



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 курсу «Операционные системы»*

*на тему:*

«Системный вызов open»

Студент группы ИУ7-66Б

\_\_\_\_\_  
(Подпись, дата)

**В. М. Мансуров**

\_\_\_\_\_  
(И.О. Фамилия)

Преподаватель

\_\_\_\_\_  
(Подпись, дата)

**Н. Ю. Рязанова**

\_\_\_\_\_  
(И.О. Фамилия)

2023 г.

## Флаги системного вызова `open()`

Версия ядра: 6.3.2

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

`O_CREAT` — если имя пути не существует, то файл создается как обычный файл.

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

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

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

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

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

`O_DSYNC` — операции записи в файл будут завершены в соответствии с требованиями целостности данных синхронизированного завершения ввода-вывода.

`O_NOATIME` — запрет на обновление времени последнего доступа к файлу при его чтении.

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

`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`.

# 1 Схемы алгоритмов

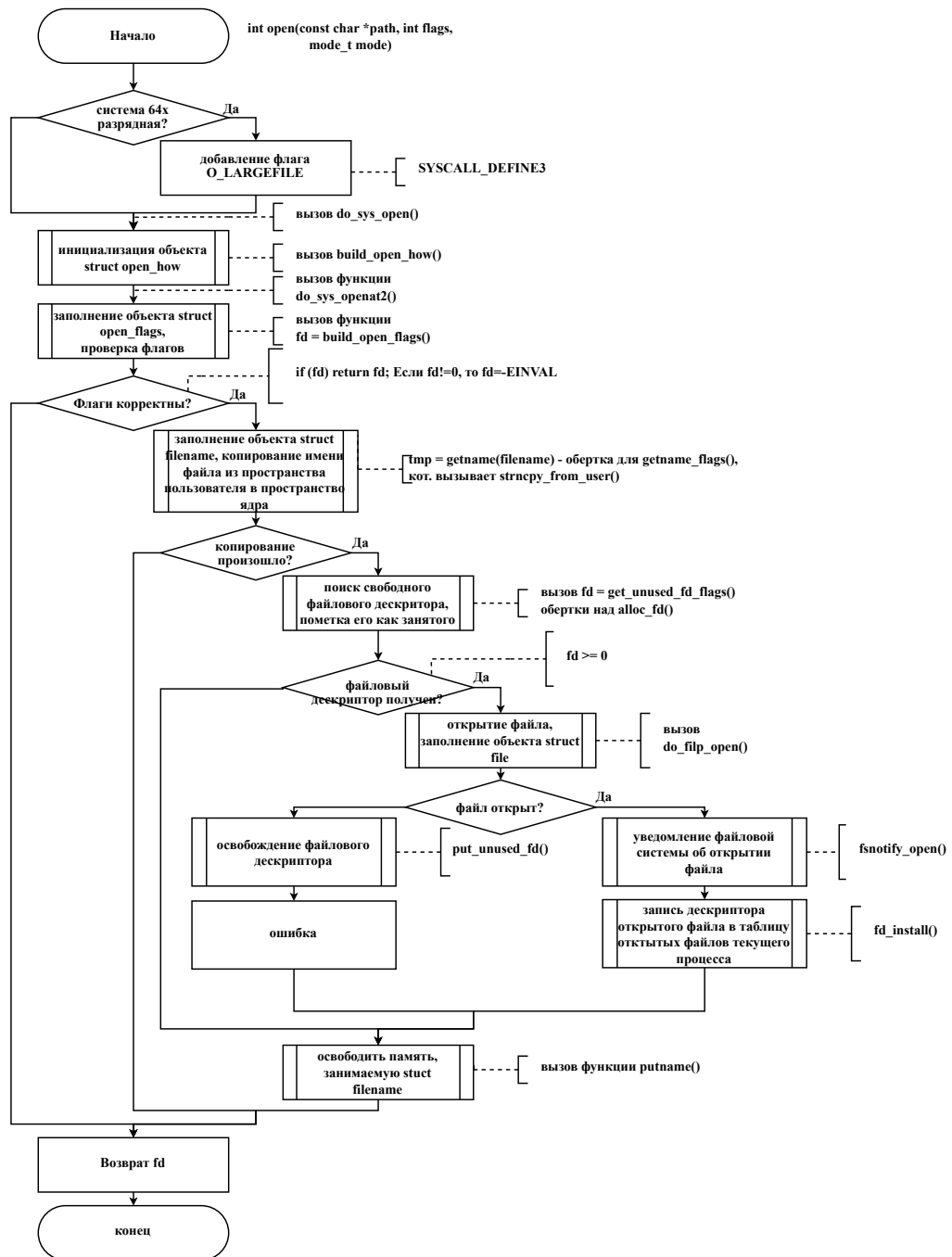


Рисунок 1 – SYSCALL\_DEFINE3

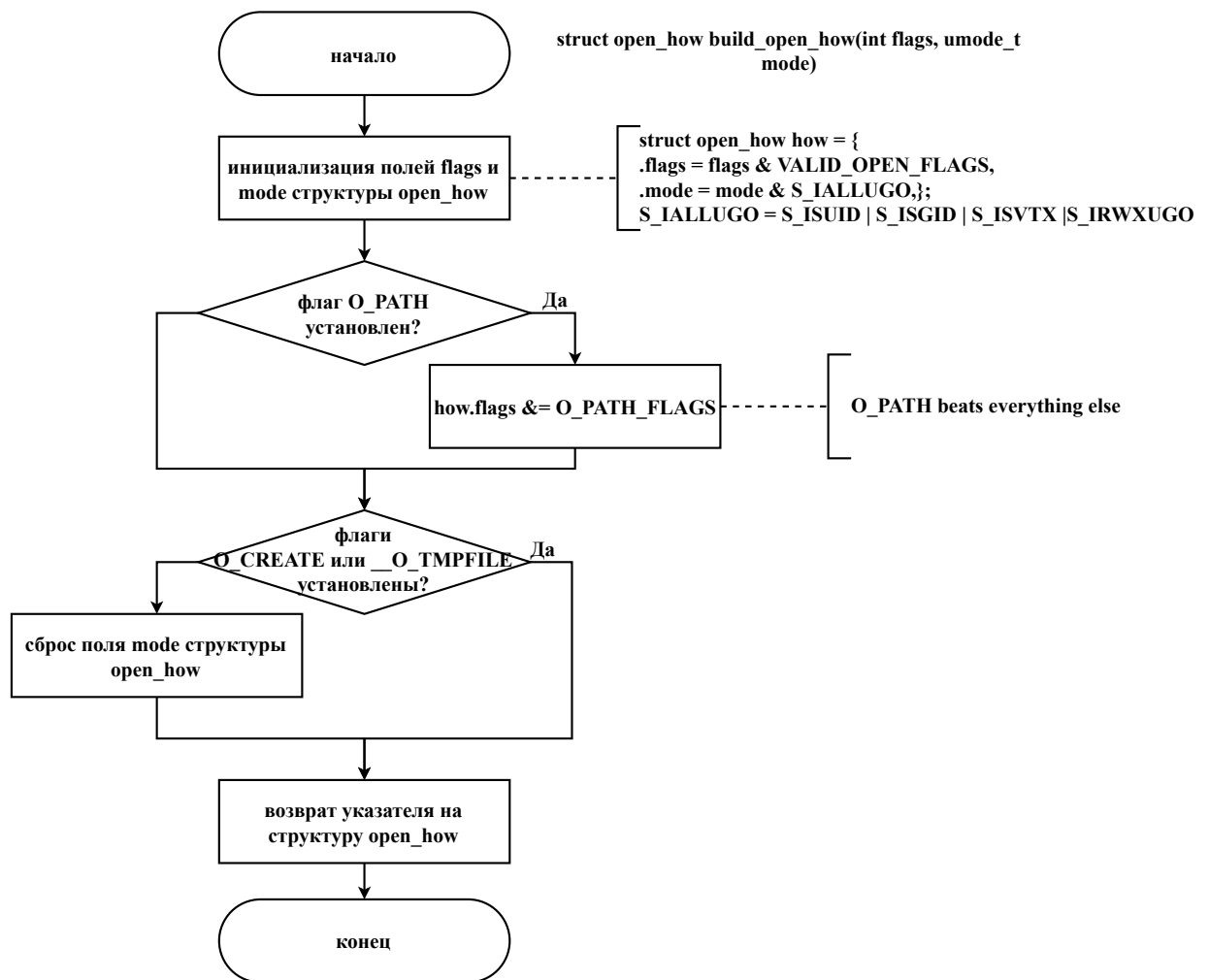


Рисунок 2 – build\_open\_how()

