

# Разработка метода фаззинг-тестирования драйверов файловых систем для UEFI-окружения

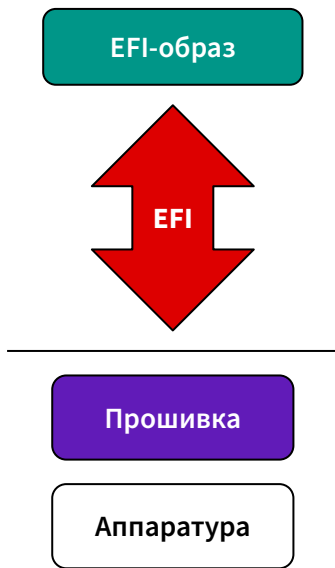
Набережнев Павел Александрович (НИУ ВШЭ)

Научный консультант: Виталий Юрьевич Чепцов (ИСП РАН)

Научный руководитель: Алексей Владимирович Хорошилов (ИСП РАН)

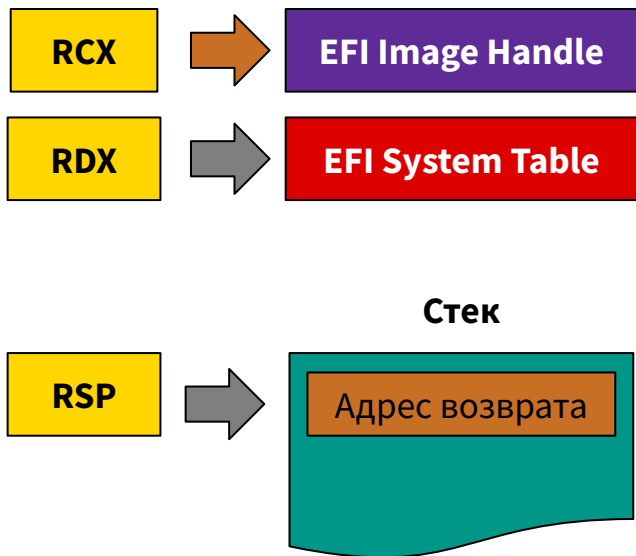
Москва, 2025 г.

# Unified Extensible Firmware Interface



- Спецификация  
→ <https://uefi.org/specifications>
- i386, IA-64, AMD64, ARM32, ARM64, RISC-V, LoongArch
- Инициализация прошивки платформы
- Формат разметки загрузочных носителей (GPT)
- Загрузка исполняемых образов в среду
- Предоставляет интерфейс взаимодействия с прошивкой платформы для исполняемых образов

