

Министерство образования Республики Беларусь
Учреждение образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра Информатики

Дисциплина: Архитектура вычислительных систем

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовому проекту на тему

«"Мобильные" аналоги системных утилит»

БГУИР КП I – 40 04 01 034 ПЗ

Выполнил

студент гр. 053504

Слижик Н.С

Проверил:

Давыдчик А.В.

Минск 2023

СОДЕРЖАНИЕ

Введение	3
1 Операционные системы и среды	4
1.1 Описание системы	4
1.2 Утилиты	5
2 Платформа программного обеспечения	7
2.1 ОС Andoid	7
2.2 Среда разработки Visual Studio	8
2.3 Kotlin	11
3 Теоритическое обоснование разработки программного продукта	12
4 Проектирование функциональных возможностей программы	13
5 Архитектура разрабатываемой программы	14
ЗАКЛЮЧЕНИЕ	15
СПИСОК ЛИТЕРАТУРНЫХ ИСТОЧНИКОВ	16
ПРИЛОЖЕНИЯ	17

Введение

Для любого приложения важна операционная система для которой он создается. Скомпилировав код на одном устройстве, далеко не факт, что он будет запланировано работать на другом. Поэтому очень желательно уметь работать с разными платформами. В рамках данной курсовой была выбрана мобильная система Android.

Android – операционная система для смартфонов, планшетов, электронных книг, цифровых проигрывателей, наручных часов, фитнес-браслетов, игровых приставок, ноутбуков телевизоров, проекторов и других устройств. В 86 % смартфонов, проданных во всём мире во втором квартале 2014 года, была установлена операционная система Android. На конференции разработчиков в мае 2017 года Google объявила, что за всю историю Android было активировано более 2 млрд Android-устройств. Таким образом, за счет своей популярности, удобства использования и вариативности устройств в которых она применяется, она является наиболее приоритетным для использования и была выбран для данного проекта.

В ходе данной работы создается утилита. Утилитарное программное обеспечение – это тип системного программного обеспечения, используемого для управления, организации, оптимизации и улучшения функционирования компьютерной системы.

Цель данного проекта - создание утилиты и приобретение навыков работы с мобильной ОС Android.

1 ОПЕРАЦИОННЫЕ СИСТЕМЫ И СРЕДЫ

1.1 Описание системы

Операционные системы и среды - это специальный уровень программного обеспечения, который отвечает за управление всеми устройствами компьютера и обеспечивает пользователя, имеющими простой, доступный интерфейс, программами для работы с аппаратурой.

Как известно, компьютер выполняет действия в соответствии с предписаниями программы, созданной на одном из языков программирования. При работе на компьютере часто возникает необходимость выполнить операции с прикладной программой в целом, организовать работу внешних устройств, проверить работу различных блоков, скопировать информацию и т. д. По существу, эти операции используют для работы с любой программой, воспринимаемой как единое целое. Поэтому, целесообразно, из всего разнообразия операций, выполняемых компьютером, выделить типовые и реализовать их с помощью специализированных программ, которые следует принять в качестве стандартных средств, поставляемых вместе с аппаратной частью. Программы, организующие работу устройств и не связанные со спецификой решаемой задачи, вошли в состав комплекса программ, названного операционными системами.

Операционная система - это совокупность программных средств, обеспечивающая управление аппаратной частью компьютера и прикладными программами, а также их взаимодействие между собой и пользователем.

Операционная система образует автономную среду, не связанную ни с одним языком программирования. Любая же прикладная программа связана с операционной системой и может эксплуатироваться только на тех компьютерах, где имеется аналогичная системная среда. Прикладные программные средства, разработанные в среде одной операционной системы, не могут быть использованы в среде другой операционной системы, если нет возможности конвертации (преобразования) программ. В таком случае говорят о программной несовместимости компьютеров. Обычно операционная система хранится на жёстком диске, а при его отсутствии выделяется отдельный гибкий диск, который называется системным диском. При включении компьютера операционная система автоматически загружается с диска в оперативную память и занимает в ней определённое

место. Операционная система создаётся не для отдельной модели компьютера, а для серии компьютеров, в структуре которых заложена и развивается во всех последующих моделях определённая концепция.

