



Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский государственный технический университет имени  
Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)

---

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

## Отчет по лабораторной работе по дисциплине «Операционные системы»

Тема Системный вызов open

Студент Ляпина Н.В.

Группа ИУ7-62Б

Оценка (баллы) \_\_\_\_\_

Преподаватель Рязанова Н.Ю.

Москва — 2023 г.

# Описание структур

Листинг 1: Структура struct filename

```
1 struct filename {
2     const char    *name; /* pointer to actual string */
3     const __user char *uptr; /* original userland pointer */
4     int          refcnt;
5     struct audit_names *aname;
6     const char    iname[];
7 };
```

Листинг 2: Структура struct open\_flags

```
1 struct open_flags {
2     int open_flag;
3     umode_t mode;
4     int acc_mode;
5     int intent;
6     int lookup_flags;
7 };
```

Листинг 3: Структура struct nameidata

```
1 #define EMBEDDED_LEVELS 2
2 struct nameidata {
3     struct path path;
4     struct qstr last;
5     struct path root;
6     struct inode *inode; /* path.dentry.d_inode */
7     unsigned int flags, state;
8     unsigned seq, next_seq, m_seq, r_seq;
9     int last_type;
10    unsigned depth;
11    int total_link_count;
12    struct saved {
13        struct path link;
14        struct delayed_call done;
15        const char *name;
16        unsigned seq;
17    } *stack, internal[EMBEDDED_LEVELS];
18    struct filename *name;
19    struct nameidata *saved;
20    unsigned root_seq;
21    int dfd;
22    vfsuid_t dir_vfsuid;
23    umode_t dir_mode;
24 } __randomize_layout;
```

## Флаги системного вызова `open()`

`O_EXEC` — открыть только для выполнения (результат не определен, при открытии директории).

`O_RDONLY` — открыть только на чтение.

`O_RDWR` — открыть на чтение и запись.

`O_SEARCH` — открыть директорию только для поиска (результат не определен, при использовании с файлами, не являющимися директорией).

`O_WRONLY` — открыть только на запись.

`O_APPEND` — файл открывается в режиме добавления, перед каждой операцией записи файловый указатель будет устанавливаться в конец файла.

`O_CLOEXEC` — включает флаг `close-on-exec` для нового файлового дескриптора, указание этого флага позволяет программе избегать дополнительных операций `fcntl F_SETFD` для установки флага `FD_CLOEXEC`.

`O_CREAT` — если файл не существует, то он будет создан.

`O_DIRECTORY` — если файл не является каталогом, то `open` вернёт ошибку.

`O_DSYNC` — файл открывается в режиме синхронного ввода-вывода (все операции записи для соответствующего дескриптора файла блокируют вызывающий процесс до тех пор, пока данные не будут физически записаны).

`O_EXCL` — если используется совместно с `O_CREAT`, то при наличии уже созданного файла вызов завершится ошибкой.

`O_NOCTTY` — если файл указывает на терминальное устройство, то оно не станет терминалом управления процесса, даже при его отсутствии.

`O_NOFOLLOW` — если файл является символической ссылкой, то `open` вернёт ошибку.

`O_NONBLOCK` — файл открывается, по возможности, в режиме `non-blocking`, то есть никакие последующие операции над дескриптором файла не заставляют в дальнейшем вызывающий процесс ждать.

`O_RSYNC` — операции записи должны выполняться на том же уровне, что и `O_SYNC`.

`O_SYNC` — файл открывается в режиме синхронного ввода-вывода (все операции записи для соответствующего дескриптора файла блокируют вызывающий процесс до тех пор, пока данные не будут физически записаны).

`O_TRUNC` — если файл уже существует, он является обычным файлом и заданный режим позволяет записывать в этот файл, то его длина будет урезана до нуля.

`O_LARGEFILE` — позволяет открывать файлы, размер которых не может быть представлен типом `off_t` (`long`).

`O_TMPFILE` — при наличии данного флага создаётся неименованный временный файл.

