



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 дисциплине «Операционные системы»

Тема Системный вызов open

Студент Костев Д.И

Группа ИУ7-61Б

Оценка (баллы) _____

Преподаватель Рязанова Н.Ю.

Используемые структуры

Листинг 1 – Структура 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 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 nameidata

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

Схемы

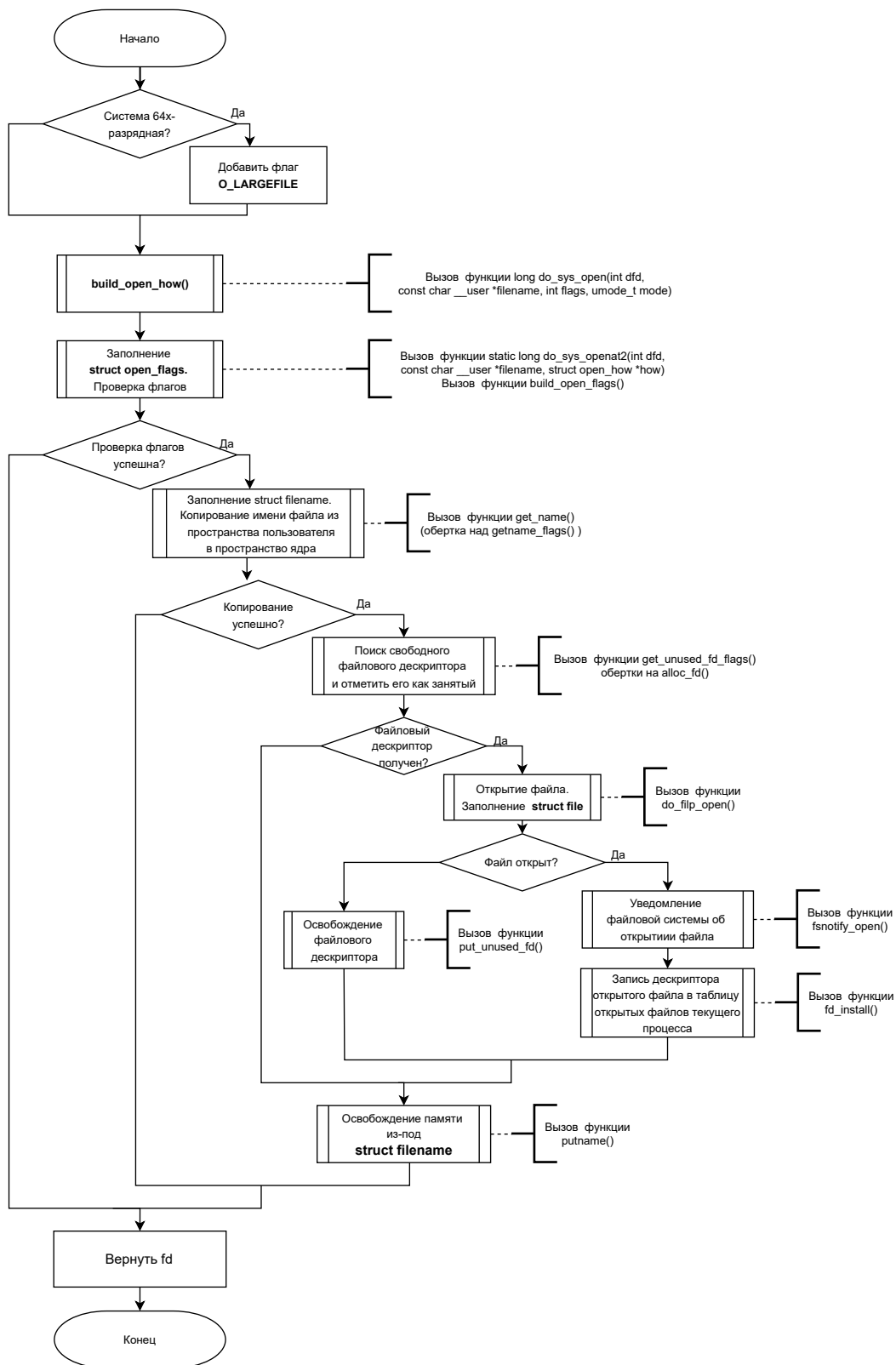


Рисунок 1 – Схема работы функции open