1.2 Утилиты

Операционные системы и утилиты - это программы, которые используются для управления и обслуживания компьютерных систем. Однако между этими двумя типами программ существует ключевое различие. Операционные системы предназначены для управления аппаратными и программными ресурсами компьютера, в то время как утилиты предназначены для выполнения конкретных задач.

Одной из важнейших функций операционной системы является управление памятью компьютера. Это включает в себя отслеживание того, какие программы используют те или иные ресурсы, и выделение ресурсов по мере необходимости. Операционная система также должна быть многозадачной, то есть она должна быть способна выполнять несколько программ одновременно. Для этого операционная система должна уметь переключаться между программами и решать любые возникающие конфликты.

Утилиты, с другой стороны, предназначены для выполнения конкретных задач. Обычными примерами утилит являются программы, позволяющие управлять файлами, обеспечивающие безопасность и обслуживание. Хотя утилиты могут поставляться в комплекте с операционной системой, они могут быть и самостоятельными программами.

В целом, ключевое различие между операционными системами и утилитами заключается в том, что операционные системы предназначены для управления аппаратными и программными ресурсами компьютера, в то время как утилиты предназначены для выполнения конкретных задач.

Ресурсы компьютера управляются, обслуживаются и контролируются служебными программами. Утилита может быть любым типом программного обеспечения, которое работает в сочетании с антивирусной программой, программой резервного копирования или дисковым инструментом. Windows является наиболее широко используемой операционной системой (ОС) для распространения программ-утилит. Некоторые системные утилиты могут быть установлены в операционной системе, а другие добавляются позже. Программное обеспечение, которое создают пользователи, является

самодостаточным и способно выполнять определенную задачу. Это компьютерная программа, которая помогает функционировать компьютеру в целом. К ним относятся антивирусные программы, программы резервного копирования, файловые менеджеры и средства сжатия дисков. Программы, которые можно использовать для управления, обслуживания и контроля ресурсов компьютера, значительно облегчают работу.

Операционные системы, такие как Windows, Linux и Android, позволяют пользователю запускать на компьютере или мобильном устройстве такие программы, как MS Office, блокнот и игры. Для запуска основных программ, таких как браузеры, на компьютере должна быть установлена рабочая операционная система.

2. ПЛАТФОРМА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

2.1 ОС Android

Некоторые обозреватели отмечают, что Android проявляет себя лучше одного из своих конкурентов, Apple iOS, в ряде особенностей, таких как веб-сёрфинг, интеграция с сервисами Google Inc. и прочих. Также Android, в отличие от iOS, является открытой платформой, что позволяет реализовать на ней больше функций.

- В Android-аппаратах, как правило, присутствует microSD-кардридр, делающий возможным быстрый перенос файлов с компьютера на телефон. Кроме того, в Windows Phone 7 невозможна прямая передача каких-либо файлов с телефона и на телефон, кроме как через программы синхронизации (Zune), в то время как телефоны на Android экспортируют файловую систему карты памяти как *USB mass storage device* (в версиях начиная с 3.0 на усмотрение производителя) или с помощью Media Transfer Protocol (MTP).

- В отличие от Windows Phone 7, в Android заявлена полноценная реализация Bluetooth-стека, позволяющая в том числе передачу и приём файлов. Присутствует реализация FTP-сервера, режима точки доступа к сети (службы PAN) и групповой одноранговой сети через Bluetooth (службы GN).

- Несмотря на изначальный запрет на установку программ из «непроверенных источников» (например, с карты памяти), это ограничение отключается штатными средствами в настройках аппарата, что позволяет устанавливать программы на телефоны и планшеты без интернет-подключения (например, пользователям, не имеющим Wi-Fi-точки доступа и не желающим тратить деньги на мобильный интернет, который обычно стоит дорого), а также позволяет всем желающим бесплатно писать приложения для Android и тестировать на своём аппарате.