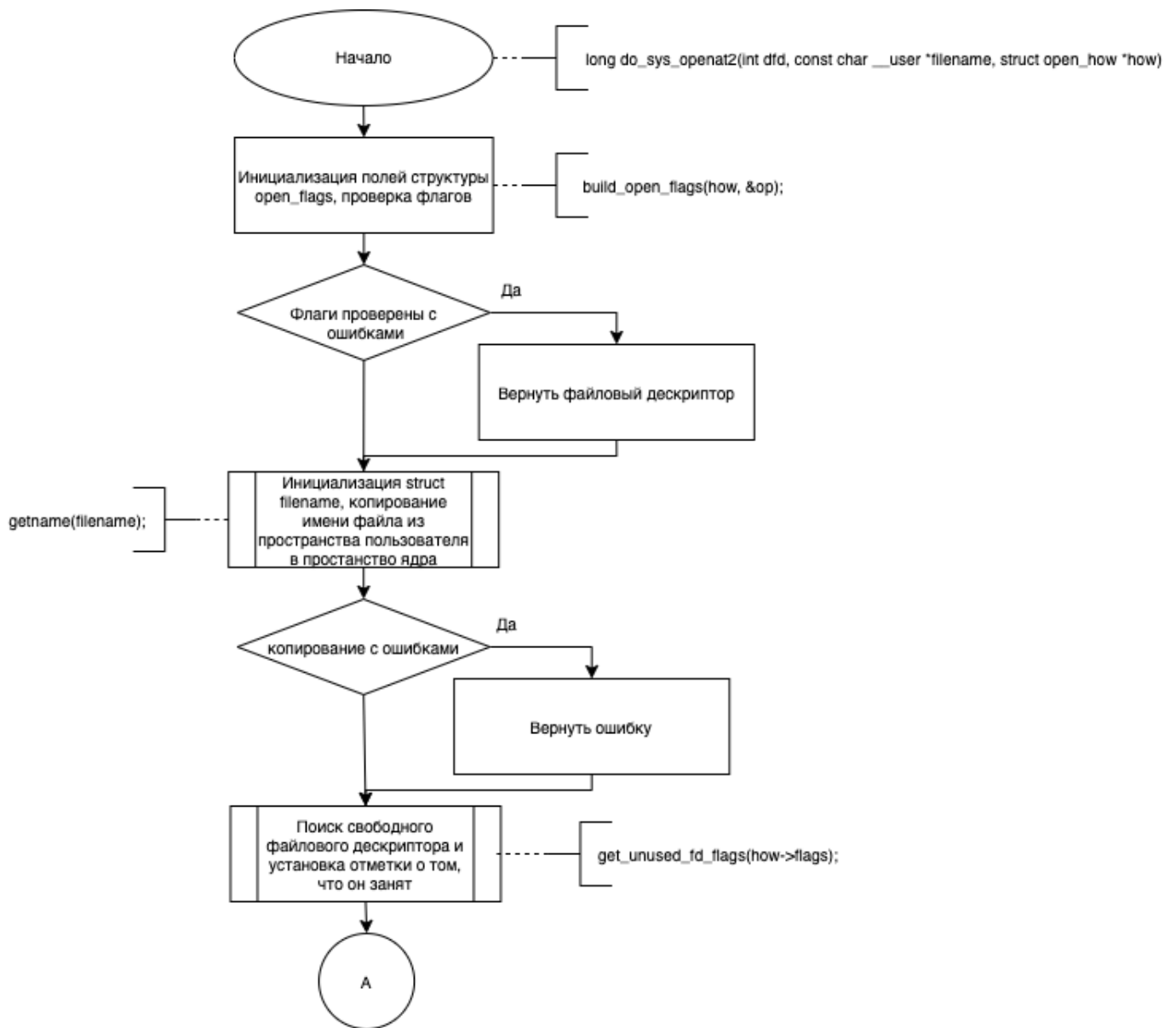


Рис. 1: Схема алгоритма работы системного вызова do\_sys\_open2. Часть 1

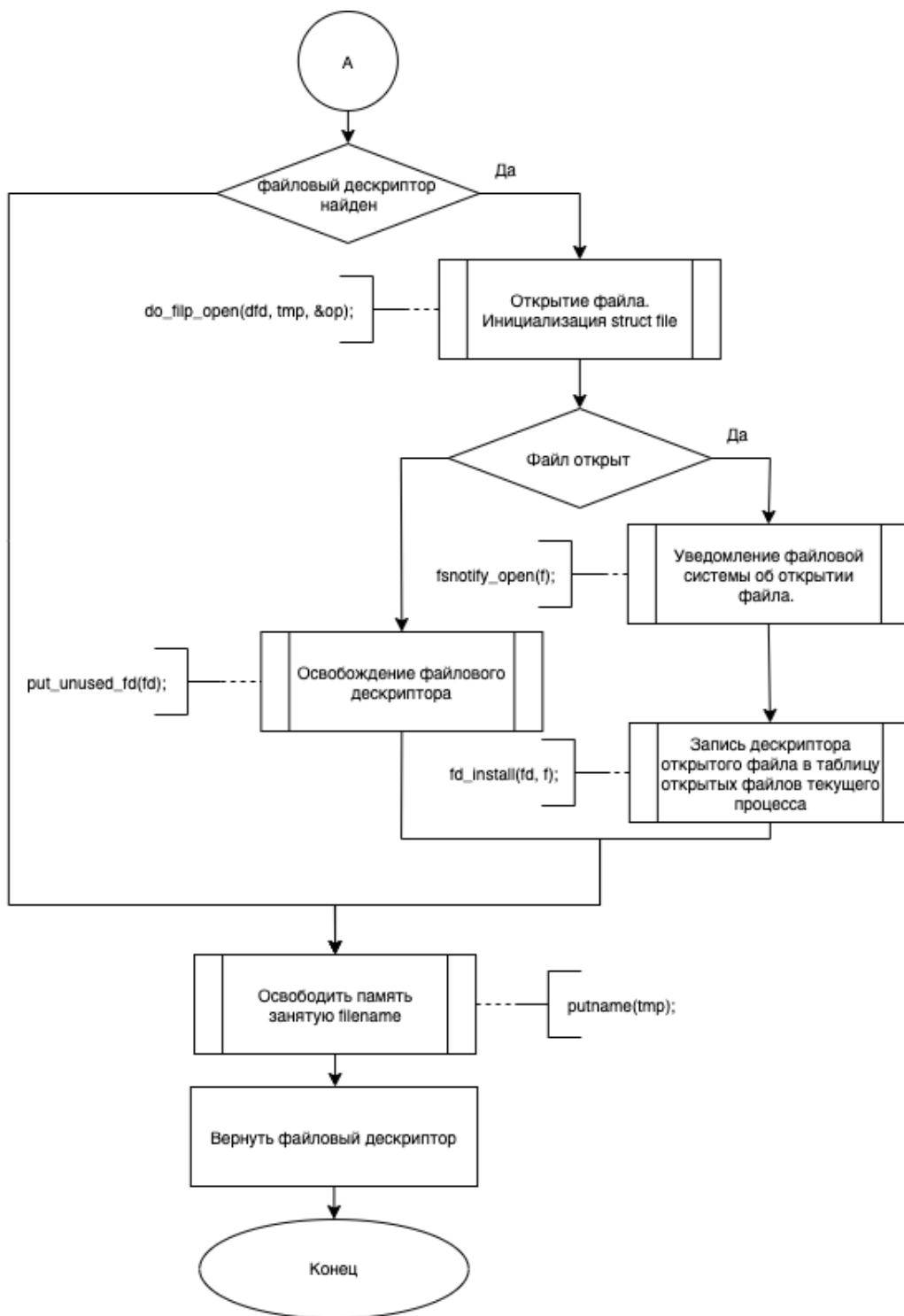


Рис. 2: Схема алгоритма работы системного вызова `do_sys_open2`. Часть 2