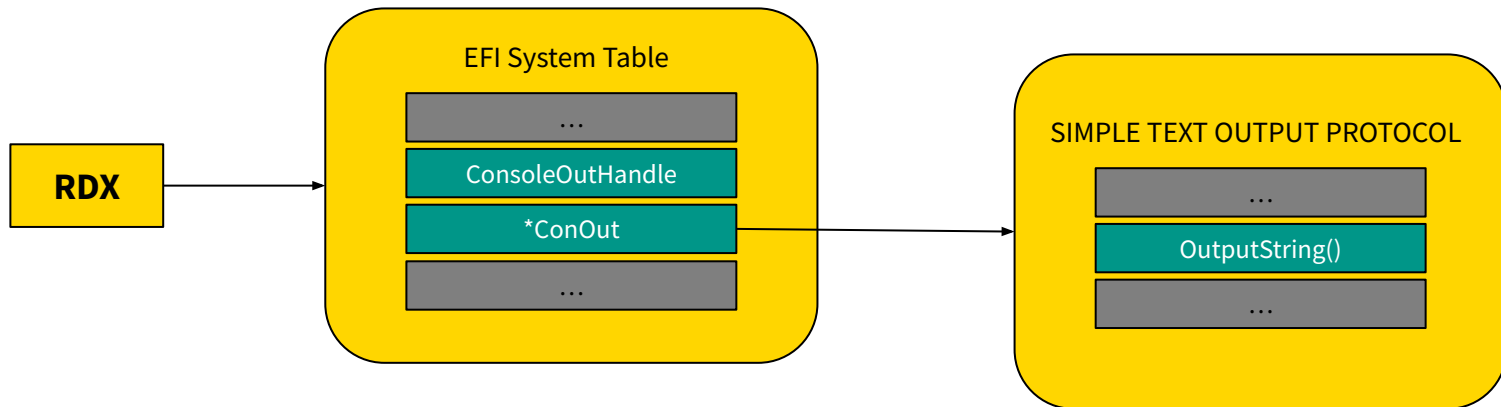
# UEFI-протоколы



Состояние машины после  
загрузки исполняемого образа на  
платформе AMD64

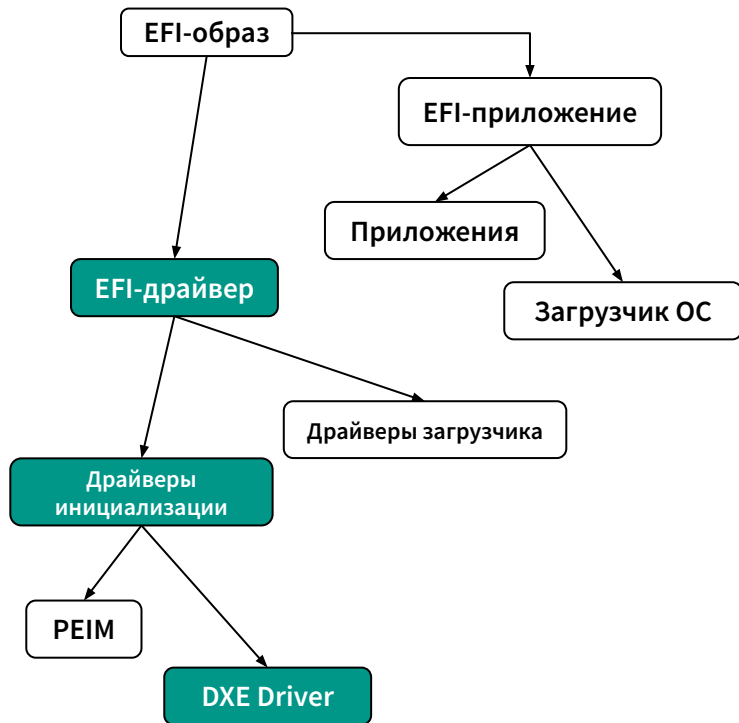
- Интерфейсы называются протоколами
- Протоколы описываются спецификацией и представляют собой таблицы
- Таблицы могут содержать данные, указатели и указатели на другие таблицы-протоколы
- EFI System Table - корневая таблица UEFI-окружения

# UEFI-протоколы



Переходы по таблицам для доступа к функции `OutputString()` протокола `SIMPLE TEXT OUTPUT`, выводящей на экран строку

# UEFI-образ



- Представляют собой исполняемые файлы формата PE/PE64 с измененным заголовком
- Драйвера поставляют интерфейс
- Обычные приложения после завершения возвращают управление
- Загрузчики ОС “закрывают” среду UEFI

# Цель

Разработать метод фаззинг-тестирования драйверов файловых систем в UEFI-окружении

# Задачи

- Фаззинг-тестирование FAT и EFI NTFS драйверов до достижения наибольшего покрытия по коду, аналогично ext4
- Устранение багов, выявляемых по ходу проведения фаззинг-тестирования

# 1. Анализ имеющихся методов

- Знакомство с фаззинг-тестированием

- Кулямин В. В. Обзор методов динамического анализа программного обеспечения. Труды ИСП РАН, том 35, вып. 4, 2023 г., стр. 7-44. DOI: 10.15514/ISPRAS-2023-35(4)-1
- A. Takanen, J. DeMott, C. Miller, A. Kettunen Fuzzing for Software Security Testing and Quality Assurance. 2-nd Edition. ISBN 13: 978-1-60807-850-9

- Анализ существующей методики фаззинг-тестирования драйверов с использованием LibFuzzer

- <https://llvm.org/docs/LibFuzzer>
- <https://github.com/acidanthera/OpenCorePkg>



# 1. Анализ имеющихся методов

Filename	Line Coverage ⇅				Branch Coverage ⇅			Function Coverage ⇅		
	Rate	Total	Hit	Rate	Total	Hit	Rate	Total	Hit	
Delete.c	<div><div></div></div>	38.9 %	36	14	11.8 %	34	4	100.0 %	1	1
DirectoryCache.c	<div><div></div></div>	73.6 %	53	39	32.4 %	34	11	100.0 %	5	5
DirectoryManage.c	<div><div></div></div>	38.7 %	460	178	32.2 %	230	74	42.3 %	26	11
DiskCache.c	<div><div></div></div>	68.8 %	141	97	45.3 %	64	29	85.7 %	7	6
FileName.c	<div><div></div></div>	43.5 %	147	64	59.2 %	98	58	30.0 %	10	3
FileSpace.c	<div><div></div></div>	29.5 %	241	71	23.8 %	143	34	41.7 %	12	5
Flush.c	<div><div></div></div>	54.6 %	119	65	36.2 %	94	34	88.9 %	9	8
Hash.c	<div><div></div></div>	44.4 %	36	16	25.0 %	8	2	66.7 %	6	4
Info.c	<div><div></div></div>	43.0 %	149	64	24.5 %	102	25	66.7 %	9	6
Init.c	<div><div></div></div>	81.7 %	131	107	73.9 %	92	68	66.7 %	3	2
Misc.c	<div><div></div></div>	25.6 %	176	45	12.0 %	92	11	41.2 %	17	7
Open.c	<div><div></div></div>	61.8 %	89	55	35.6 %	59	21	100.0 %	4	4
OpenVolume.c	<div><div></div></div>	84.6 %	13	11	50.0 %	8	4	100.0 %	1	1
ReadWrite.c	<div><div></div></div>	48.1 %	162	78	36.8 %	106	39	58.3 %	12	7
UnicodeCollation.c	<div><div></div></div>	17.5 %	57	10	0.0 %	14		28.6 %	7	2

