



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 № 6
по курсу «Операционные системы»
на тему: «Системный вызов open»

Студент ИУ7-61Б
(Группа)

(Подпись, дата)

Мицевич М. Д.
(И. О. Фамилия)

Преподаватель

(Подпись, дата)

Рязанова Н. Ю.
(И. О. Фамилия)

2022 г.

1 Используемые структуры

В данной лабораторной работе в качестве операционной системы использовалась Ubuntu Linux 20.04

Листинг 1.1 – Структура filename

```
struct filename {
    const char          *name;  /* pointer to actual
        string */
    const __user char    *uptr;  /* original userland
        pointer */
    int                  refcnt;
    struct audit_names    *aname;
    const char           inode[];
};
```

Листинг 1.2 – open_flags

```
struct open_flags {
    int open_flag;
    umode_t mode;
    int acc_mode;
    int intent;
    int lookup_flags;
};
```

Листинг 1.3 – nameidata

```
struct nameidata {
    struct path          path;
    struct qstr          last;
    struct path          root;
    struct inode          *inode; /* path.dentry.d_inode */
    unsigned int         flags, state;
    unsigned             seq, m_seq, r_seq;
    int                  last_type;
    unsigned             depth;
    int                  total_link_count;
    struct saved {
        struct path link;
        struct delayed_call done;
        const char *name;
        unsigned seq;
    };
};
```

```
    } *stack, internal[EMBEDDED_LEVELS];  
    struct filename *name;  
    struct nameidata *saved;  
    unsigned        root_seq;  
    int             dfd;  
    kuid_t          dir_uid;  