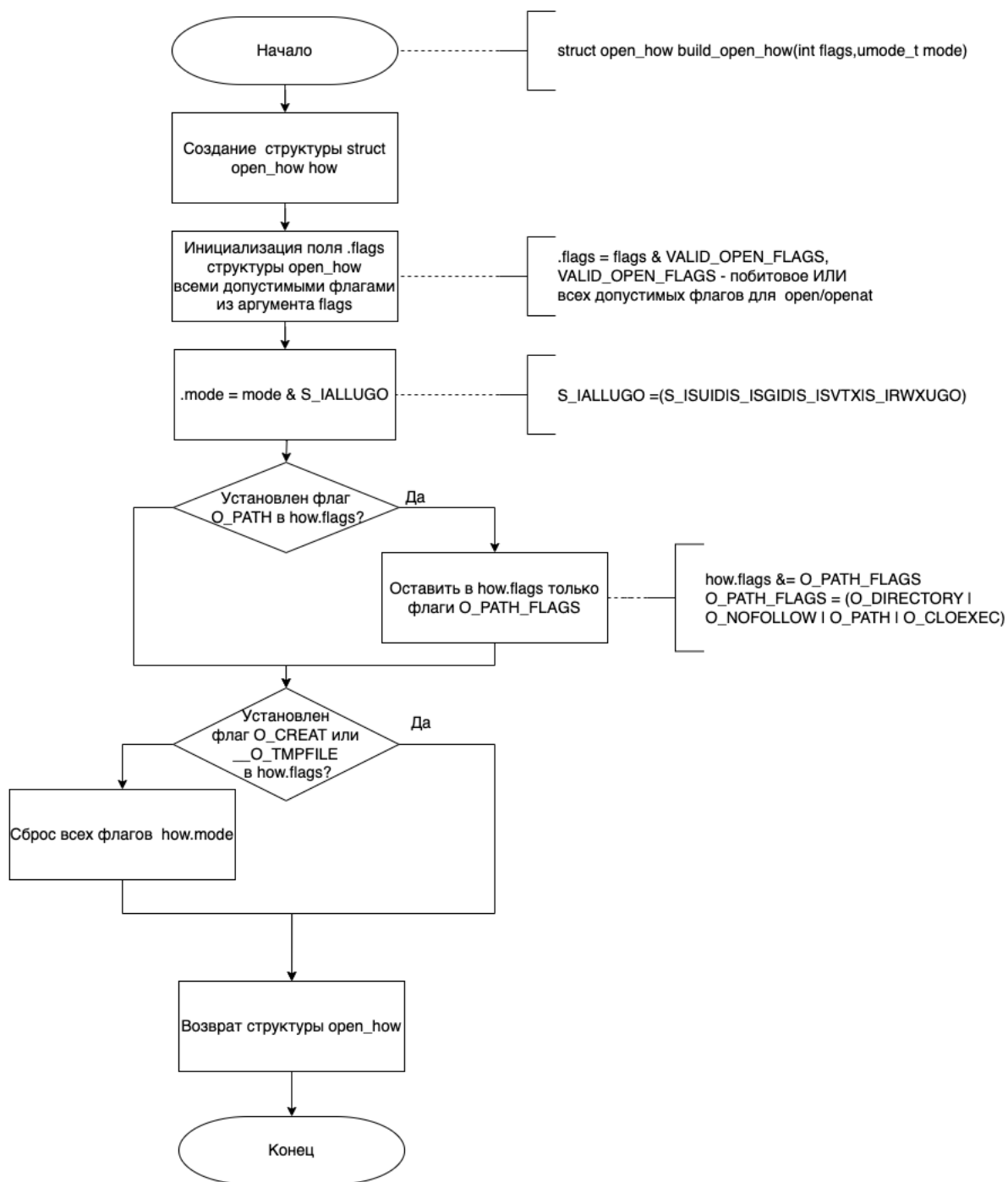


Рис. 3: Схема алгоритма работы системного вызова build\_open\_how

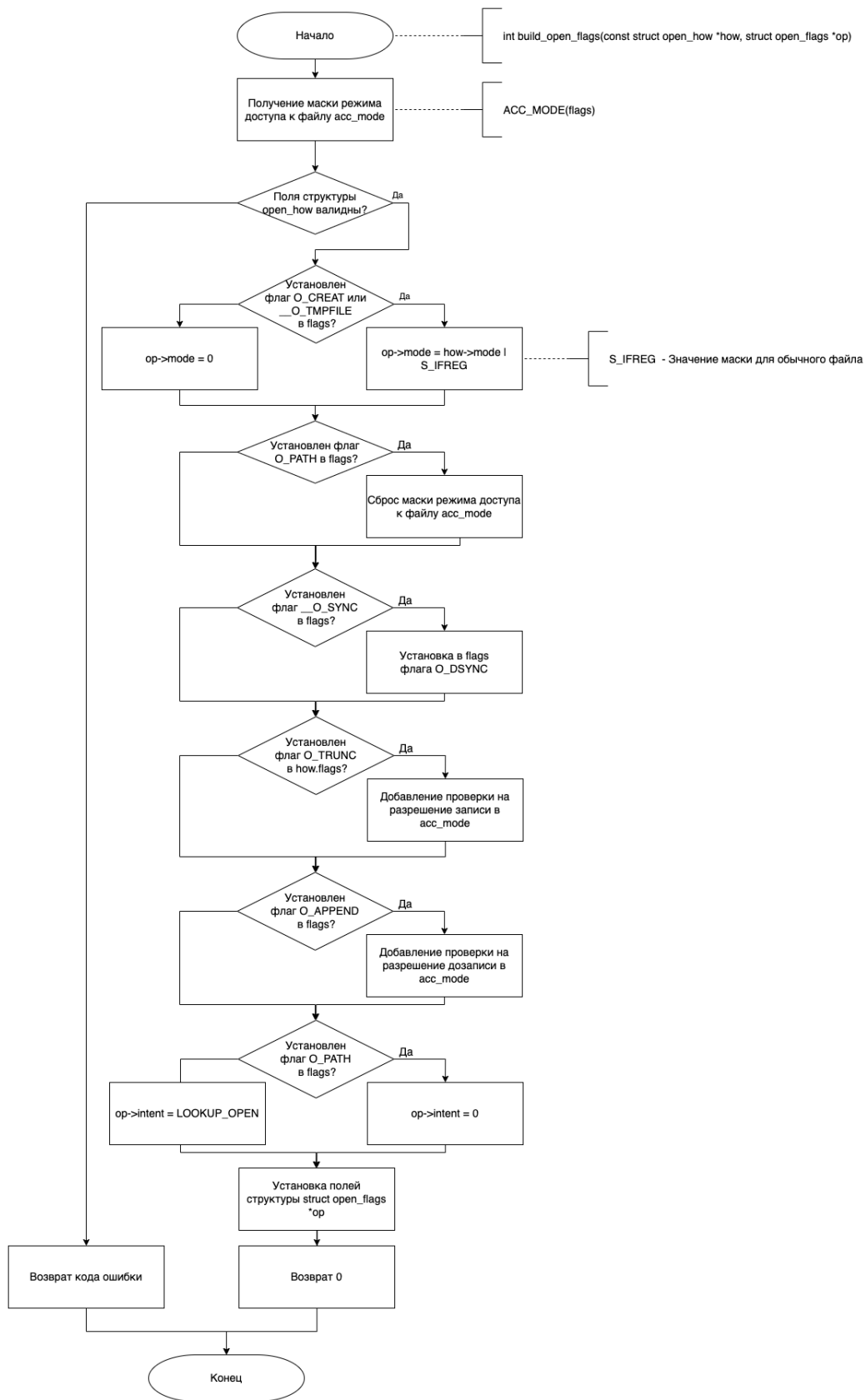