Результат фаззинг-тестирования для драйвера FAT

# 1. Анализ имеющихся методов

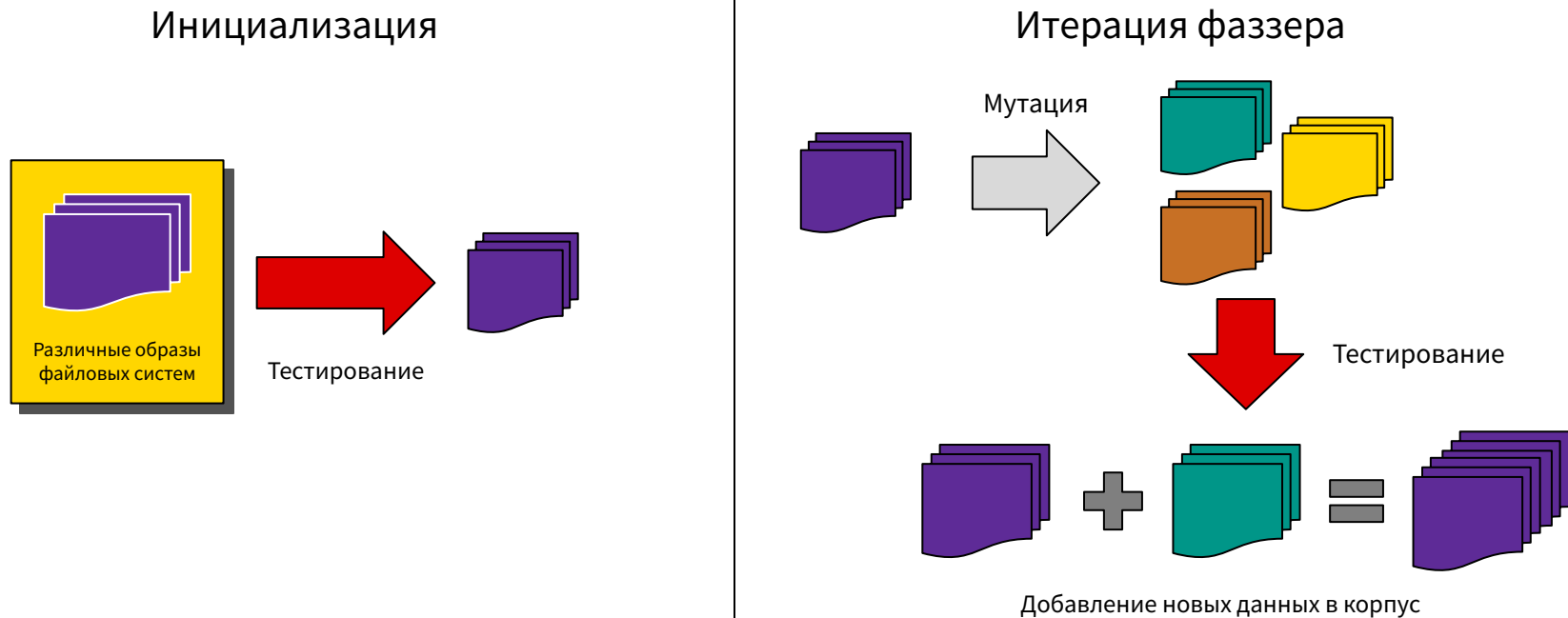
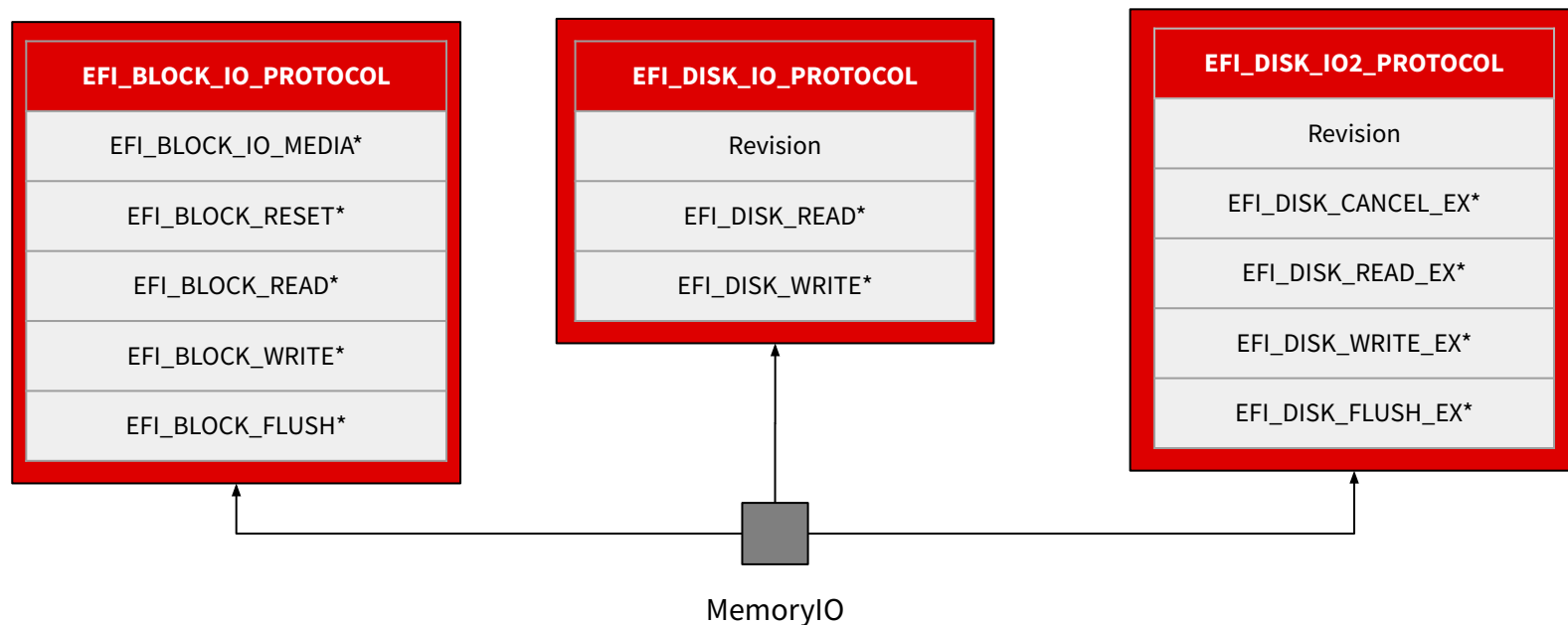


