

РАЗРАБОТКА ПРИЛОЖЕНИЯ ДЛЯ АНАЛИЗА И ОБРАБОТКИ ЛОГ-ФАЙЛОВ С ПОЛЕТНОЙ ИНФОРМАЦИЕЙ

DEVELOPMENT OF AN APPLICATION FOR THE ANALYSIS AND PROCESSING OF LOG FILES WITH FLIGHT INFORMATION

**V. Gladun
A. Pantykhin
I. Malinin
N. Voinov**

Summary. The work is about development of a program that allows extracting data from log files with flight information and analyzing it. The program must be able to work with many such files and structure the received data, build graphs and generate reports. The flight data of the tested aircraft are analyzed for further conclusions and possible adjustments to the aircraft design.[1]

Software of this type allows you to simplify the process of testing test samples, provide engineers with information about possible shortcomings or malfunctions.

As a result of the work, a system was implemented that allows reading and processing large data streams obtained from log files, as well as analyzing the information received. The information is structured in the form of reports and graphs for the convenience of presenting information to engineers.

Keywords: log files, CRC, flight information, data analysis.

Гладун Владимир Вадимович

Санкт-Петербургский политехнический университет Петра Великого
vladimir.gldn@gmail.com

Пантюхин Андрей Максимович

Санкт-Петербургский политехнический университет Петра Великого
panandafog@gmail.com

Малинин Илья Игоревич

Санкт-Петербургский политехнический университет Петра Великого
malinin.ilja@gmail.com

Воинов Никита Владимирович

Кандидат технических наук, Доцент,
Санкт-Петербургский политехнический университет Петра Великого
voinov_nv@spbstu.ru

Аннотация. В статье рассматривается разработка программы, позволяющей извлекать данные из лог-файлов с полетной информацией и анализировать их. Программа должна уметь работать с большим количеством таких файлов и структурировать полученные данные, строить графики и формировать отчеты. Анализируются данные полета тестируемого самолета для дальнейших выводов и возможных корректировок конструкции самолета.[1]

ПО такого типа позволяет упростить процесс тестирования тестовых образцов, предоставить инженерам информацию об возможных недочетах или неисправностях.

По итогам работы была реализована система, позволяющая считывать и обрабатывать большие потоки данных, получаемые из лог-файлов, а также анализировать, полученную информацию. Информация структурируется в виде отчетов и графиков для удобства представления информации инженерам.

Ключевые слова: лог-файлы, CRC, полетная информация, анализ данных.

Введение

Проектирование летательного средства является очень сложным и долгим процессом. Методом проб и ошибок создаются как большие авиалайнеры, так и небольшие беспилотные средства. При запуске тестового летательного средства замеряется перечень параметров, по которым в дальнейшем происходит анализ полета данного летательного средства. По полученным данным инженеры приходят к выводу о неполадках и возможных недочетах конструкции.

Во время тестового полета происходит запись изменения параметров, положения самолета. Эти данные хранятся в виде байт-кода в лог-файлах и без специаль-

ного программного обеспечения проблематично вычленивать нужные данные и ознакомиться с ними. Для этого создается система, позволяющая быстро и удобно представить информации о полете летательного средства. Поток данных структурируется в отчеты и представляется в виде графиков для быстрого нахождения неисправностей конструкции инженерами.

Программа мониторинга полетных данных помогает эксплуатанту выявлять, количественно определять, оценивать и устранять эксплуатационные риски. Разработанную систему можно эффективно использовать для поддержки ряда задач по обеспечению летной годности и безопасности полетов.

Анализ предметной области

В науке анализ данных использует более сложный подход с передовыми методами для изучения и экспериментов с данными. С другой стороны, в бизнес-контексте данные используются для принятия решений на основе данных, которые позволят компании повысить свою общую производительность.