Рис. 4: Схема алгоритма работы системного вызова build\_open\_flags

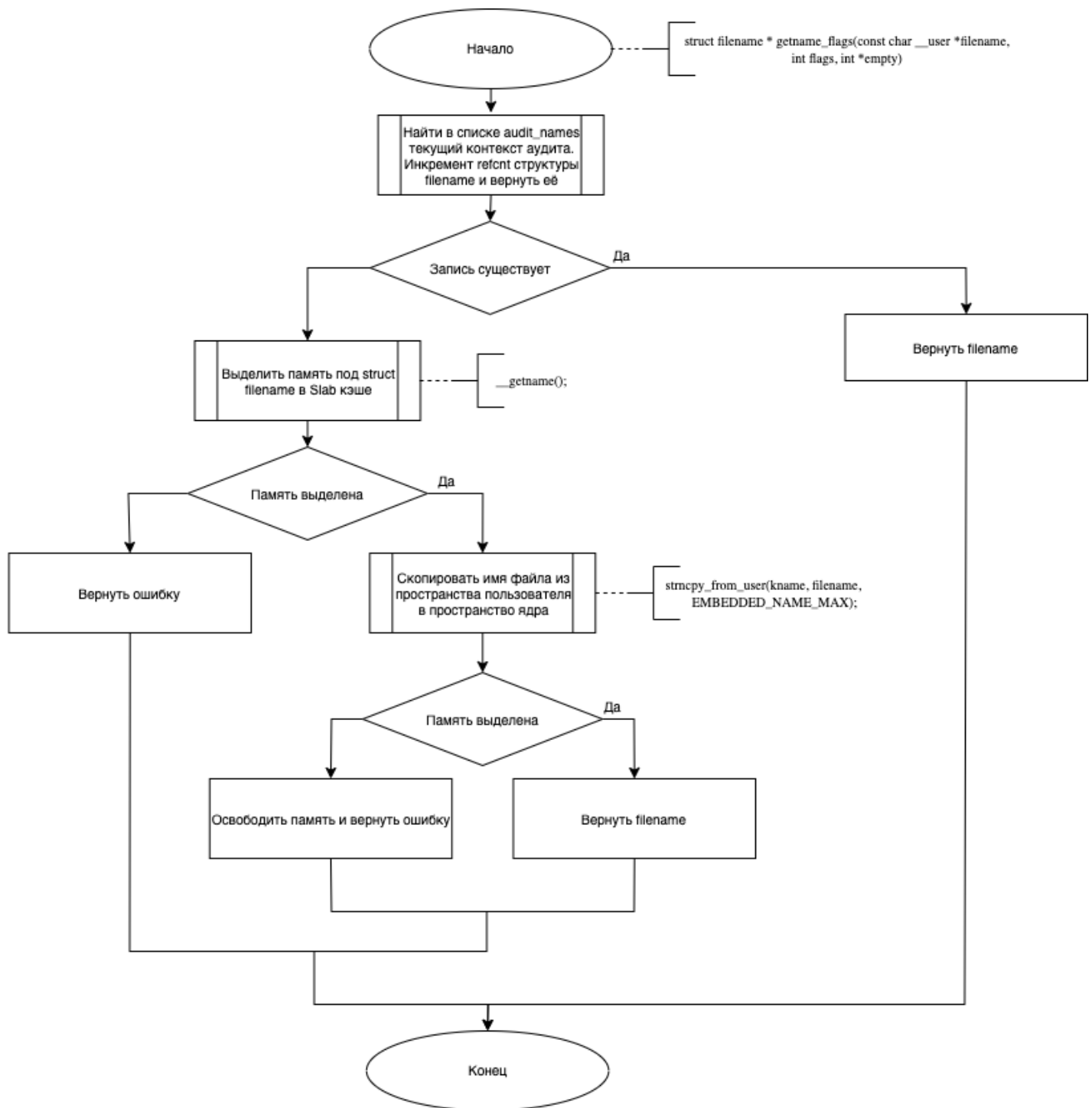


Рис. 5: Схема алгоритма работы системного вызова `getname_flags`



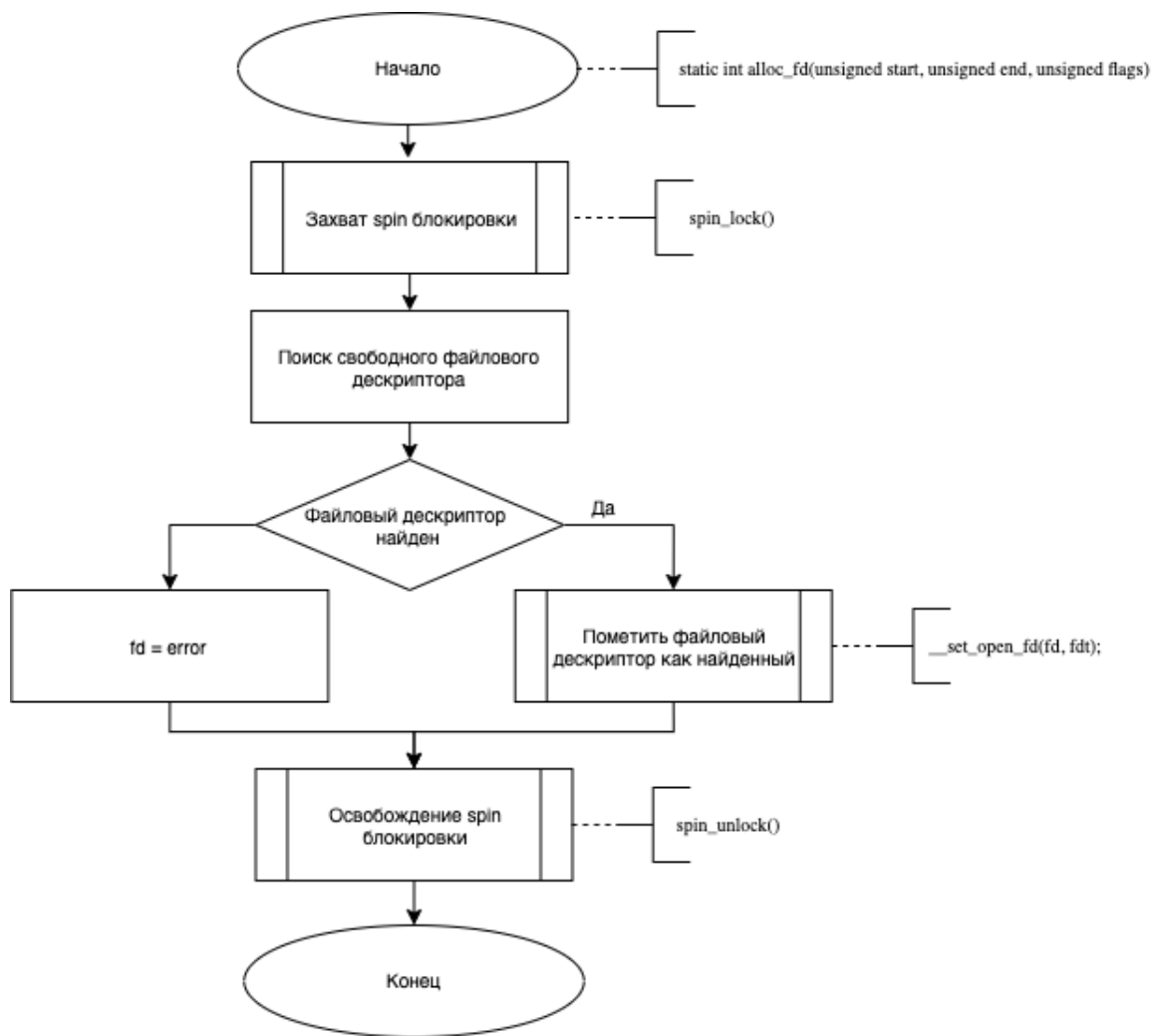