- Android доступен для различных аппаратных платформ, таких как ARM, MIPS, x86.

- Существуют альтернативные Google Play магазины приложений: Amazon Appstore (*англ.*), Opera Mobile Store, Yandex.Store, GetUpps!, F-Droid.

- В версии 4.2 введена поддержка многопользовательского режима.

Критика

- Наличие в некоторых Android-устройствах сервисов Google,

обеспечивающих возможность передачи идентификационной информации на серверы компании, например, информацию о перемещении пользователя в реальном времени.

- В версии Android 1.6 разработчики добавили Native Development Kit, который позволяет писать собственные низкоуровневые модули для системы на языках C/C++, опираясь на стандартные Linux-библиотеки. Хотя, например, стандартная библиотека языка Си на платформе Android, известная как Bionic, как раз не является стандартной и полностью совместимой с libc.

- Для доступа к Google Play и другим сервисам от Google необходимо использовать проприетарные приложения, которые производитель телефона имеет право устанавливать на телефон только после заключения контракта с Google.

- Конкуренты Android выступили с критикой платформы, обвиняя её в чрезмерной фрагментации, создающей препятствия разработчикам. Google опровергла все обвинения, заявив, что никаких подобных проблем нет.

- Подвергается критике решение Google не размещать в широком доступе код Android 3.0 *Honeycomb*, доступный только для участников Open Handset Alliance или по индивидуальному запросу после подписания соглашения. Google мотивирует это неготовностью платформы и мерой предупреждения небрежной её реализации. Ричард Столлман заявлял, что «всё просто и ясно: за исключением ядра Linux, Android 3 представляет собой несвободный софт» и «несмотря на то, что телефоны с Android на сегодняшний день не так плохи, как смартфоны Apple или Windows, нельзя сказать, что они уважают вашу свободу». В настоящий момент Google открыл исходный код для всех версий Android вплоть до 4.4, а также отправил в Linux все изменения, в соответствии с GPL.

По данным Lookout Security Mobile, за 2011 год у пользователей Android-смартфонов было украдено около миллиона долларов США (напр., отправка СМС без ведома владельца телефона). Однако это зачастую возникает у пользователей, которые невнимательно читают списки разрешений.

2.2 Среда разработки Visual Studio Code

Visual Studio Code (рисунок 6) - это сервис, который позиционируется как «легкий» редактор кода для кроссплатформенной разработки веб - и облачных приложений, разработанный *Microsoft* для *Windows*, *Linux* и *macOS*.

Включает в себя отладчик, инструменты для работы с Git, подсветку синтаксиса, *IntelliSense* и средства для рефакторинга. Имеет широкие возможности для кастомизации: пользовательские темы, сочетания клавиш и файлы конфигурации. Распространяется бесплатно, разрабатывается как программное обеспечение с открытым исходным кодом, но готовые сборки распространяются под проприетарной лицензией.