Для проверки целостности данных используется код обнаружения ошибок CRC. [6]

CRC — это код обнаружения ошибок, используемый для проверки целостности данных. Он работает так же, как контрольная сумма, добавляется в конец данных полезной нагрузки и передается (или сохраняется) вместе с этими данными.

Значение проверки называется избыточным, поскольку оно не добавляет к сообщению никакой дополнительной информации. (Эти данные добавляются только для обнаружения ошибок и целостности данных). 0111 1010 содержит ту же информацию, что и 0111 1010 1000 (где последние 4 бита — CRC), однако в первом случае получатель не имеет возможности проверить правильность приема данных. [7]

CRC специально разработаны для обнаружения распространенных ошибок передачи данных. Он также может обнаруживать изменение порядка битов или байтов.

Существует несколько возможных реализаций CRC:

- Побитовая реализация. Расчеты выполняются побитно, что требует много времени расчета. Вариант полезен для понимания алгоритма, но не для практического использования в приложениях ПО.
- Метод таблицы поиска. Реализация по байтам.

Побайтовый метод требует сохранения массива из 256 чисел. CRC8: 256 байт, CRC16: 256×16 битных слов (512 байт), CRC32: 256×32 битных слов (1 КБ), CRC64: 256×64 битных слов (2 КБ). Обычно это самый быстрый метод реализации CRC.

Существует большое разнообразие порождающих многочленов для алгоритмов нахождения контрольной CRC — 8, 16 и 32, подобранных на основе теории кодирования и многочисленных исследований. [8]

CRC-8: $x^8 + x^7 + x^6 + x^4 + x^2 + 1$ — используется в лог-файлах.

В программе используется прямой табличный алгоритм вычисления контрольной суммы.