Схема работы фаззера с libFuzzer

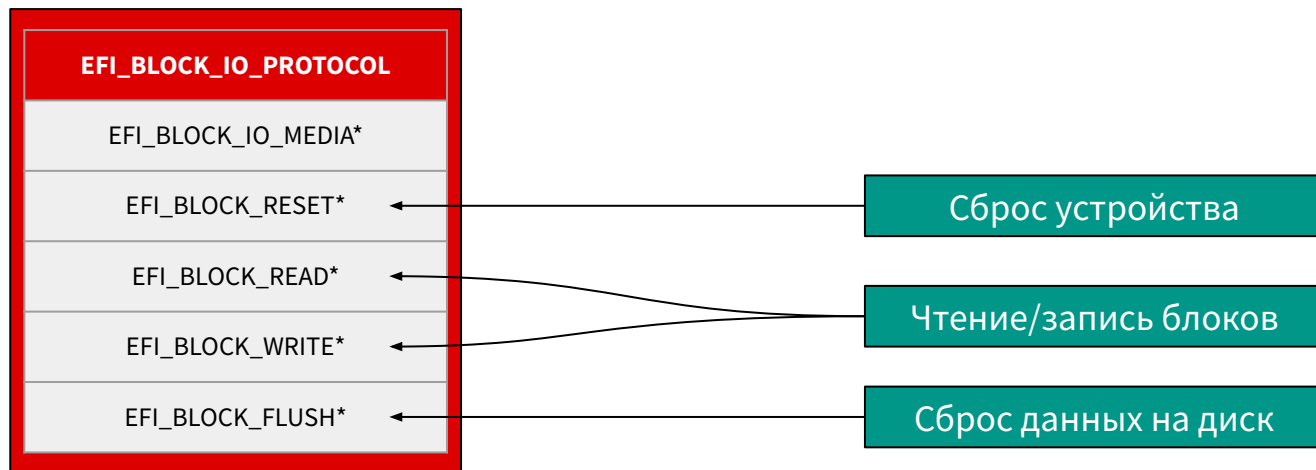
## 2. Разработка метода



Эмуляция устройства, поставляющего протоколы для работы с диском

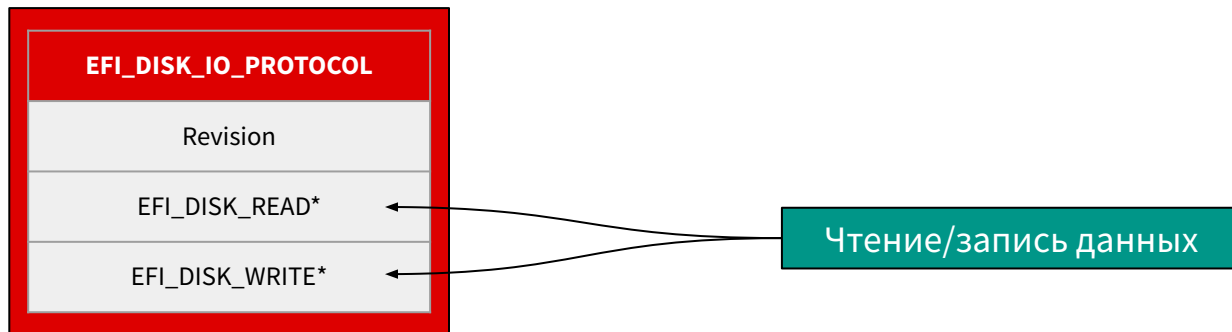
## 2. Разработка метода

Функции для работы с диском как с блочным устройством



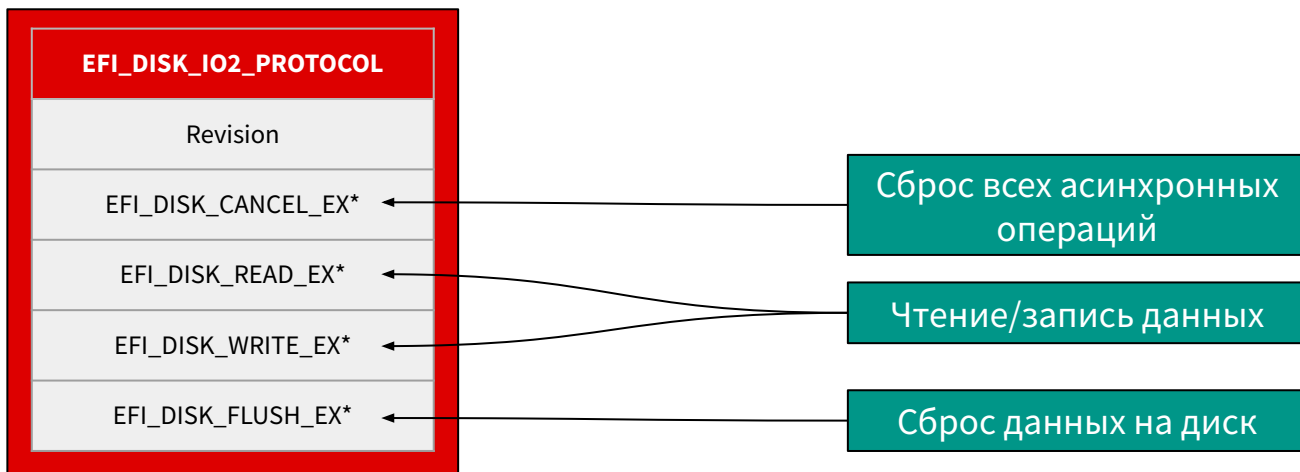
## 2. Разработка метода

Функции для работы с диском как с файловым потоком



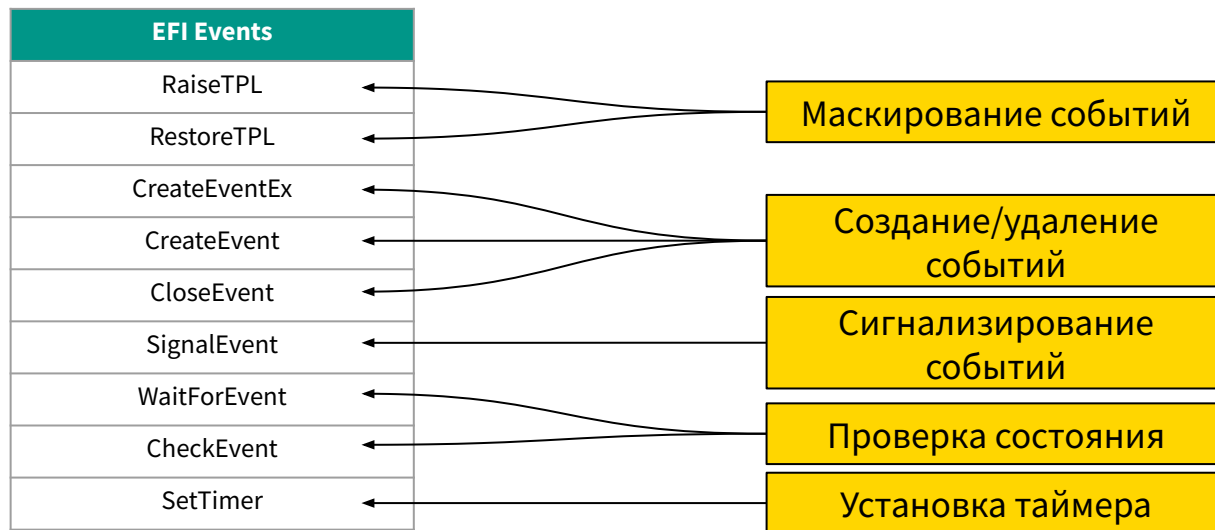