Рис. 6: Схема алгоритма работы системного вызова `alloc_fd`

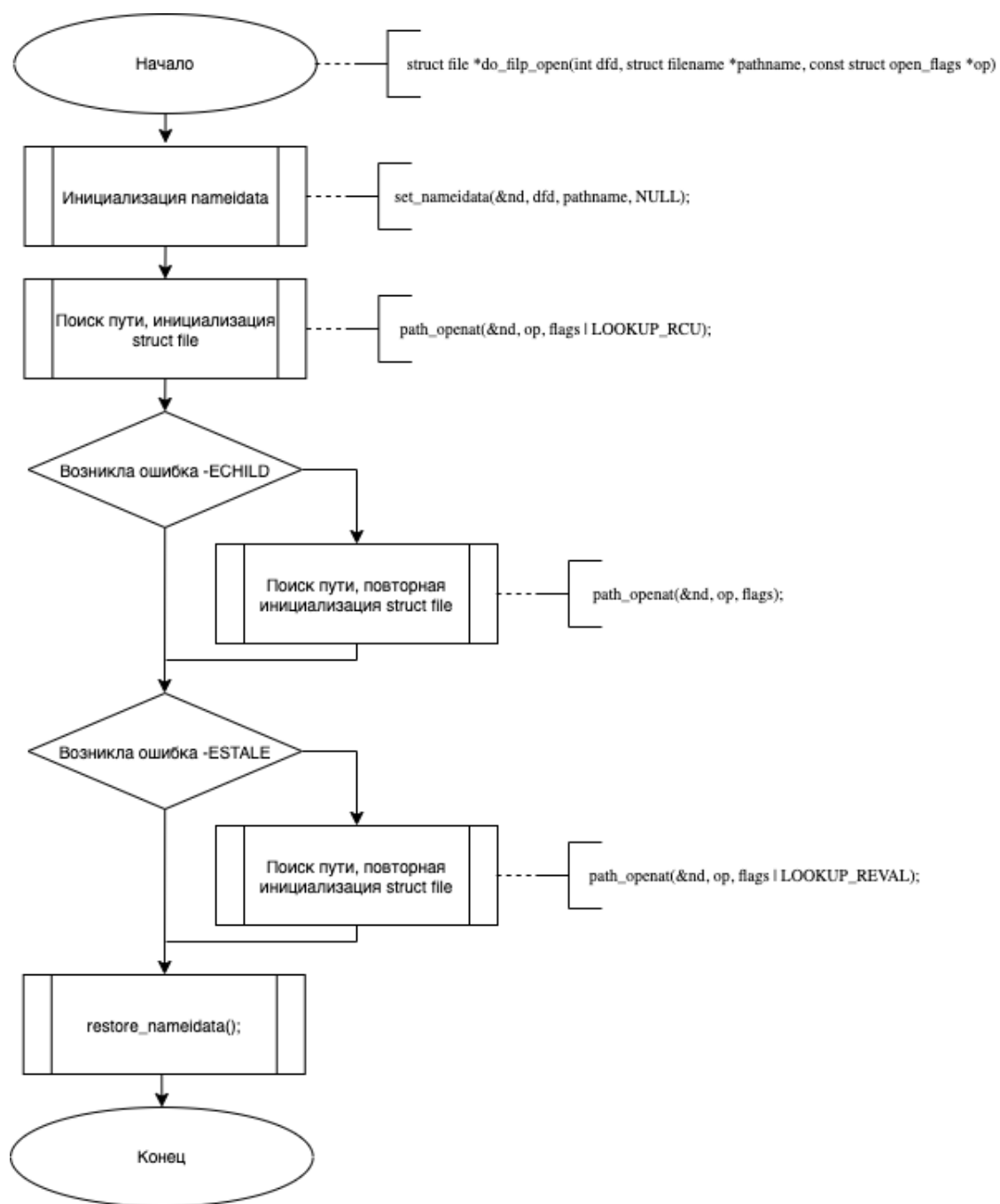


Рис. 7: Схема алгоритма работы системного вызова do\_filp\_open

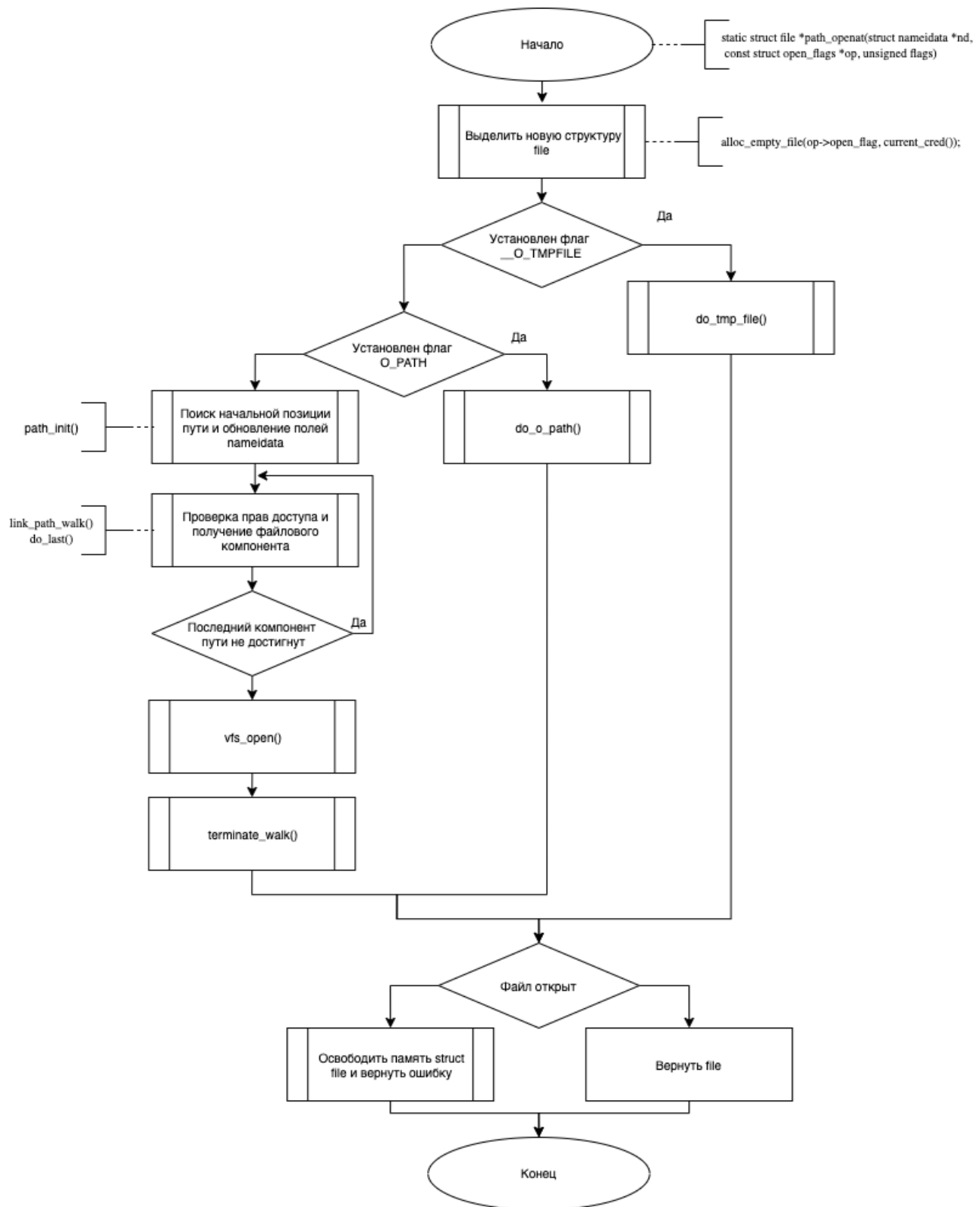