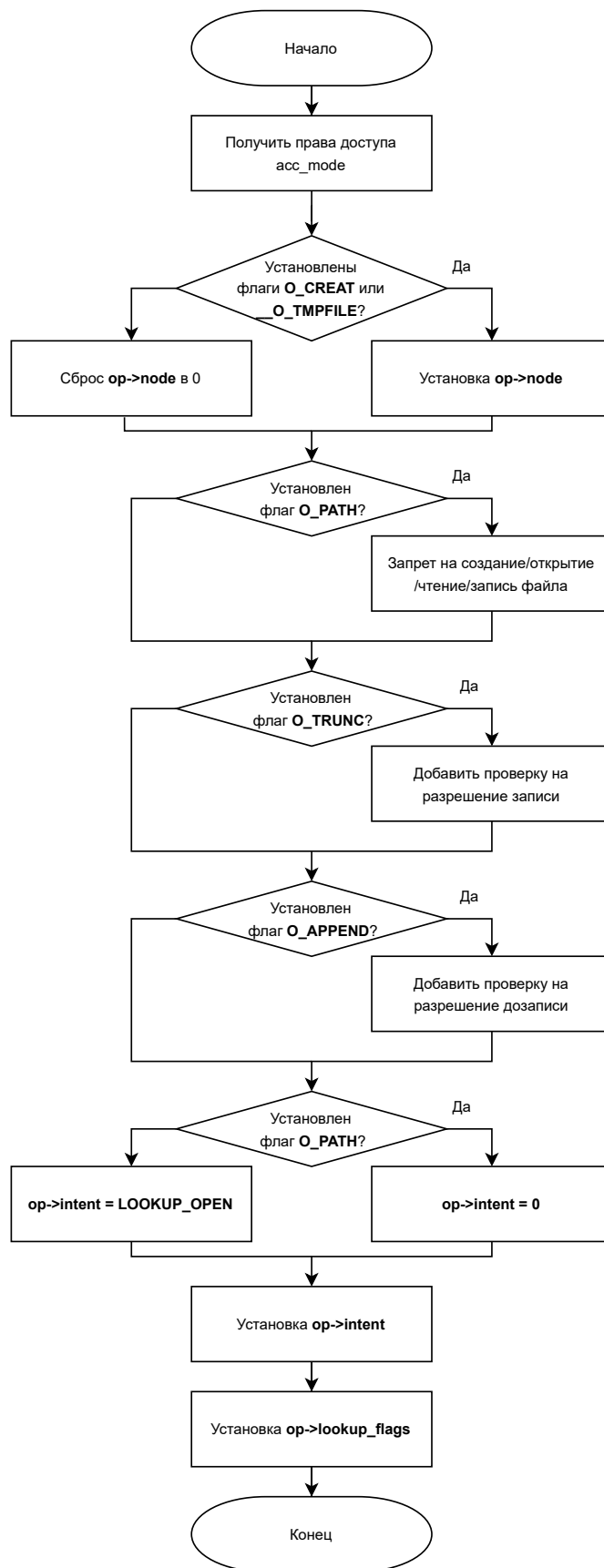


Рисунок 2 – Схема работы функции `build_open_flags`

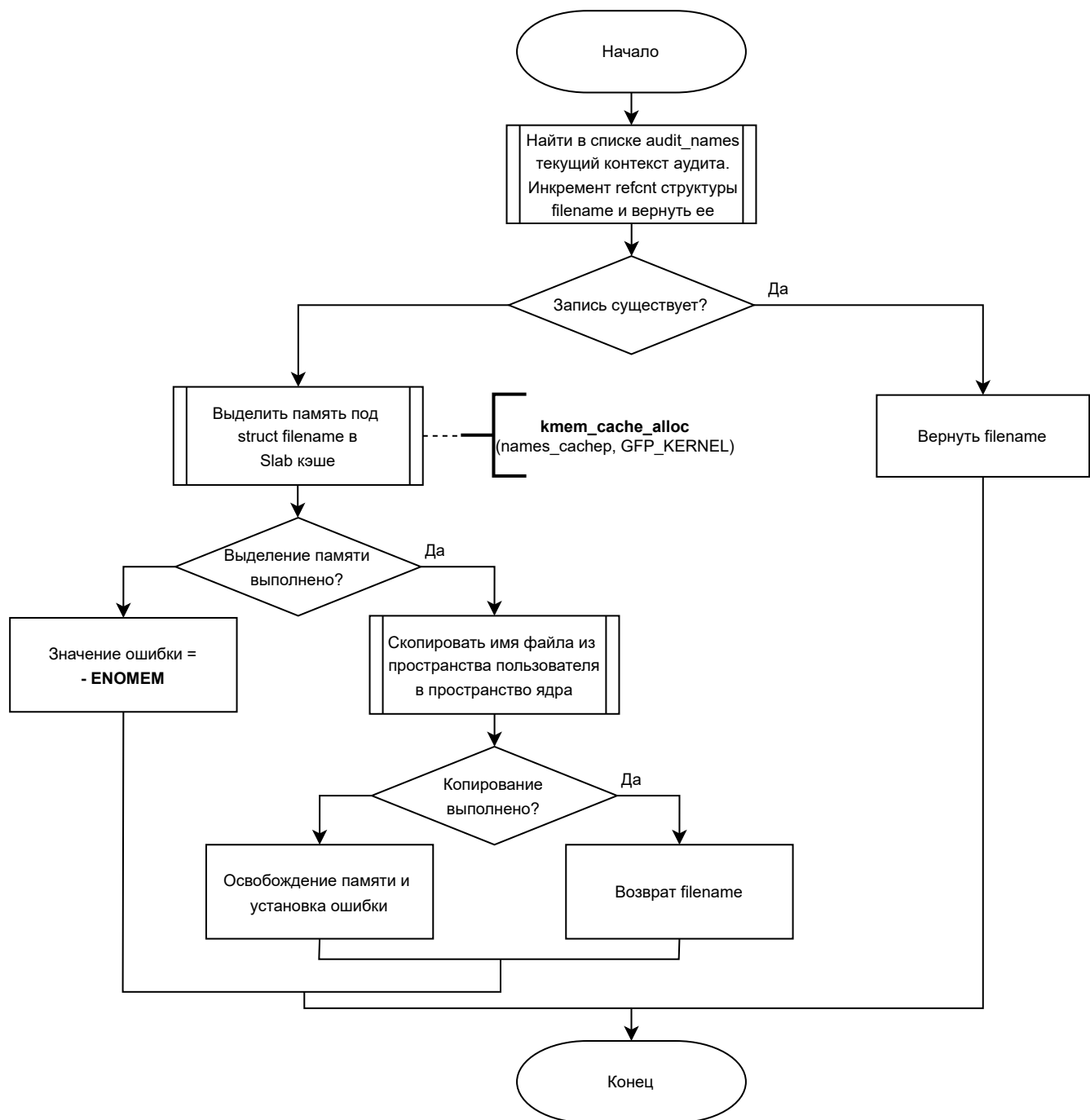


Рисунок 3 – Схема работы функции getname_flags

