помощь в использовании системы. Разделы руководства пользователя:

1. Введение.
2. Назначение и условия применения.
3. Подготовка к работе.
4. Описание операций.
5. Аварийные ситуации.
6. Рекомендации по освоению.

Кроме того, необходимо обозначить все аварийные ситуации и способы их устранения, а также привести список минимальных системных требований.

Контрольные вопросы

1. Имя компьютера.
2. Имя пользователя.
3. Версия операционной системы.
4. Физическая память.
5. PID процесса.
6. Приоритет процесса.

7. Файл подкачки.
8. Количество модулей, используемых процессом.
9. Размер рабочего множества страниц.
10. Маска привязки процесса к процессорам.
11. Количество страничных ошибок.
12. Оперативная память.
13. Определение системным утилитами.
14. Понятие лог-файла.
15. Тестирование программного обеспечения.
16. Стратегия тестирования.
17. Файловые менеджеры.
18. Понятие Операционная система.
19. Файловые системы. Понятие и описание.
20. Межпроцессное взаимодействие - Отображение файлов.
21. Межпроцессное взаимодействие - Почтовые ящики.
22. Межпроцессное взаимодействие - Сокеты.
23. Межпроцессное взаимодействие - Каналы.
24. Межпроцессное взаимодействие - Семафор.
25. Межпроцессное взаимодействие - Разделяемая память.
26. Иерархия каталогов и файловых систем.
27. Свойства файлов, папок и дисков.
28. Директория. Понятие и описание.
29. Понятие Процесс. Описание и назначение процессов.
30. Понятие Поток. Описание и назначение потоков.
31. Понятие Многопоточность. Описание и назначение потоков.
32. Логическая организация механизма передачи информации.
33. Организация памяти.
34. Физическое и логическое адресные пространства.
35. Понятие о виртуальной памяти.
36. Принцип адресации.

- 37. Способы организации виртуальной памяти.
- 38. Системные вызовы для управления файлами.
- 39. Системные вызовы для управления каталогами.
- 40. Системные вызовы для процессами.
- 41. Windows Win32 API.

Приложение А

Образец оформления титульного листа курсовой работы

Федеральное агентство связи

Федеральное государственное бюджетное образовательное учреждение высшего
образования

«ПОВОЛЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ И
ИНФОРМАТИКИ»

Факультет ИСТ

Кафедра Программного обеспечения и управления в технических систем (ПОУТС)

ДОПУСТИТЬ К ЗАЩИТЕ

ЗАЩИЩЕН (А) С ОЦЕНКОЙ _____

Руководитель _____
(подпись) (ФИО)

Руководитель _____
(подпись) (ФИО)

(дата)

КУРСОВАЯ РАБОТА

по дисциплине Операционные системы и оболочки

наименование дисциплины (модуля)

наименование темы (при наличии)

ВЫПОЛНИЛ (А)

студент (ка) _____
(группа) (ФИО)

(№ зачетной книжки)

Самара

2020

Приложение В
Образец оформления содержания курсовой работы
Содержание

Задание.....	4
Введение.....	
1 Проектирование приложения	
1.1	
1.2	
.....	
2 Разработка приложения	
2.1	
2.2	
3 Пример работы разработанного приложения	
3.1	
.....	
3.2	
.....	
Заключение.....	
Список использованных источников.....	
Приложение А –	
Приложение Б –	