Рис. 8: Схема алгоритма работы системного вызова path\_openat

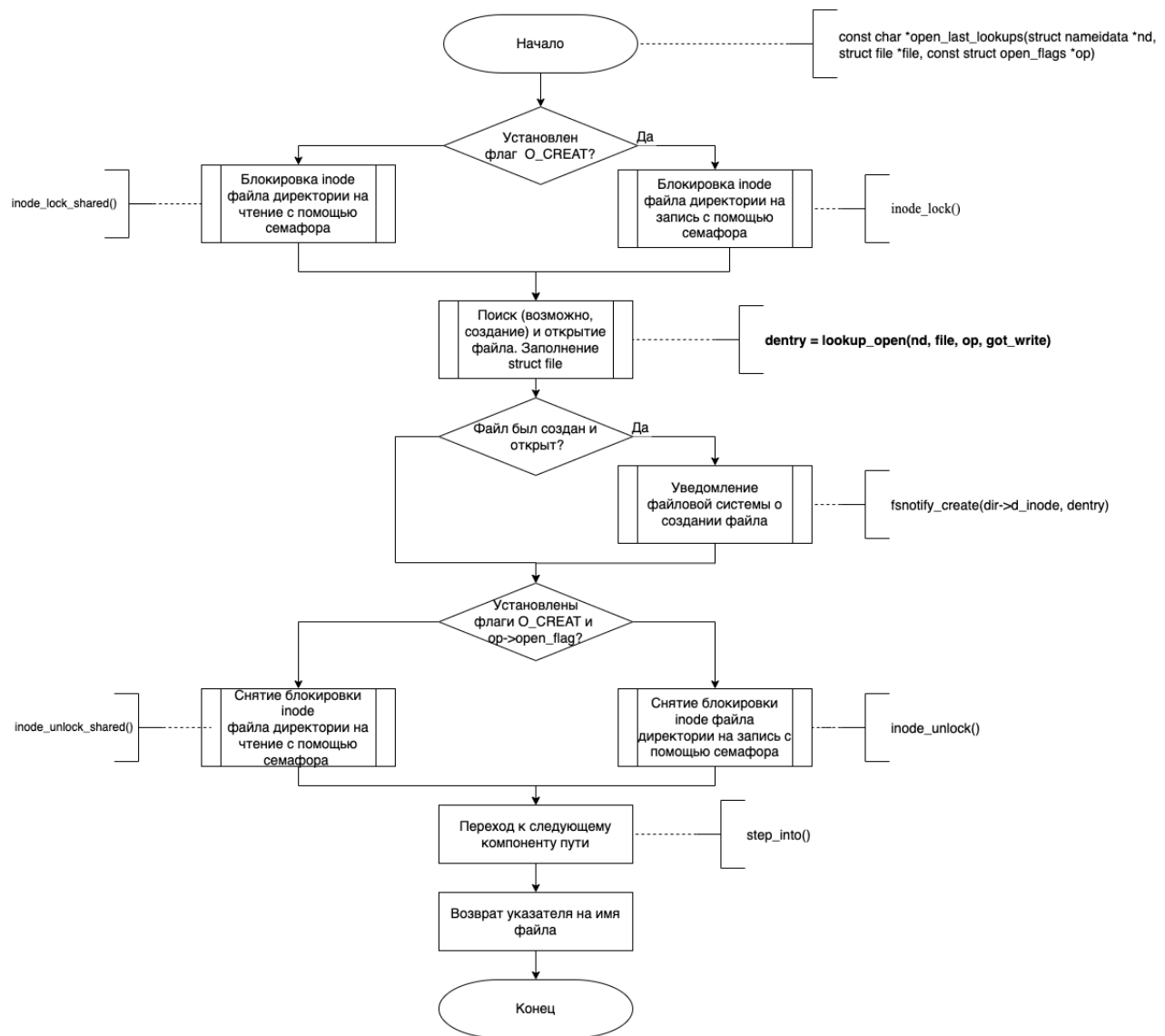


Рис. 9: Схема алгоритма работы системного вызова open\_last\_lookups

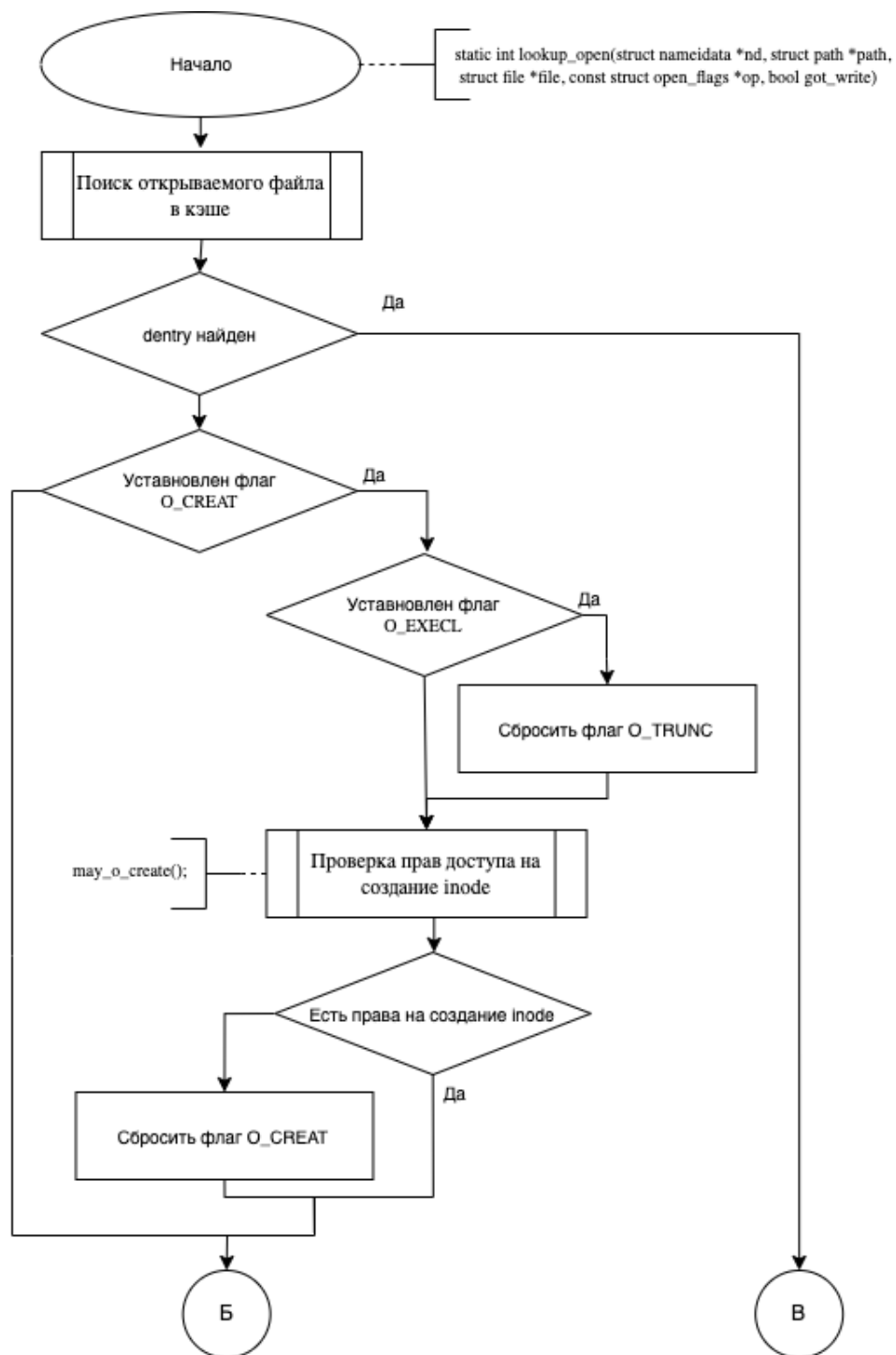