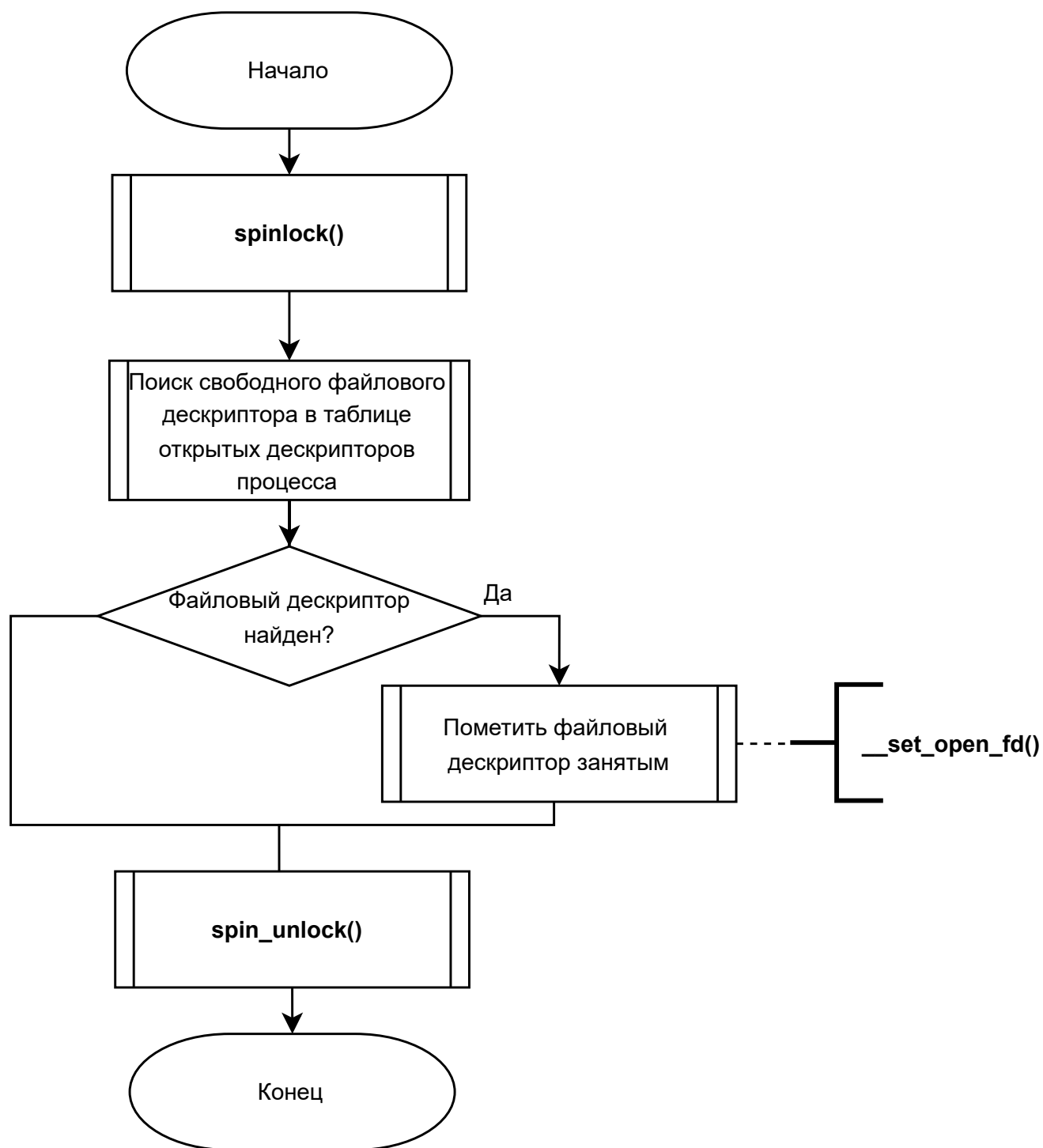


Рисунок 4 – Схема работы функции `alloc_fd`

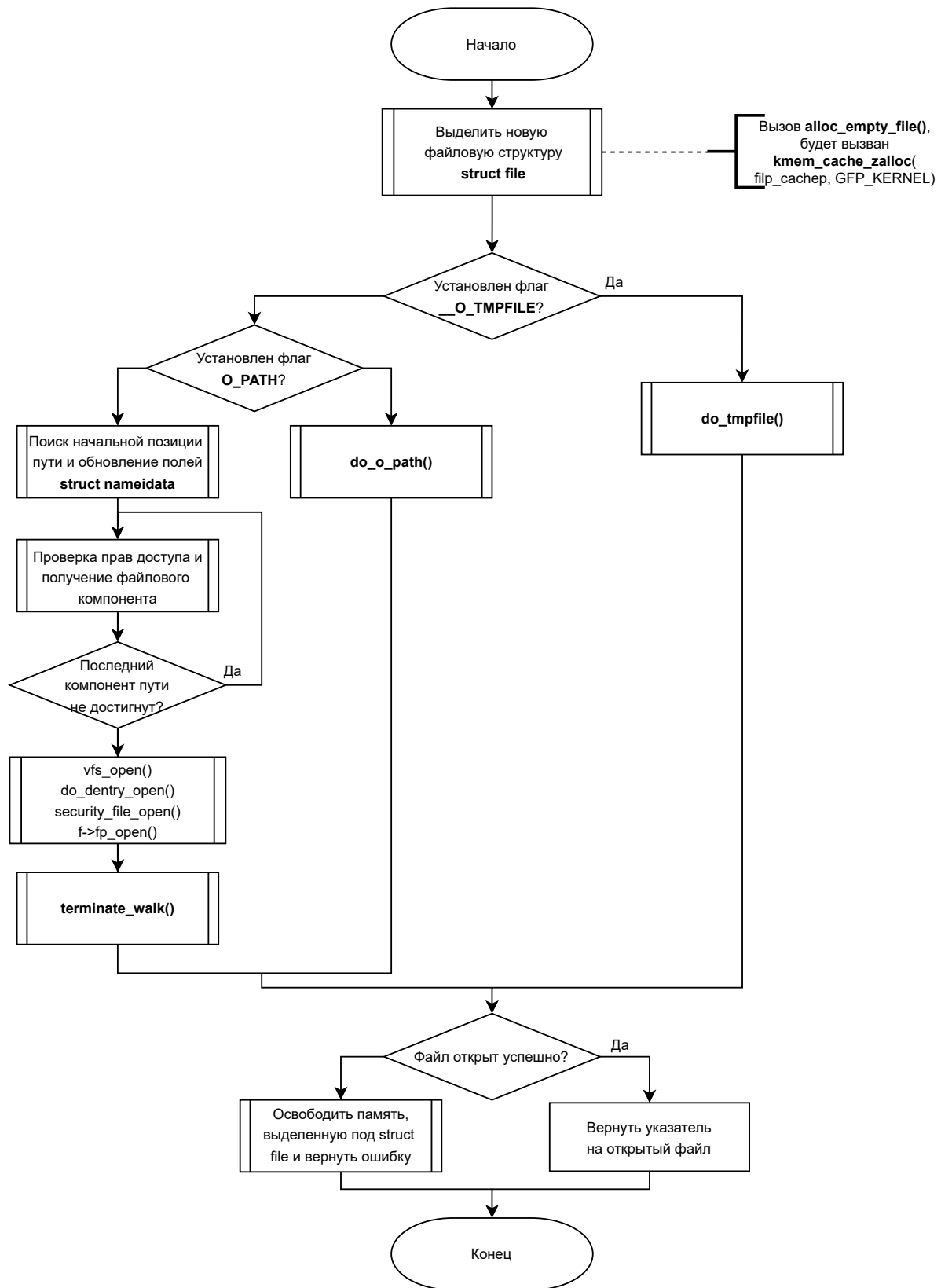


Рисунок 5 – Схема работы функции path_openat