    umode_t          dir_mode;  
} __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.

2 Схемы алгоритмов

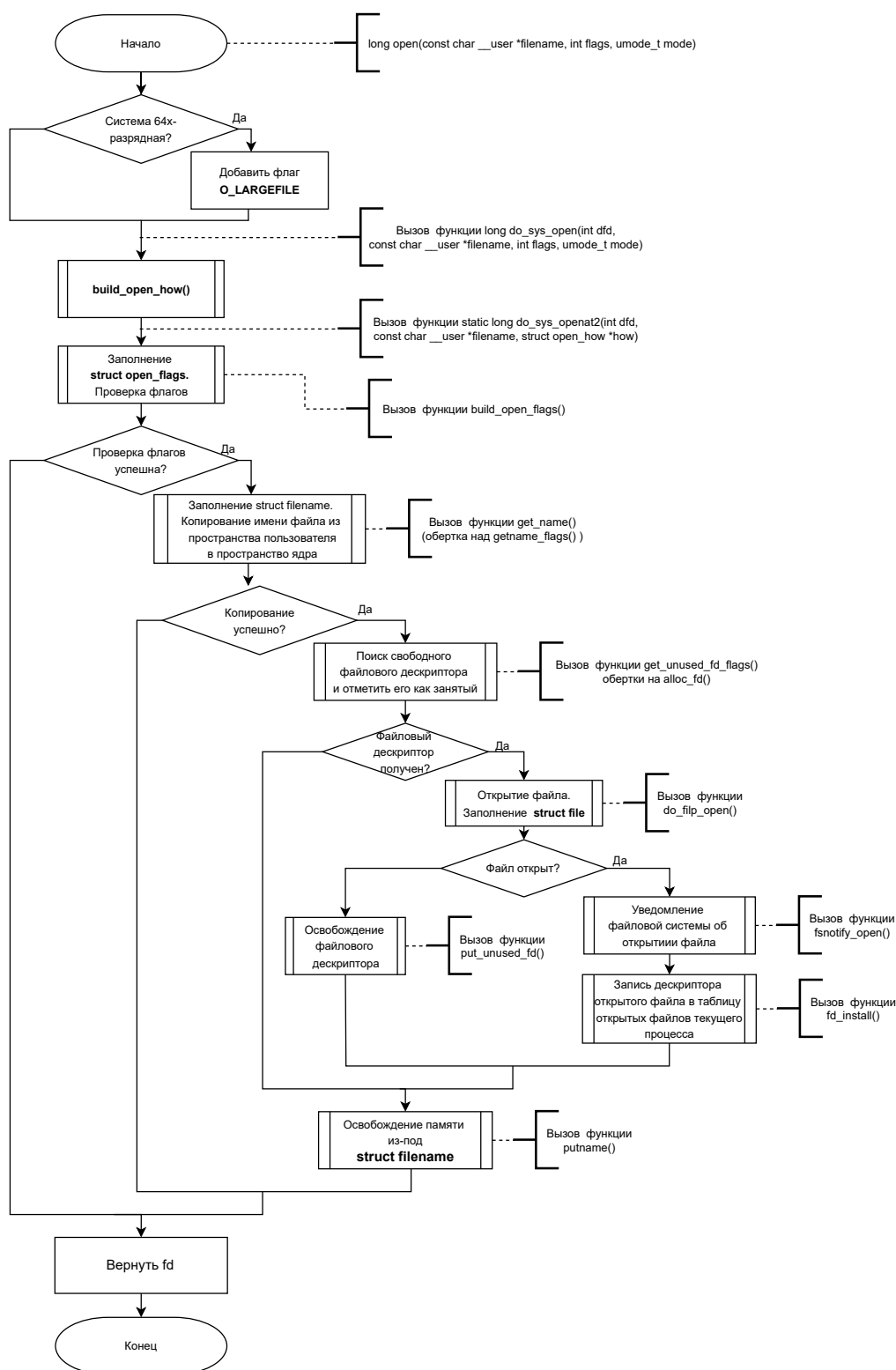


Рисунок 2.1 – Схема работы функции open

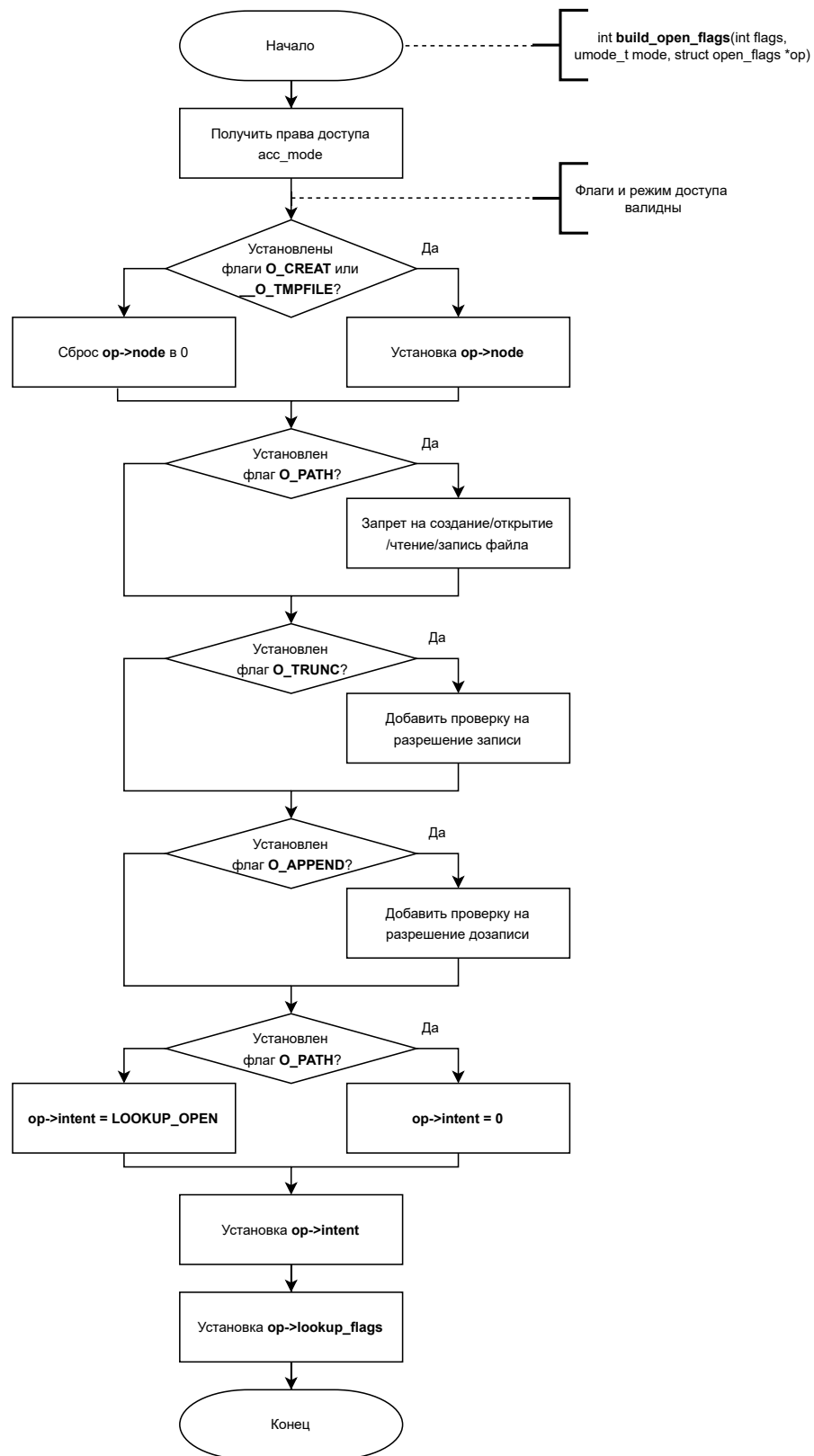


Рисунок 2.2 – Схема работы функции build_open_flags