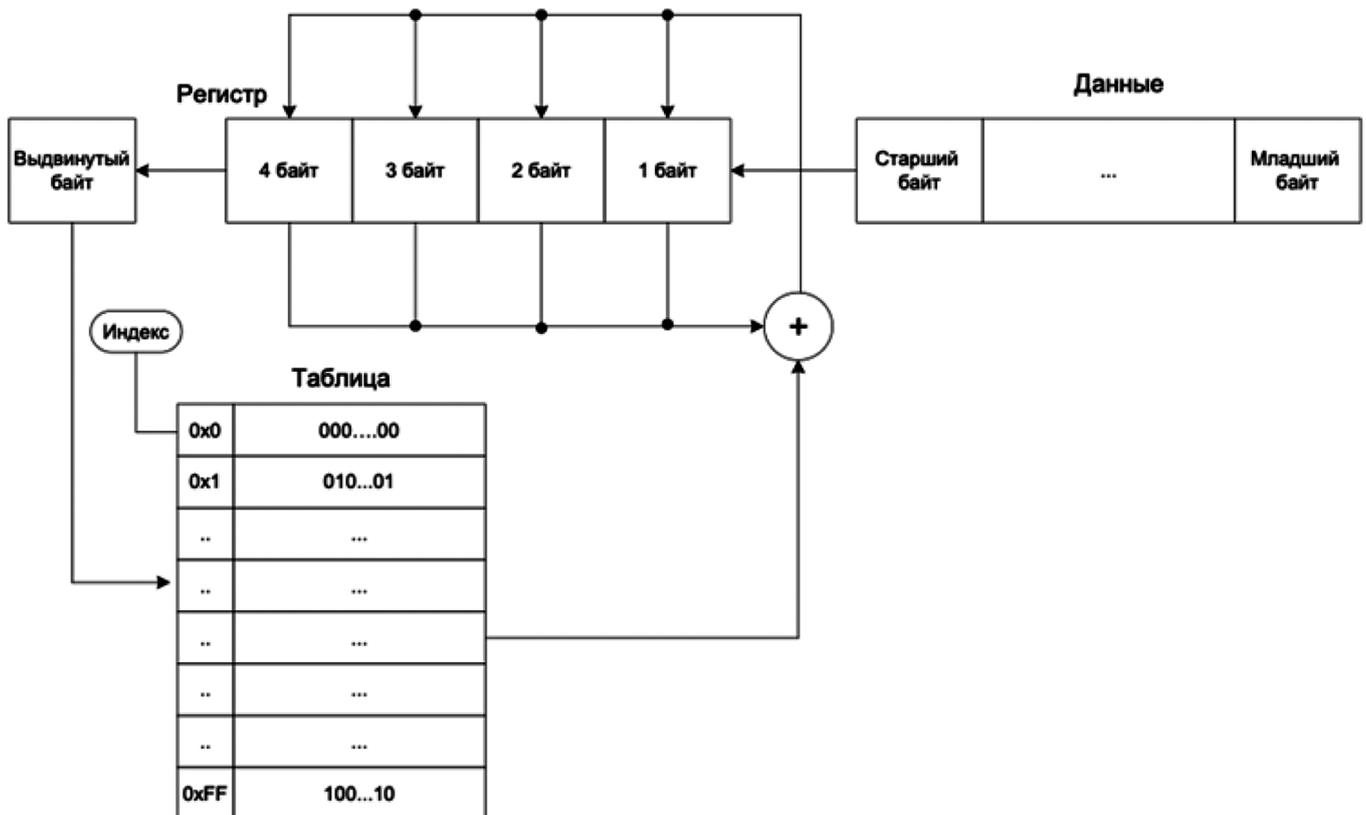


Рис. 1. Схематическое представление прямого табличного алгоритма

Прямой табличный алгоритм основывается на том, что при вычислении контрольной суммы можно оперировать байтами, а также с таблицей, рассчитанной на основе образующего полинома.

Разработанная система проверяет каждый полученный пакет из лог-файлов на целостность. Программой анализируются только проверенные на целостность пакеты.

В авиации полет самолета делят на стадии:

- Взлет — это фаза полета, в которой самолет проходит через переход от движения по земле (такси) к полету в воздухе, обычно начиная с взлетно-посадочной полосы. Обычно двигатели работают на полную мощность во время взлета. После движения такси самолет останавливается на стартовой линии взлетно-посадочной полосы. Перед взлетом двигатели, особенно поршневые двигатели, обычно работают на высокой мощности, чтобы проверить наличие проблем, связанных с двигателем.
- Набор высоты — после взлета самолет должен подняться на определенную высоту, прежде чем он сможет плавать на этой высоте безопасным и экономичным способом.
- Круиз — это ровная часть путешествия самолета, где полет наиболее экономичен. Это происходит между фазами подъема и спуска и обычно составляет большую часть полета. Технически круиз состоит из изменений направления (направления полета) только при постоянной воздушной скорости и высоте. Он заканчивается, когда самолет приближается к месту назначения, где начинается этап спуска в рамках подготовки к посадке.
- Спуск — это любая часть, где самолет уменьшает высоту.
- Посадка — это последняя часть полета, где самолет возвращается на землю. Самолеты обычно приземляются в аэропорту на прочной взлетно-посадочной полосе, обычно построенной из асфальтобетона, бетона, гравия или травы. Чтобы приземлиться, скорость воздуха и скорость спуска снижаются туда, где объект опускается достаточно медленно, чтобы обеспечить мягкое приземление. Посадка достигается за счет замедления и спуска на взлетно-посадочную полосу.

В работе эти этапы сжаты до трех основных. Это позволило максимально упростить нахождения ошибок, возникших во время полета и выявленных на этапе обработки и анализа данных, ускорить работу программы.

Основными этапами полета самолета являются: взлет, прямой полет, снижение. Эти стадии могут происходить в разном порядке и с разными временными промежутками.

Для определения этих стадий и возможности отследить вышеприведенные этапы полета самолета была разработана методика, основанная на анализе нескольких параметров, по значениям которых в определенные промежутки времени можно точно определить в какой из данных стадий находится самолет.