## 2. Разработка метода

Асинхронные функции для работы с диском как с файловым потоком



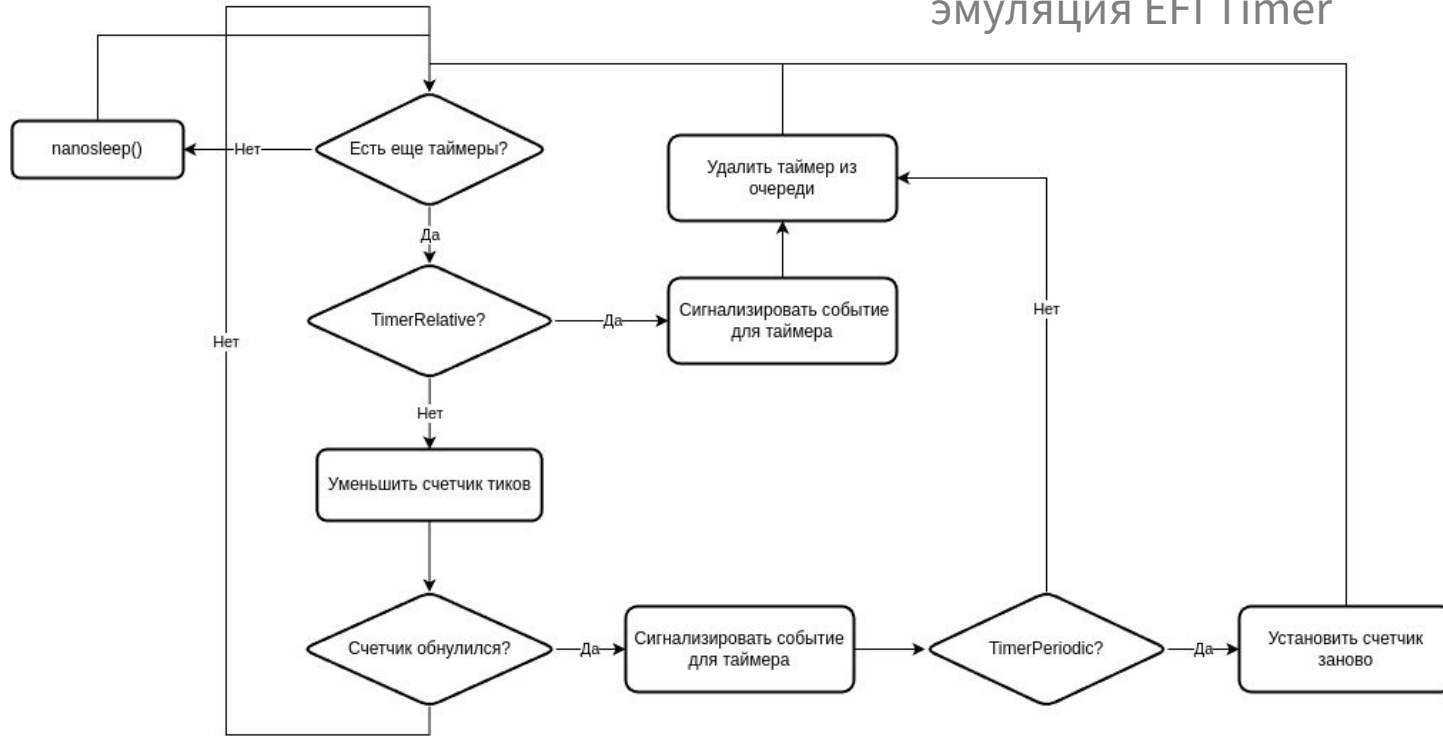
## 2. Разработка метода

Эмулируемые функции для работы с событиями в EFI



## 2. Разработка метода

В отдельном потоке происходит эмуляция EFI Timer





## 2. Разработка метода

EFI_FILE_PROTOCOL
EFI_FILE_OPEN*
EFI_FILE_CLOSE*
EFI_FILE_DELETE*
EFI_FILE_READ*
EFI_FILE_WRITE*
EFI_FILE_GET_POSITION*