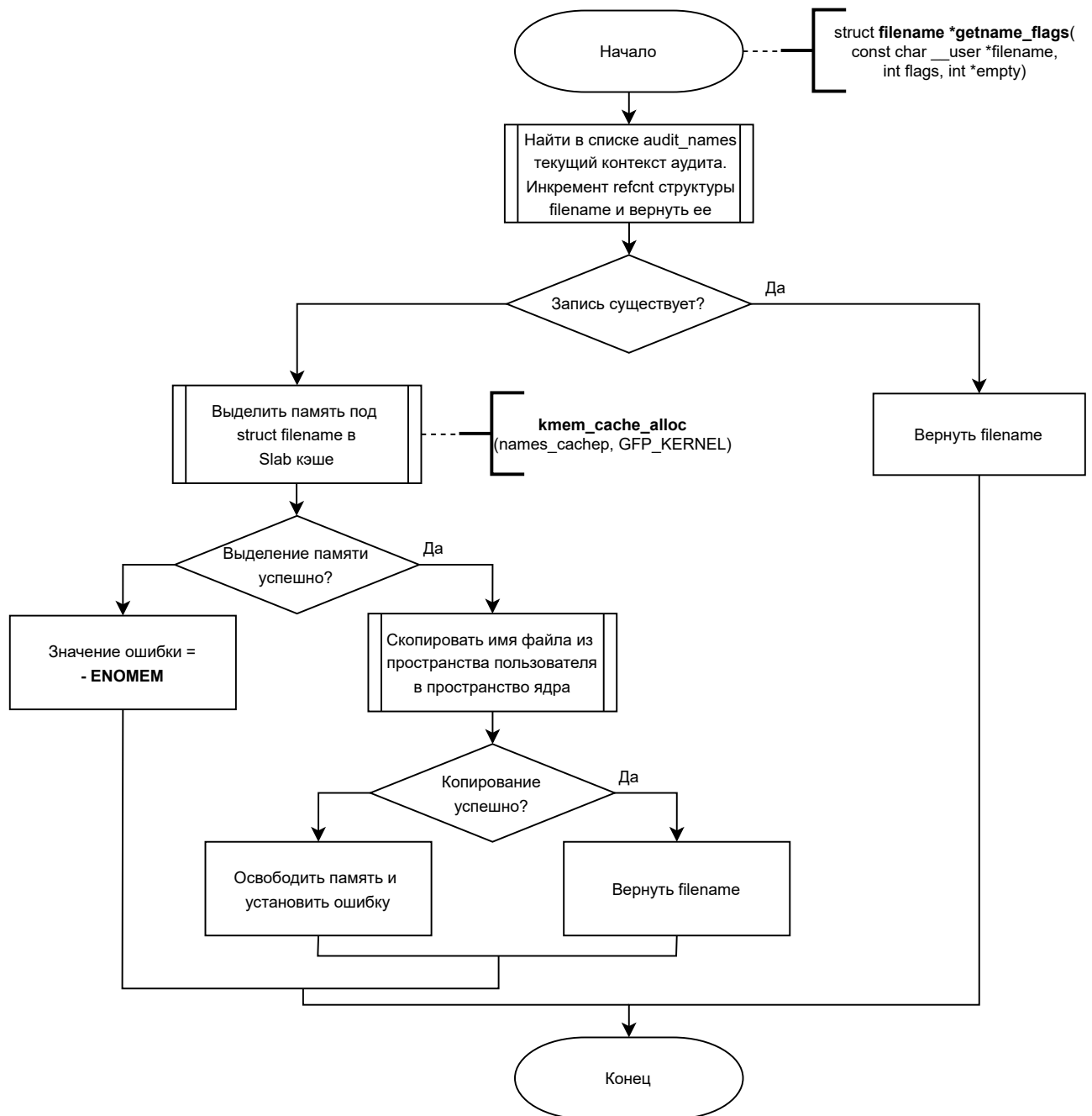


Рисунок 2.3 – Схема работы функции getname_flags

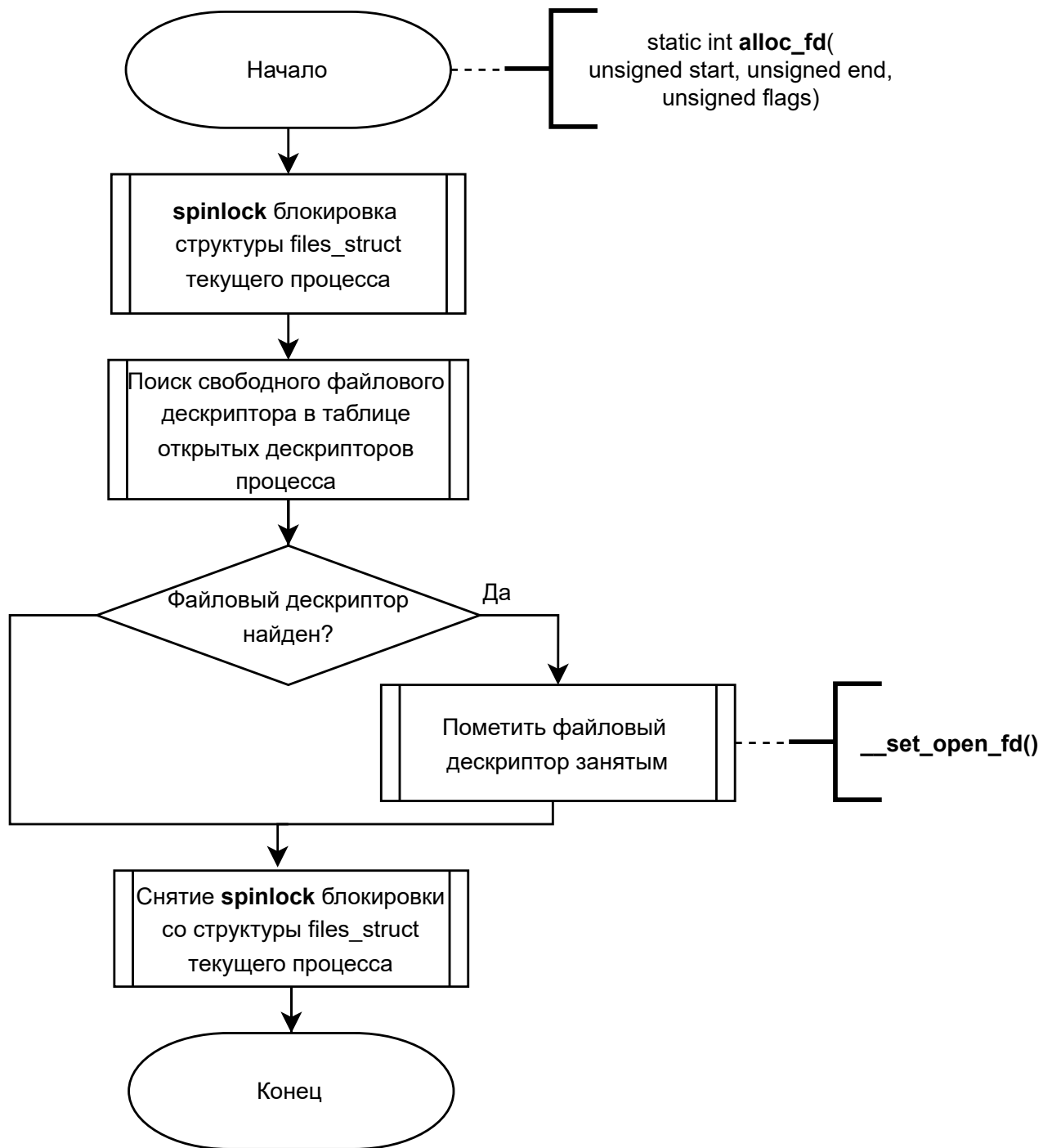


Рисунок 2.4 – Схема работы функции `alloc_fd`

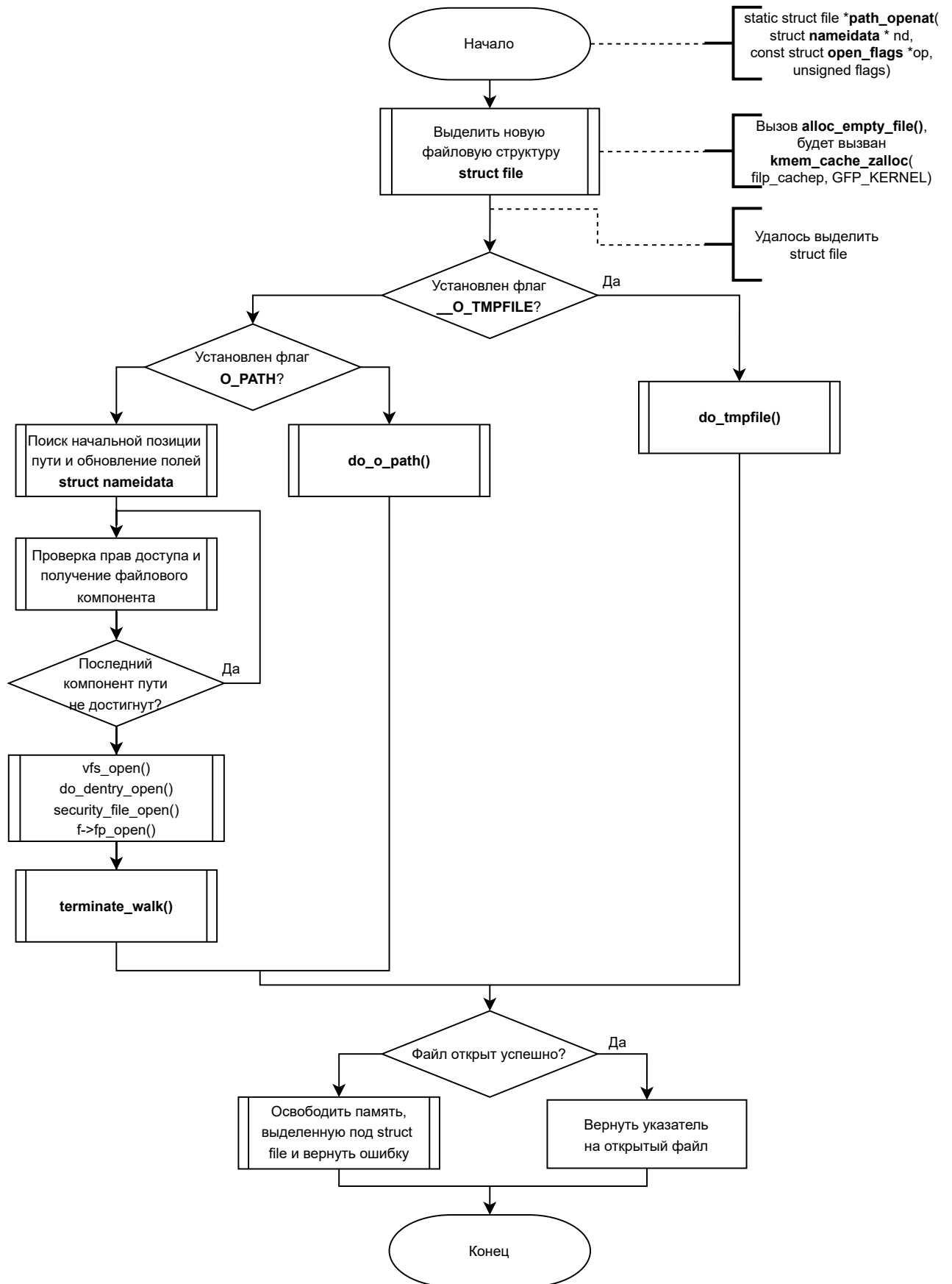


Рисунок 2.5 – Схема работы функции path_openat

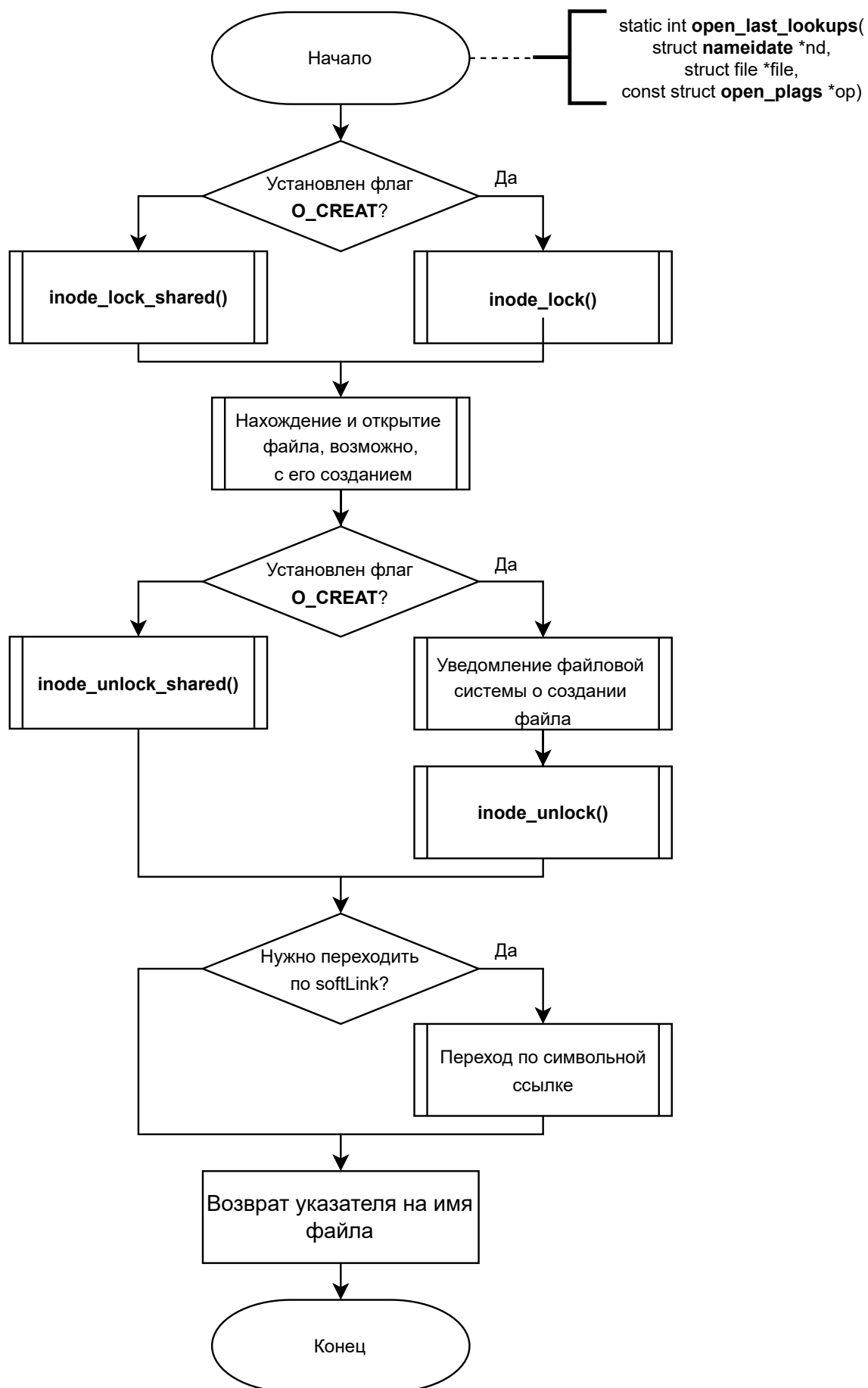