Выбор технологий и инструментов

Для данной задачи был выбран язык Python. Python является интерпретируемым высокоуровневым языком программирования. Язык поддерживает разработку программ с использованием структурного, функционального, объектно-ориентированного подходов. Важными и отличительными чертами для выбора данного языка при разработке данной системы являются кроссплатформенность, динамическая типизация, автоматическое управление памятью, интроспекция, высокоуровневые структуры данных.

Синтаксис легко читаем, что сильно упрощает разработку и модернизацию системы в дальнейшем. С помощью Python можно легко создать как системное, так и прикладное программное обеспечение, программы для научных расчетов. Python обладает обширной стандартной библиотекой, а также для Python написано большое количество прикладных библиотек, позволяющих решать самые разные математические задачи, что сильно упрощает работу с научными расчетами.

Лог-файлы записаны в бинарные файлы и прочитать их без специального программного обеспечения трудно. Разработанная система считывает данные из лог-файлов и переводит их в целочисленные представления. Некоторые параметры в лог-файлах записаны не в системе СИ, поэтому программа переводит данные параметры в международную систему единиц.

В отчете присутствуют лишь основные параметры, по которым проводится анализ, а также определяется стадия полета.

Отчет представлен в форматах html и pdf, однако pdf файл занимает меньше памяти жесткого диска компьютера. В отчетах приведены параметры, по значениям которых можно понять о наличии выходов за пределы, заданных значений параметров, а также смене стадий полета.

Реализация проекта

Результатом работы является разработанное приложение на языке Python для анализа лог-файлов полета самолета. Данное приложение может быть использовано при тестировании летательных средств.

Графический интерфейс разработан с использованием библиотеки Tkinter. Python реализует Tkinter в качестве модуля. Tkinter — это обертка расширений C, которые используют библиотеки Tcl/Tk. Tkinter позволяет разрабатывать настольные приложения. Это очень хороший инструмент для программирования с графическим интерфейсом на Python. Tkinter — хорошо подходит для реализации интерфейса разработанной системы по следующим причинам: используется очень мало кода, чтобы создать функциональное настольное приложение, многослойный дизайн, портативный во всех операционных системах, включая Windows, macOS и Linux, предустановлено со стандартной библиотекой Python.

Пользователю представлено 5 вкладок с различным функционалом, требуемым для работы с программным обеспечением такого типа. Скриншоты некоторых элементов интерфейса программы представлены ниже.

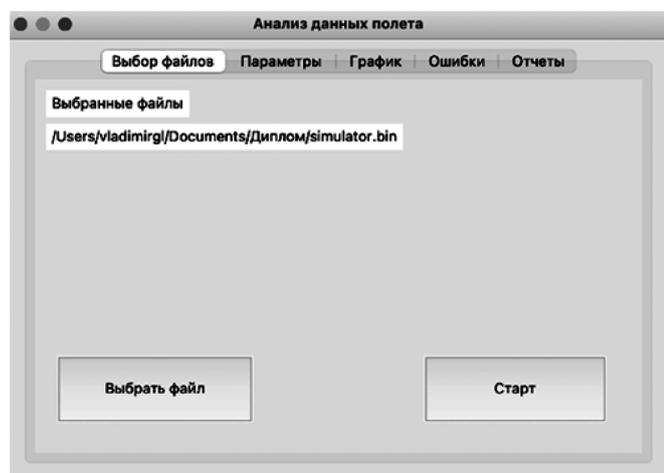


Рис. 2. Вкладка «Выбор файлов», пример выбора и загрузки файла в программу

Разработанная система позволяет считывать большие объёмы информации, получаемой из лог-файлов,