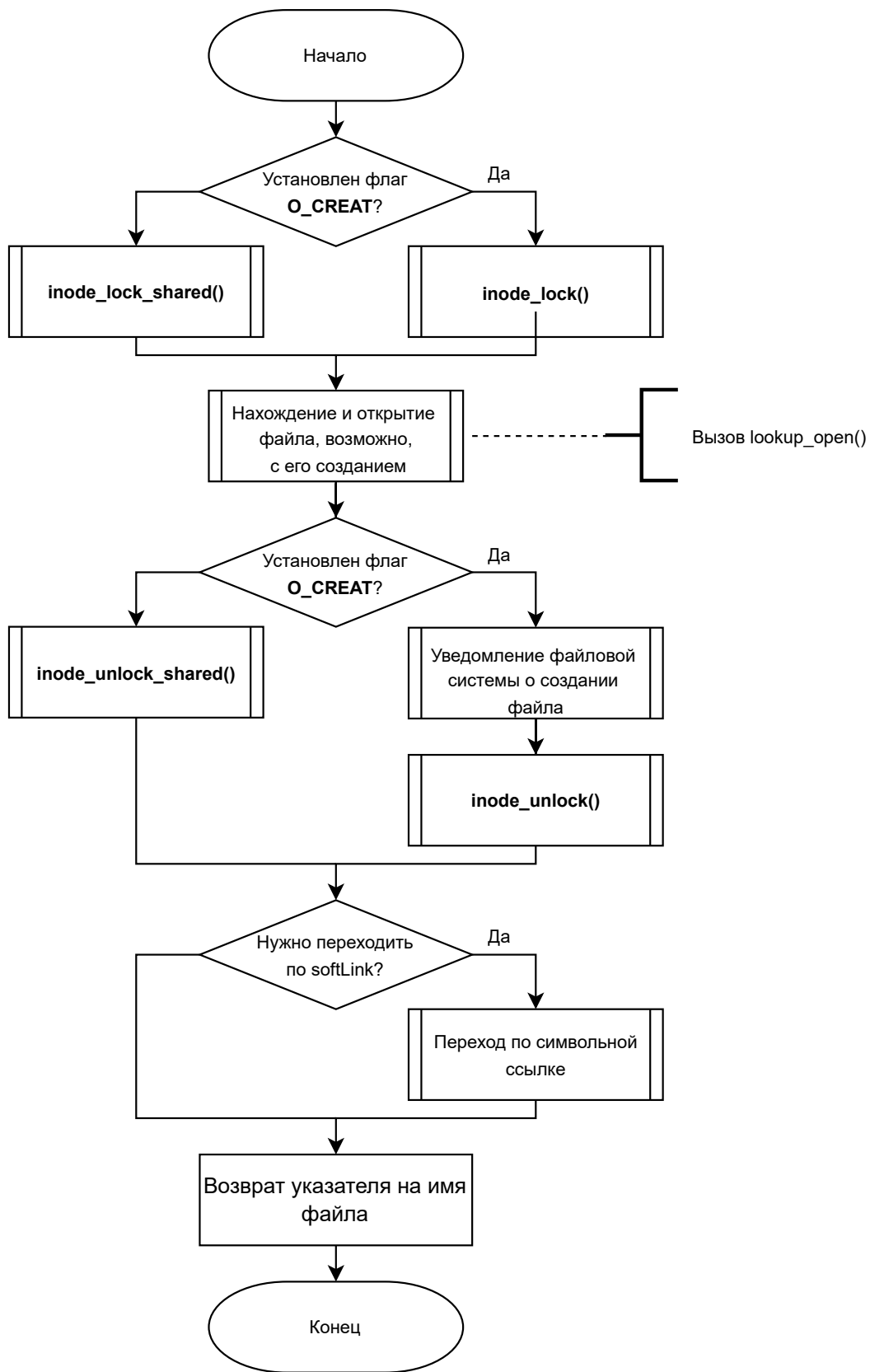


Рисунок 6 – Схема работы функции `open_last_lookups`

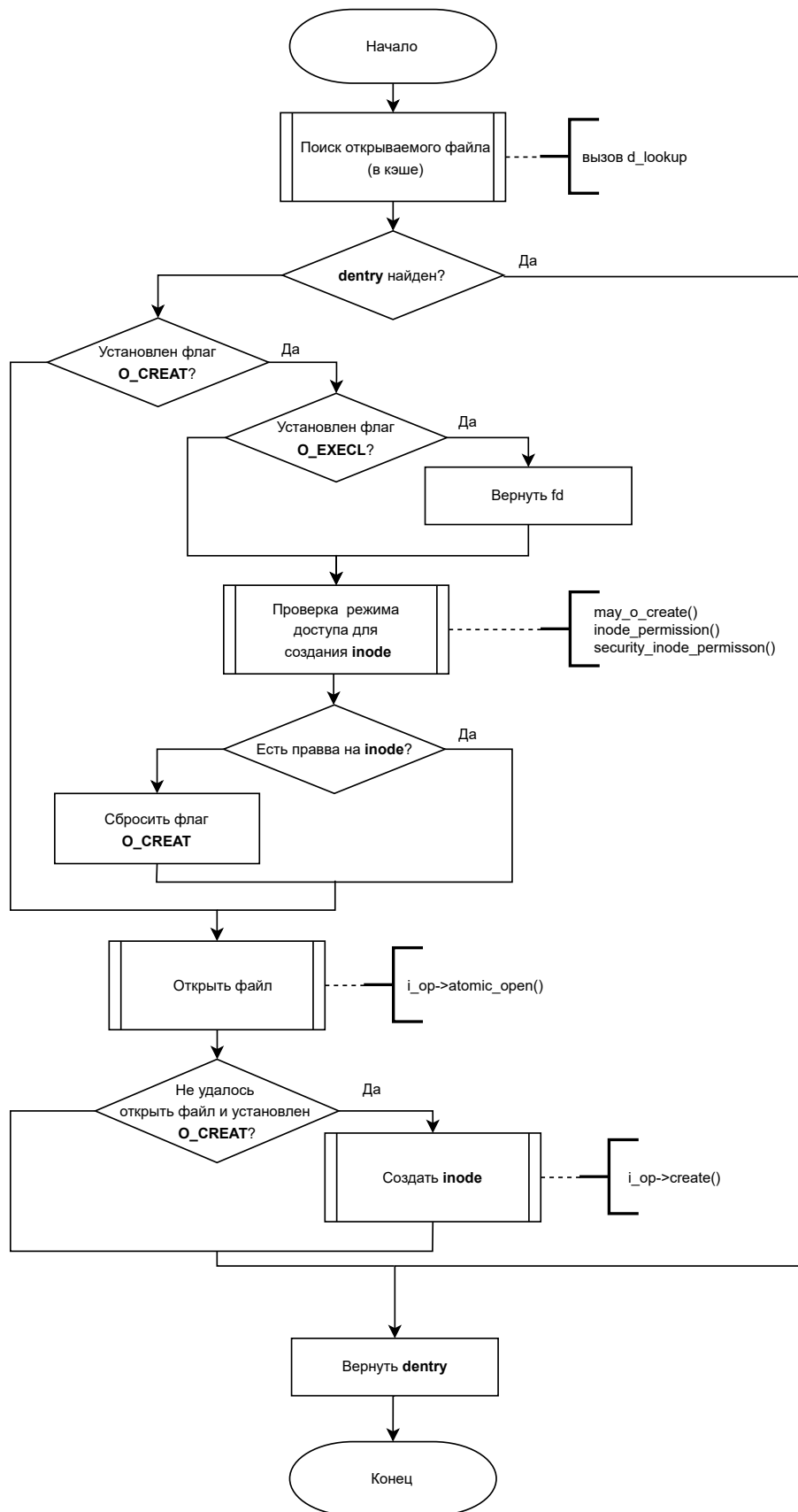


Рисунок 7 – Схема работы функции lookup_open