Рисунок 2.6 – Схема работы функции open_last_lookups

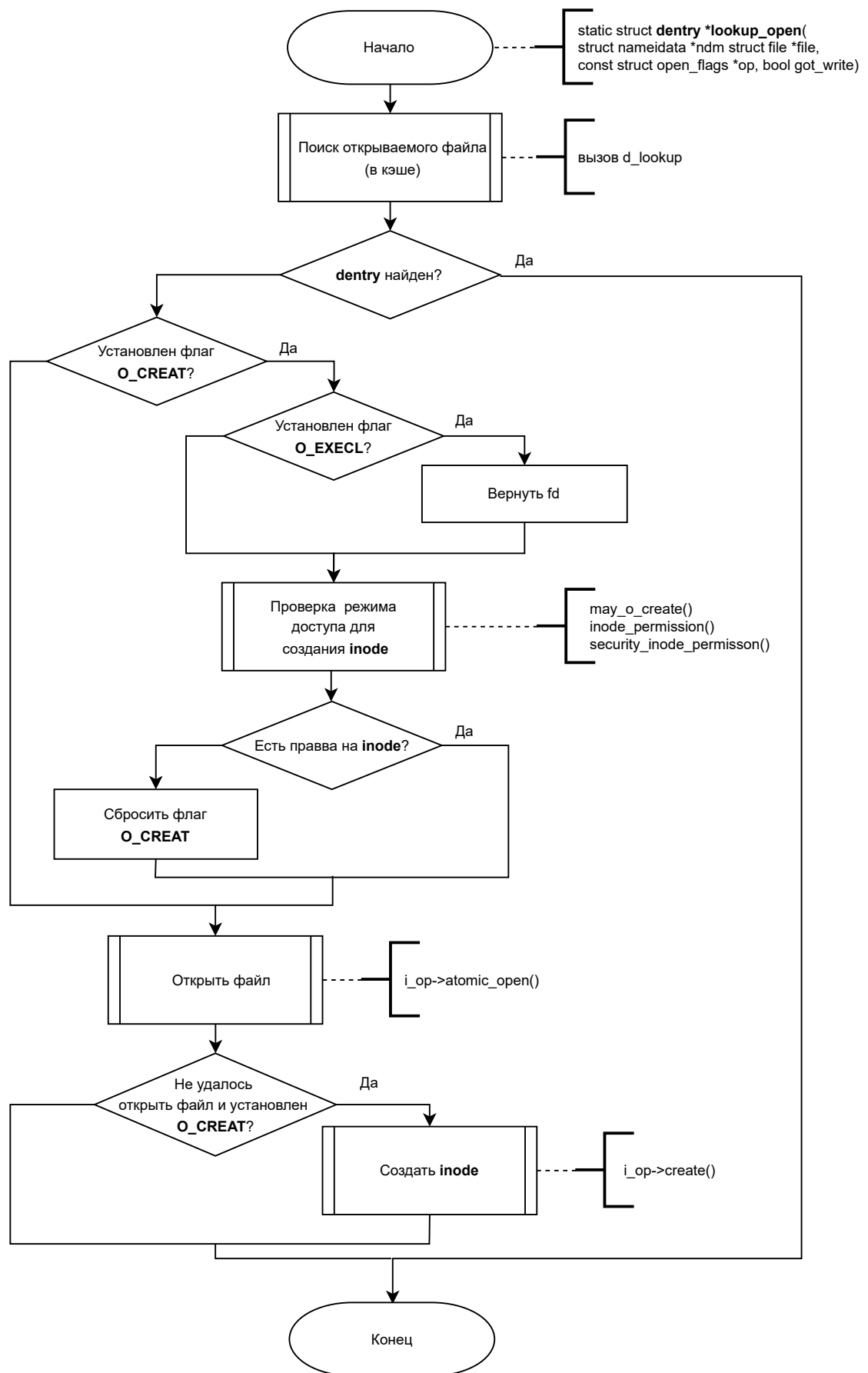


Рисунок 2.7 – Схема работы функции lookup_open

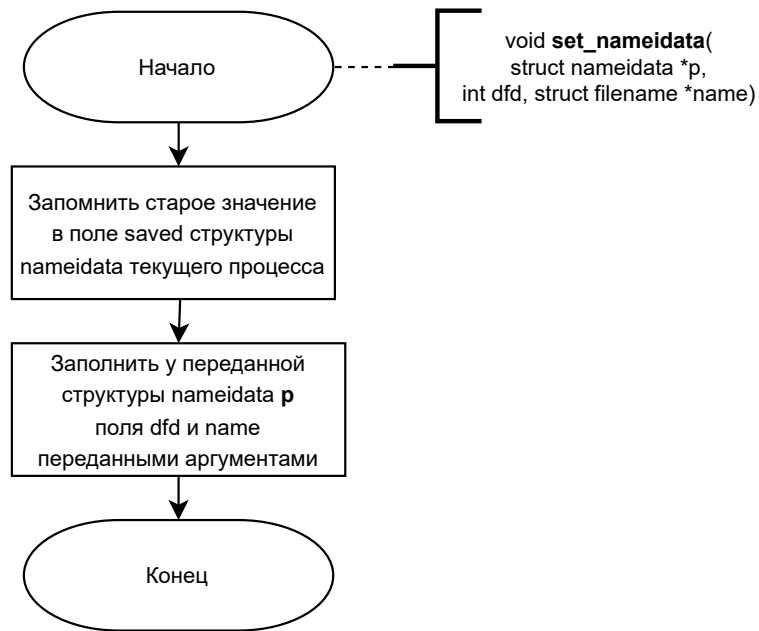


Рисунок 2.8 – Схема работы функции set_nameidata

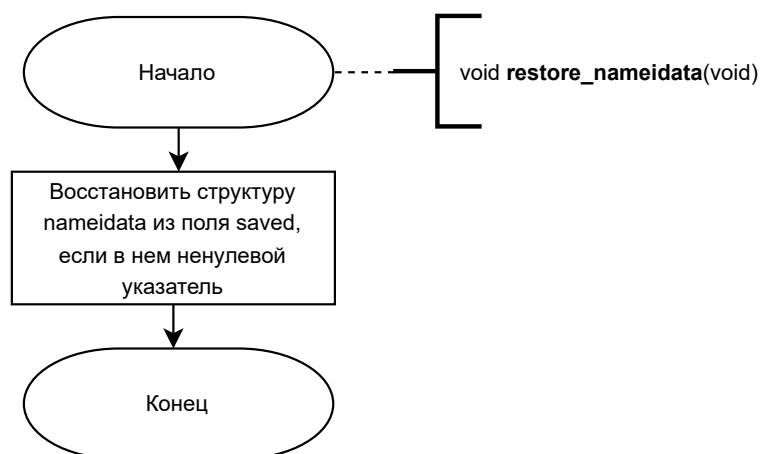


Рисунок 2.9 – Схема работы функции restore_nameidata