EFI_FILE_SET_POSITION*
EFI_FILE_FLUSH*
EFI_FILE_GET_INFO*
EFI_FILE_SET_INFO*
EFI_FILE_OPEN_EX*
EFI_FILE_READ_EX*
EFI_FILE_WRITE_EX*
EFI_FILE_FLUSH_EX*

Протокол для доступа к файловой системе

EFI_FILE_INFO
SizeFileInfo
FileSize
FilePhysicalSize
CreateTime
LastAccessTime
ModificationTime
Attribute
FileName[]

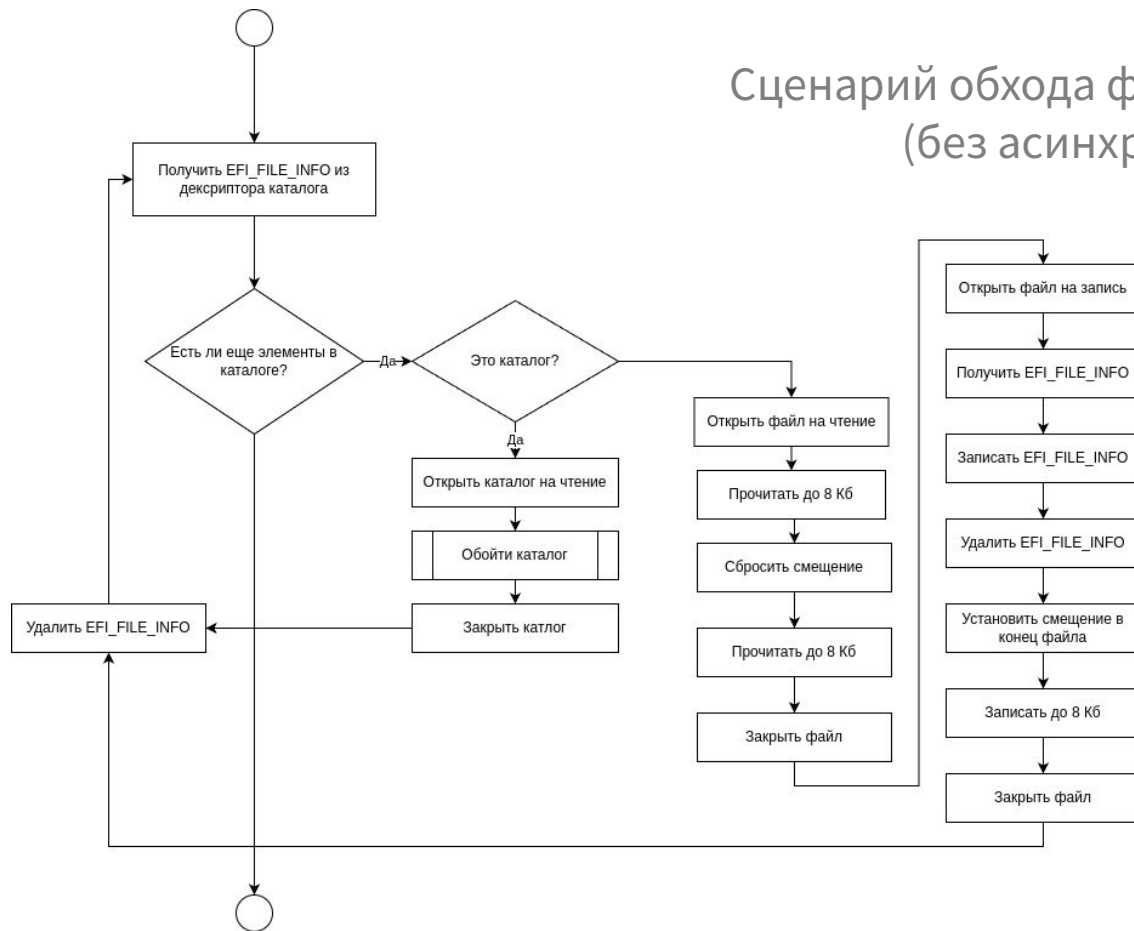
EFI_FILE_SYSTEM_INFO
SizeFileSystemInfo
ReadOnly
VolumeSize
FreeSpace
BlockSize
VolumeLabel[]