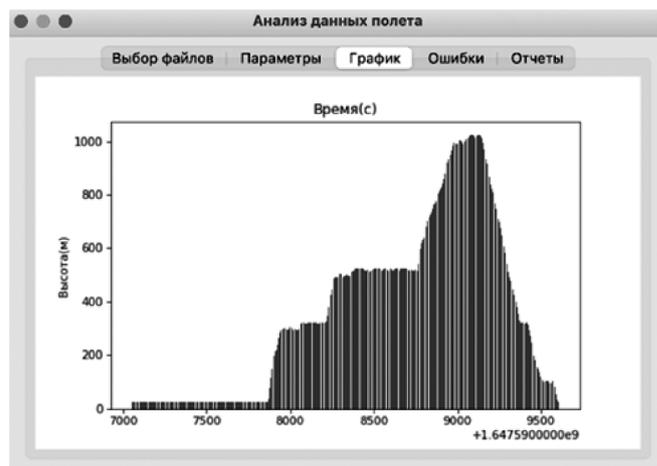


Рис. 3. Вкладка «График», построенный график

Date	N	Air Speed	Air Alt	GPS Alt	Rot Alt	RPM	Engine Temp	Exit Temp	Roll
2022-03-18 09:50:58	0	0.0	0.0	0.0	0.0	128	146	0.0	
2022-03-18 09:50:58	1	0.0	0.0	1000	0.0	128	146	1.0	
2022-03-18 09:50:58	2	0.0	0.0	1000	0.0	128	146	1.0	
2022-03-18 09:50:58	3	0.0	0.0	500	0.0	128	146	1.0	
2022-03-18 09:50:58	4	0.0	0.0	500	0.0	128	146	1.0	
2022-03-18 09:50:58	5	0.0	0.0	500	0.0	128	146	0.0	
2022-03-18 09:50:58	6	0.0	0.0	500	0.0	128	146	0.0	
2022-03-18 09:50:58	7	0.0	0.0	500	0.0	128	146	0.0	

Рис. 4. Вкладка «Отчеты», просмотр отчета в программе

а также структурировать и анализировать. Данная работа имеет свои преимущества перед аналогами, система может распространяться бесплатно, а также быть использована на многих ОС.

ЛИТЕРАТУРА

1. Voinov N., Rodriguez Garzon K., Nikiforov I., & Drobintsev P. (2019). Big data processing system for analysis of GitHub events. In Proceedings of 2019 22nd International Conference on Soft Computing and Measurements, SCM 2019 (pp. 187–190). St. Petersburg, 23–25 May 2019. St. Petersburg. DOI: 10.1109/SCM.2019.8903782.10.1109/SCM.2019.8903782.
2. Derek Hawkins. [Электронный ресурс]. Режим доступа: <https://derek-m-hawk.medium.com/almost-end-to-end-log-file-analysis-with-python-59d4cb30f930>
3. Python Documentation. [Электронный ресурс]. Режим доступа: <https://docs.python.org/3/>
4. Model Oriented Approach for Industrial Software Development / P.D. Drobintsev, V.P. Kotlyarov, N.V. Voinov, I.V. Nikiforov // Modeling and Analysis of Information Systems. — 2015. — Vol. 22. — No 6. — P. 750–762. — DOI 10.18255/1818-1015-2015-6-750-762.
5. Система наземной обработки информации. [Электронный ресурс]. Режим доступа: <https://spectr.spb.ru/snoi.html>
6. Целостность данных, CRC. [Электронный ресурс]. Режим доступа: <https://alexgyver.ru/lessons/crc/>
7. How to Validate Your Data with a Cyclic Redundancy Check (CRC). [Электронный ресурс]. Режим доступа: <https://quickbirdstudios.com/blog/validate-data-with-crc/>
8. CRC RevEng. [Электронный ресурс]. Режим доступа: <https://reveng.sourceforge.io/crc-catalogue/16.htm#crc.cat.arc>

© Гладун Владимир Вадимович (vladimir.gldn@gmail.com); Пантюхин Андрей Максимович (panandafog@gmail.com); Малинин Илья Игоревич (malinin.ilja@gmail.com); Воинов Никита Владимирович (voinov_nv@spbstu.ru)
Журнал «Современная наука: актуальные проблемы теории и практики»