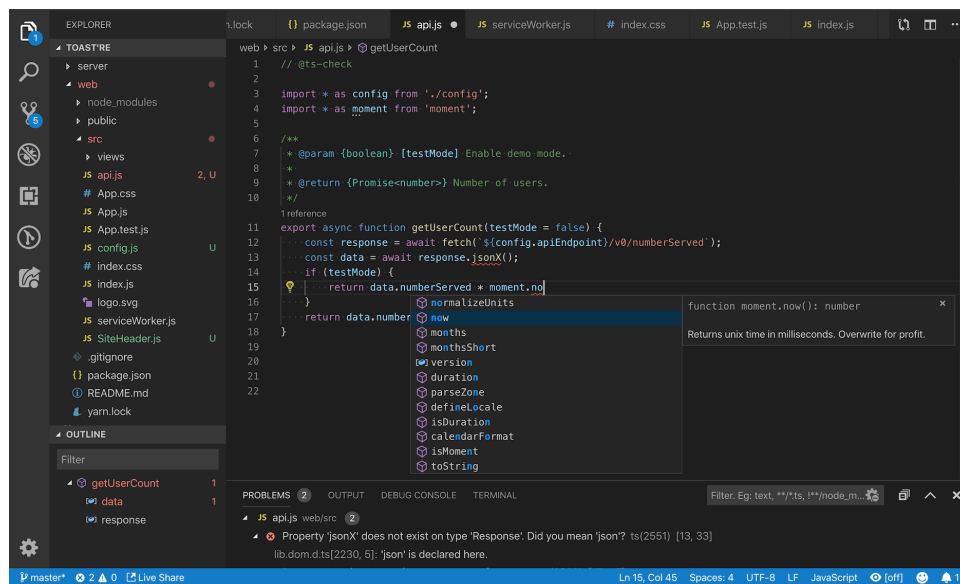


Рисунок 1 – интерфейс VS Code

Особенности Visual Studio Code, которые выделяют её среди конкурентов:

- VS Code позволяет разрабатывать как консольные приложения, так и приложения с графическим интерфейсом, в том числе с поддержкой технологии Windows Forms, а также веб - сайты, веб-приложения, веб-службы как в родном, так и в управляемом кодах для всех платформ.
- В редакторе присутствуют встроенный отладчик, инструменты для работы с Git и средства рефакторинга, навигации по коду, автодополнения типовых конструкций и контекстной подсказки.
- Продукт поддерживает разработку для платформ ASP.NET и Node.js,
- Большим плюсом редактора является поддержка большого количества языков, таких как C++, C#, Python, PHP, JavaScript и других.
- Возможности Visual Studio Code
- встроенные инструменты интеграции с GitHub, GIT, а также

Visual Studio Team Services для быстрого тестирования, сборки, упаковки и развертывания разных типов приложений;

- удобство работы с Unity-проектами;
- работа с Mono и Node.js с помощью встроенного отладчика;
- Поддержка TypeScript и JavaScript;
- публикация созданных приложений в Microsoft Azure через сервис Visual Studio Team Services;
- поддержка практически всех языков программирования;
- написание кода для конкретной задачи с его последующей интеграцией в проект (с надстройкой или напрямую);
- Обширная библиотека шаблонов, готовых фрагментов кода и сниппетов с возможностью добавления своих элементов;
- одновременная работа с несколькими проектами (в нескольких окнах);
- интерфейс можно разделить на две панели для сравнения кода;
- функция отладки.

Преимущества Visual Studio Code

- множество настроек (как всей программы, так и интерфейса);
- расширяемая библиотека дополнений и готовых решений;
- мультифункциональность (редактор поддерживает почти все языки, используемые для создания приложений);
- простота и гибкость.

Также у VS Code есть ряд полезных настроек для автоматизации рабочего процесса:

- Автосохранение. Найдите параметр Files: Auto Save. Измените значение с «Off» на «afterDelay», задайте промежуток времени — и файлы будут автоматически сохраняться в заданный промежуток.
- Размер шрифта. Параметр Font Size. По умолчанию 14, можно изменить на удобный.
- Форматирование кода. Параметр Format On Paste. Код будет автоматически форматироваться при вставке. Позволяет исправить проблемы с лишними отступами.
- Удаление конечных пробелов. Параметр Trim Trailing Whitespace.

При сохранении из документа автоматически удаляются пробелы в конце строк. Это нужно, чтобы не перегружать код лишними символами и проходить некоторые проверки на стиль кода.

Некоторые изменения в программе нужно задавать не через меню настроек, а через расширения. Например, чтобы русифицировать VS Code, нужно открыть вкладку с расширениями, ввести запрос `vscode-language-pack-ru` и нажать кнопку «Install». После этого язык интерфейса изменится на русский.