Протоколы для получения информации об  
объектах файловой системы

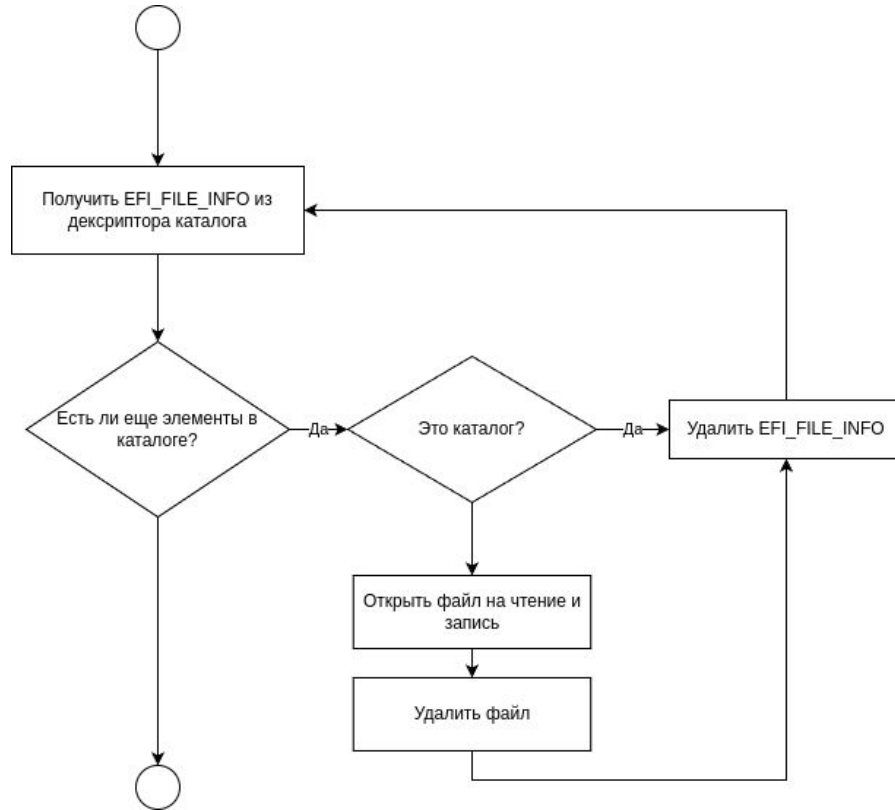
Интерфейсы, которые предоставляет  
драйвер файловой системы

## 2. Разработка метода

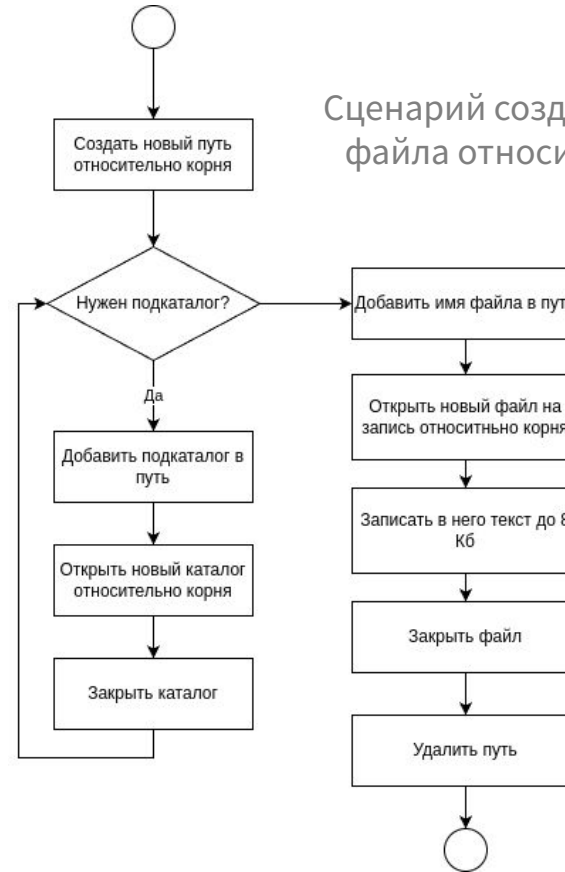
Сценарий обхода файловой системы  
(без асинхронности)