Рис. 10: Схема алгоритма работы системного вызова lookup\_open. Часть 1

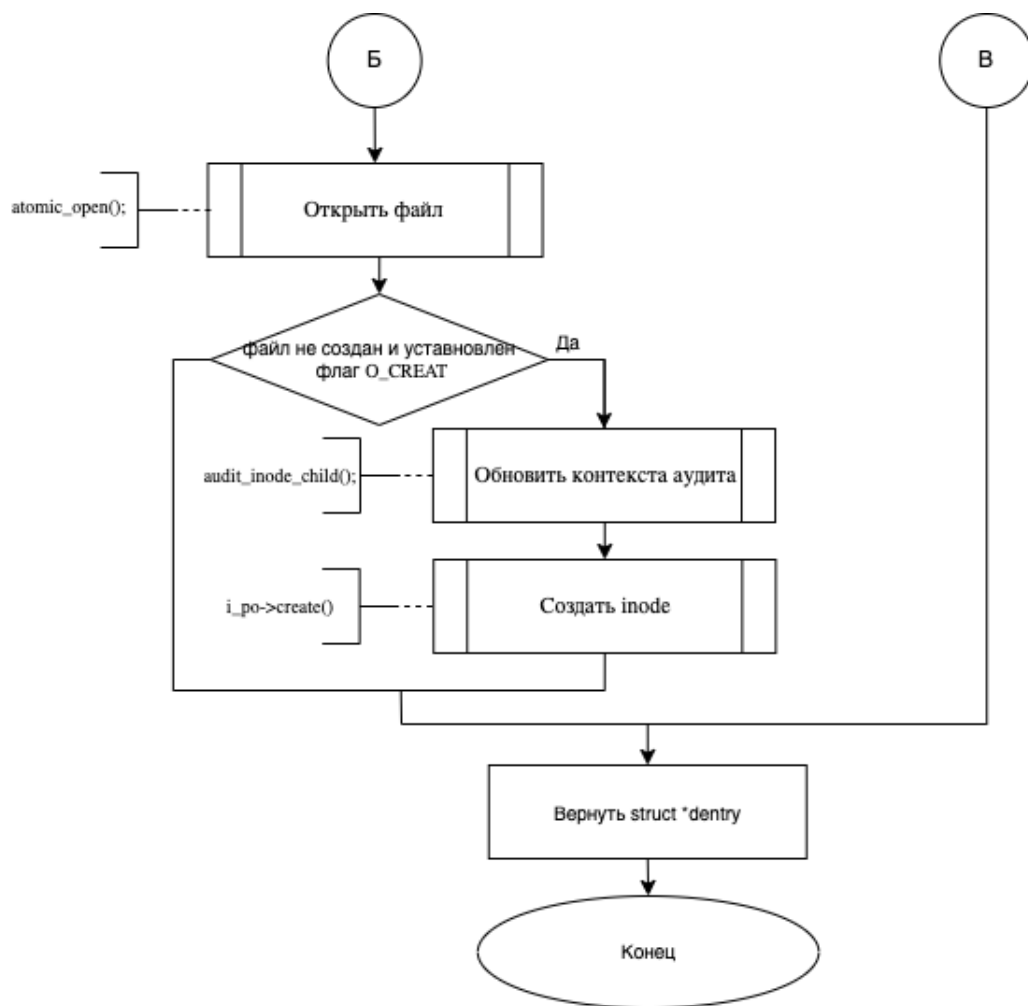


Рис. 11: Схема алгоритма работы системного вызова `lookup_open`. Часть 2