2.2 Kotlin

Kotlin представляет современный, статически типизированный и один из самых быстроразвивающихся языков программирования, созданный и развиваемый компанией JetBrains. Kotlin можно использовать для создания самых разных приложений. Это и приложения для мобильных устройств - Android, iOS. Причем Kotlin позволяет писать кроссплатформенный код, который будет применяться на всех платформах. Это и веб-приложения, причем как серверные приложения, которые отработывают на стороне сервера - бекэнда, так и браузерные клиентские приложения - фронтенд. Kotlin также можно применять для создания десктопных приложений, для Data Science и так далее.

Таким образом, круг платформ, для которых можно создавать приложения на Kotlin, чрезвычайно широк - Windows, Linux, Mac OS, iOS, Android.

Самым популярным направлением, где применяется Kotlin, является прежде всего разработка под ОС Android. Причем настолько популярным, что компания Google на конференции Google I/O 2017 провозгласила Kotlin одним из официальных языков для разработки под Android (наряду с Java и C++), а инструменты по работе с данным языком были по умолчанию включены в функционал среды разработки Android Studio начиная с версии 3.0.

Первая версия языка вышла 15 февраля 2016 года. Хотя сама разработка языка велась с 2010 года. Текущей версией языка на данный момент является версия 1.5, которая вышла 5 мая 2021 года.

Kotlin испытал влияние многих языков: Java, Scala, Groovy, C#, JavaScript, Swift и позволяет писать программы как в объектно-ориентированном, так и в функциональном стиле. Он имеет ясный и понятный синтаксис и довольно легок для обучения.

3 ТЕОРИТИЧЕСКОЕ ОБОСНОВАНИЕ РАЗРАБОТКИ ПРОДУКТА

Получение максимальной пользы без вреда используемому - это постоянное желание любого профессионала и опытного пользователя. Но добиться этого бывает не просто. Где-то мешают ограничения компонентов, где-то ограничения безопасности, а иногда просто невозможность выполнения нужных действий с помощью стандартных приложений. В таких ситуациях возможно использовать специализированные утилиты.

Преимущества утилитарного программного обеспечения:

- Повышает производительность.
- Управляет пространством.
- Управление файлами и данными.
- Помогает настроить интерфейс и рабочий стол.
- Помогает удалить бесполезные файлы.
- Обеспечивает безопасность и сохранность системы.
- Помогает восстановить файлы после потери.

4 ПРОЕКТИРОВАНИЕ ФУНКЦИОНАЛЬНЫХ ВОЗМОЖНОСТЕЙ ПРОГРАММЫ

Интерфейс приложения выполнен в соответствии с требованиями, которые предъявлялись к системе на этапе проектирования. Простота и удобство использования приложения обеспечивает выполнение качественной и многофункциональной работы без напряжения и без затрат времени на осмысление информации, которая отображается на экране.

Простота интерфейса приложения подразумевает, что не нужно усложнять восприятие и понимание информации, возникающей перед глазами пользователя. Для этого при выполнении очередного шага задания перед пользователем будет отображаться минимально необходимая информация.

Также были исключены многословные командные имёна и сообщения и элементы управления размещены на экране с учётом их смыслового значения и логической взаимосвязи. При запуске программы открывается главное окно приложения.

5 АРХИТЕКТУРА РАЗРАБАТЫВАЕМОЙ ПРОГРАММЫ

Архитектура системы:

Железо – аппаратное обеспечение с его периферийными устройствами.

Ядро – основной компонент ОС, взаимодействует с аппаратным обеспечением, выступает посредником между низкоуровневым железом и компонентами верхнего уровня.

Оболочка – интерфейс для взаимодействия между ядром ОС и пользователями, абстрагирует внутреннее устройство системы, получает команды от пользователей, запускает необходимые функции.

Утилиты – служебные программы, дающие пользователю большую часть функциональных возможностей. Например для System OS Linux – sed, cat, date, vi и прочие.

Компиляторы – технические средства, преобразующие программный код в машинный.

ЗАКЛЮЧЕНИЕ

В результате появляются продукт, предназначенный для личного пользования. Интерфейс прост и понятен любому пользователю, а любое взаимодействие, которое может произойти в области приложения, и исключительные ситуации во внутренней работе обрабатывается во избежание ошибок.