## 2. Разработка метода



Сценарий очистки каталога от файлов


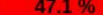




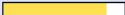








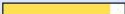


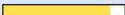


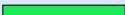














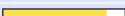










Сценарий создания тестового файла относительно корня

### 3. Фаззинг-тестирование

- Тестирование имеющихся открытых драйверов
- Сравнительный анализ
- Устранение выявленных багов в драйверах FAT и EFI NTFS

### 3. Фаззинг-тестирование

Filename	Line Coverage ↕			Branch Coverage ↕			Function Coverage ↕		
	Rate	Total	Hit	Rate	Total	Hit	Rate	Total	Hit
Delete.c	 77.8 %	36	28	 47.1 %	34	16	 100.0 %	1	1
DirectoryCache.c	 100.0 %	53	53	 70.6 %	34	24	 100.0 %	5	5
DirectoryManage.c	 85.0 %	460	391	 72.6 %	230	167	 88.5 %	26	23
DiskCache.c	 78.7 %	141	111	 76.6 %	64	49	 85.7 %	7	6
FileName.c	 79.6 %	147	117	 82.7 %	98	81	 90.0 %	10	9
FileSpace.c	 88.4 %	241	213	 82.5 %	143	118	 100.0 %	12	12
Flush.c	 88.2 %	119	105	 73.4 %	94	69	 100.0 %	9	9
Hash.c	 100.0 %	36	36	 100.0 %	8	8	 100.0 %	6	6
Info.c	 38.3 %	149	57	 31.4 %	102	32	 55.6 %	9	5
Init.c	 75.6 %	131	99	 57.6 %	92	53	 66.7 %	3	2
Misc.c	 47.2 %	176	83	 44.6 %	92	41	 64.7 %	17	11
Open.c	 78.7 %	89	70	 62.7 %	59	37	 100.0 %	4	4
OpenVolume.c	 84.6 %	13	11	 50.0 %	8	4	 100.0 %	1	1
ReadWrite.c	 82.7 %	162	134	 66.0 %	106	70	 83.3 %	12	10
UnicodeCollation.c	 47.4 %	57	27	 0.0 %	14		 71.4 %	7	5

Результат фаззинг-тестирования для драйвера FAT (без асинхронных функций)

### 3. Фаззинг-тестирование

№	Тип	Количество
1	Undefined Behavior	2
2	Memory Leak	1

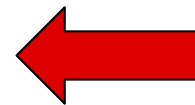
Ошибки, обнаруженные в результате тестирования FAT

№	Тип	Количество
1	Deadlock (?)	1

Ошибки, обнаруженные в результате тестирования FAT при попытке  
включить асинхронные операции

## 4. Текущее состояние проекта

- ~~Добавить поддержку асинхронных операций~~
- Включить асинхронные операции в процесс тестирования
- Уменьшить фрагментирование памяти фаззером
- Фаззинг-тестирование ext4 и сравнительный анализ с имеющимся методом.  
Улучшение сценариев, если будет необходимо
- Фаззинг-тестирование FAT и исправление ошибок
- Фаззинг-тестирование NTFS и исправление ошибок



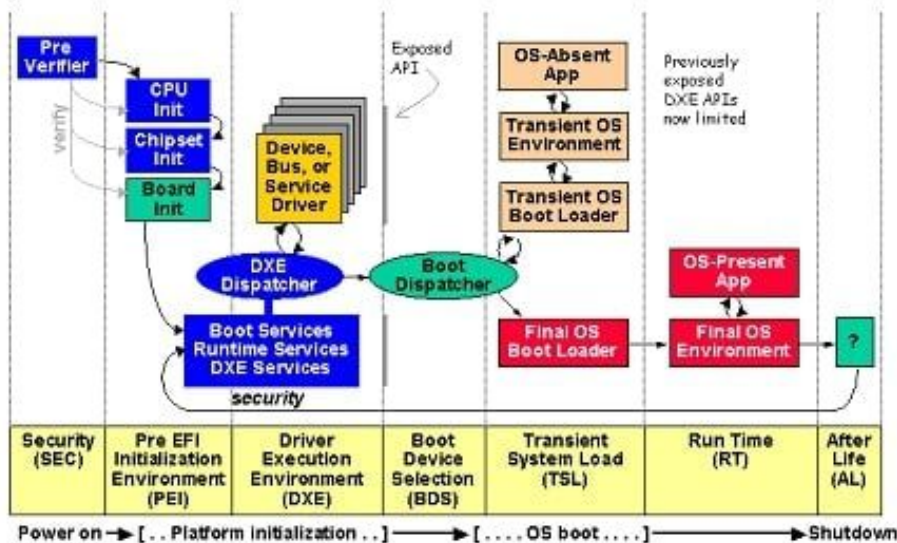
# План-график

Дата	Задача
18.10 - 31.12	Анализ имеющихся методов
<del>01.01 - 31.01</del> 01.03 - 31.03	Разработка метода
<del>01.02 - 14.02</del> 01.04 - 31.05	Фаззинг-тестирование и устранение выявленных багов
<del>15.02 - 31.03</del>	<del>Устранение выявленных багов</del>



**Спасибо за  
внимание**

# Фазы UEFI



Источник: UEFI PI Specification v 1.9  
<https://uefi.org/specifications>