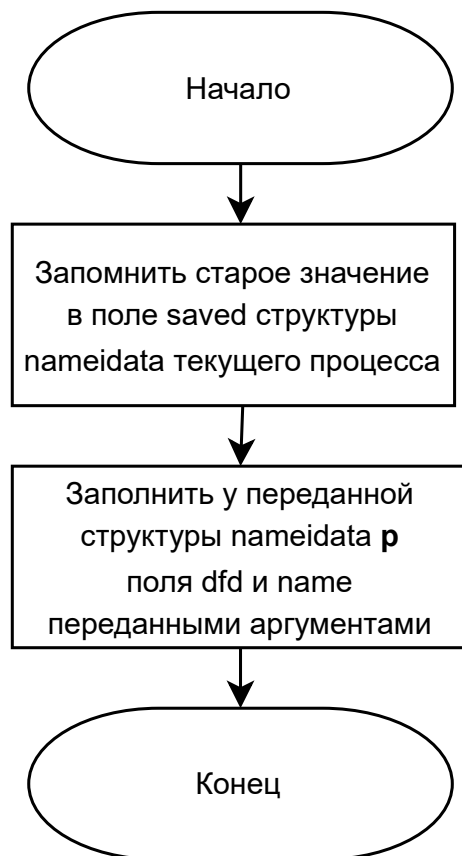


Рисунок 8 – Схема работы функции `set_nameidata`

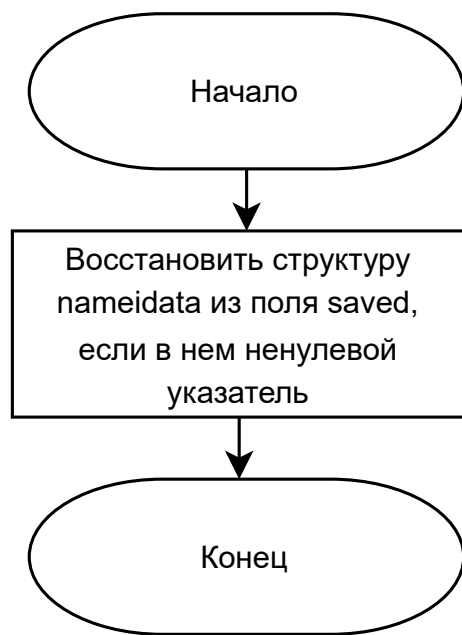


Рисунок 9 – Схема работы функции `restore_nameidata`