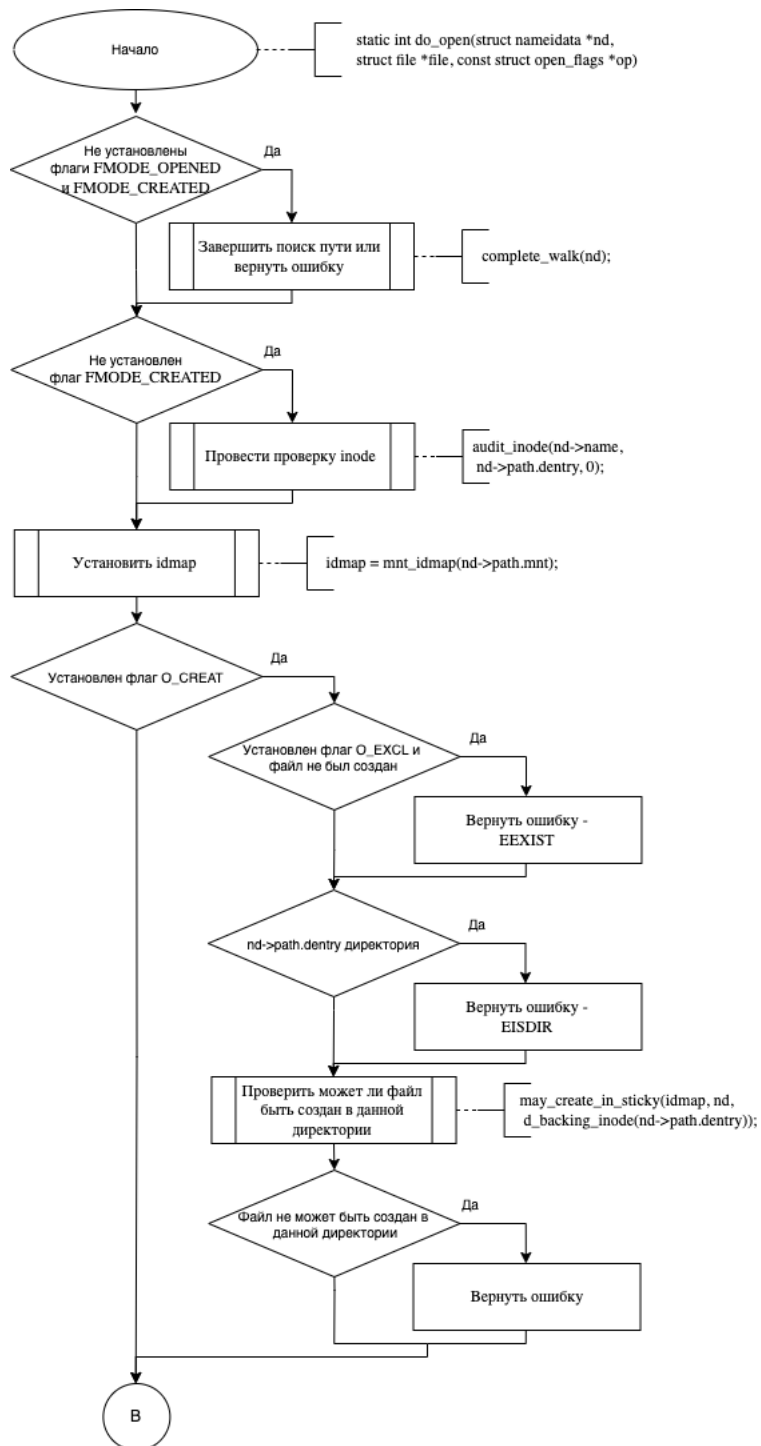


Рис. 12: Схема алгоритма работы системного вызова do\_open. Часть 1

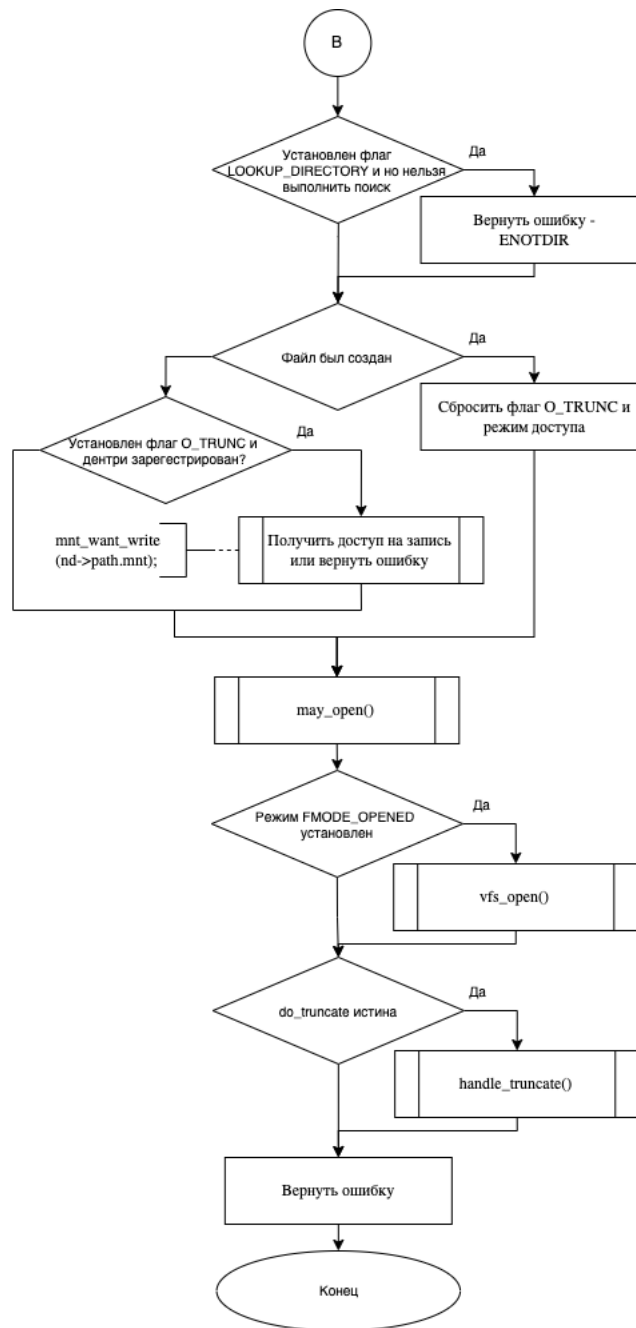


Рис. 13: Схема алгоритма работы системного вызова do\_open. Часть 2