В данной работе выполнены поставленные для этого продукта цели, требования и исследованна и реализована совместимость с особенностями операционной системой.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

- [1] Love, R. Linux System Programming / Love, R – Newton ; O'Reilly Media. Press, 2007. – 456 p .
- [2] StudFile [Электронный ресурс]. – Файловый архив студентов | Операционные системы. – Режим доступа : <https://studfile.net/preview/11218660>. – Дата доступа: 01.11.2022.
- [3] Negus, C. Ubuntu Linux Toolbox / Negus, C. – Engels ; John Wiley & Sons, Press, 2013. – 368 p .
- [4] StudFile [Электронный ресурс]. – Файловый архив студентов | Операционные системы. – Режим доступа : <https://studfile.net/preview/11218660>. – Дата доступа: 01.11.2022.
- [5] Gnome Extensions[Электронный ресурс]. – Создание расширений. – Режим доступа : <https://extensions.gnome.org>. – Дата доступа: 03.11.2022.
- [6] Литвиенко, Н.А. Технология программирования на C++. Win32 API-приложения. : Учеб. пособие / Н.А. Литвинко. – Санкт-Петербург : ИТМО, 2010. – 281 с.
- [7] Лаборатория линуксоида [Электронный ресурс]. – Введение в Linux и Bash | Каталоги. – Режим доступа : <https://younglinux.info/bash/linuxfolder>. – Дата доступа: 01.11.2022.
- [8] Habr. Основы Linux [Электронный ресурс]. – Системное администрирование | Unix. – Режим доступа : <https://habr.com/ru/post/655275/>. – Дата доступа: 01.11.2022.
- [9] SkillFactory [Электронный ресурс]. – Linux. – Режим доступа : <https://blog.skillfactory.ru/glossary/linux/>. – Дата доступа: 01.11.2022.
- [10] GTK [Электронный ресурс]. – GTK. – Режим доступа : <https://www.gtk.org/>. – Дата доступа: 01.11.2022.
- [11] OpenNet [Электронный ресурс]. – Основы Gtk+. – Режим доступа : <https://www.opennet.ru/docs/RUS/gtk/node37.html>. – Дата доступа: 01.11.2022.
- [12] Bauman National Library [Электронный ресурс]. – GIO. – Режим доступа : [https://ru.bmstu.wiki/GIO_\(Gnome_Input/Output\)](https://ru.bmstu.wiki/GIO_(Gnome_Input/Output)). – Дата доступа: 01.11.2022.
- [13] Qwerty [Электронный ресурс]. – Типы и классы ноутбуков. – Режим доступа : <https://qwertyshop.ua/stati/kakie-buvaut-noybtbki>. – Дата доступа: 01.11.2022.
- [14] Qwerty [Электронный ресурс]. – Типы и классы ноутбуков. – Режим доступа : <https://qwertyshop.ua/stati/kakie-buvaut-noybtbki>. – Дата доступа: 01.11.2022.
- [15] GnulinuxPro [Электронный ресурс]. – Операционные системы. – Режим доступа : <https://basis.gnulinux.pro/ru/latest/basis/01/01.html#>. – Дата доступа: 02.11.2022.

[16] Forbes [Электронный ресурс]. – Статья. – Режим доступа : <https://www.forbes.com/sites/patrickmoorhead/2022/09/29/the-ibm-linuxone-driving-sustainability-within-digital-transformation/?sh=49c3f4121b4a>. – Дата доступа: 02.11.2022.

[17] HtmlAcademy [Электронный ресурс]. – Средства разработки. – Режим доступа : <https://htmlacademy.ru/blog/soft/vscode>: 02.11.2022.

[18] UPower Documentation [Электронный ресурс]. – Архив. – Режим доступа : <https://upower.freedesktop.org/docs/Device.html>: 02.11.2022.

[19] DevsDay.ru [Электронный ресурс]. – Частота процессора в Linux. – Режим доступа : <https://devsday.ru/blog/details/27043>: 02.11.2022.