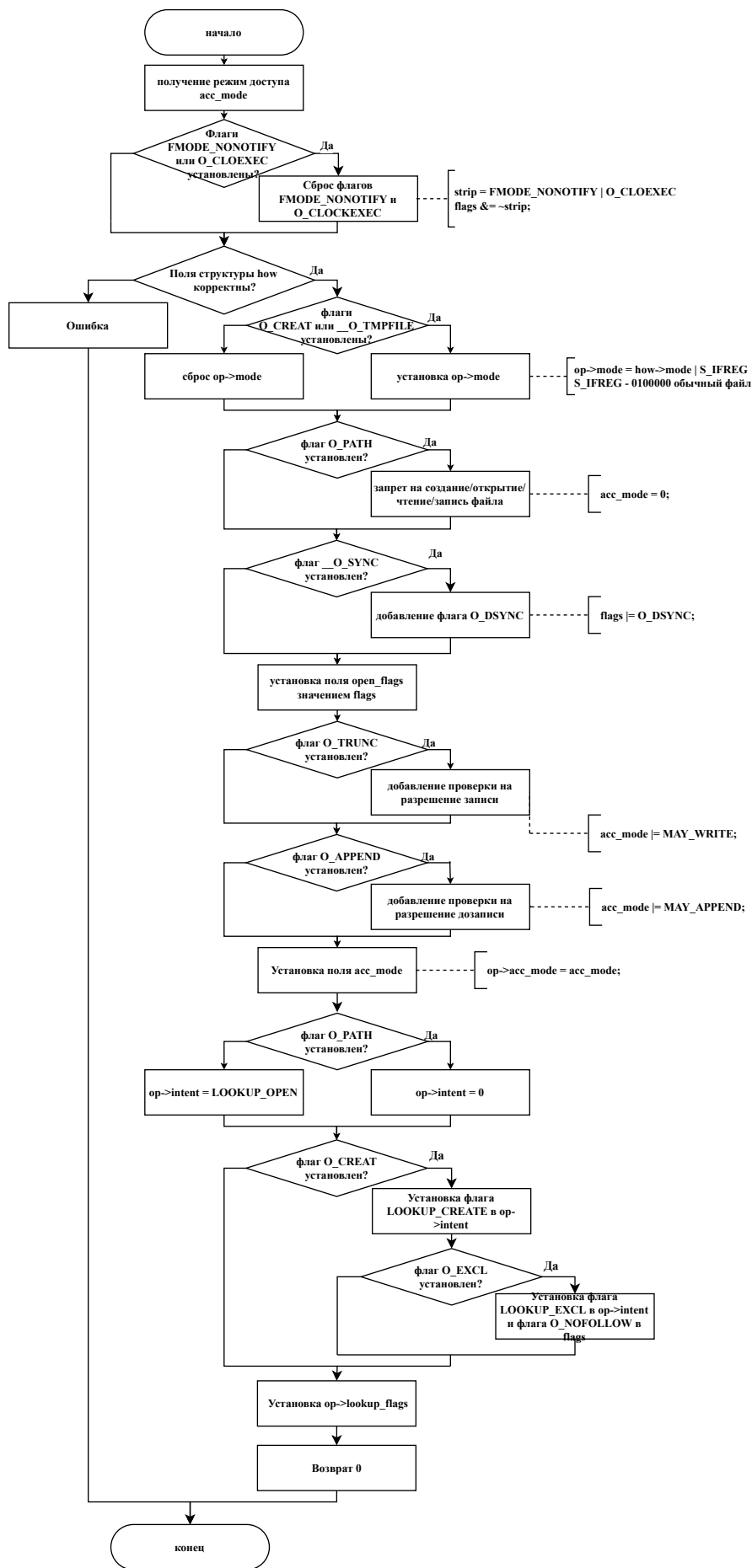


Рисунок 3 – build\_open\_flags()

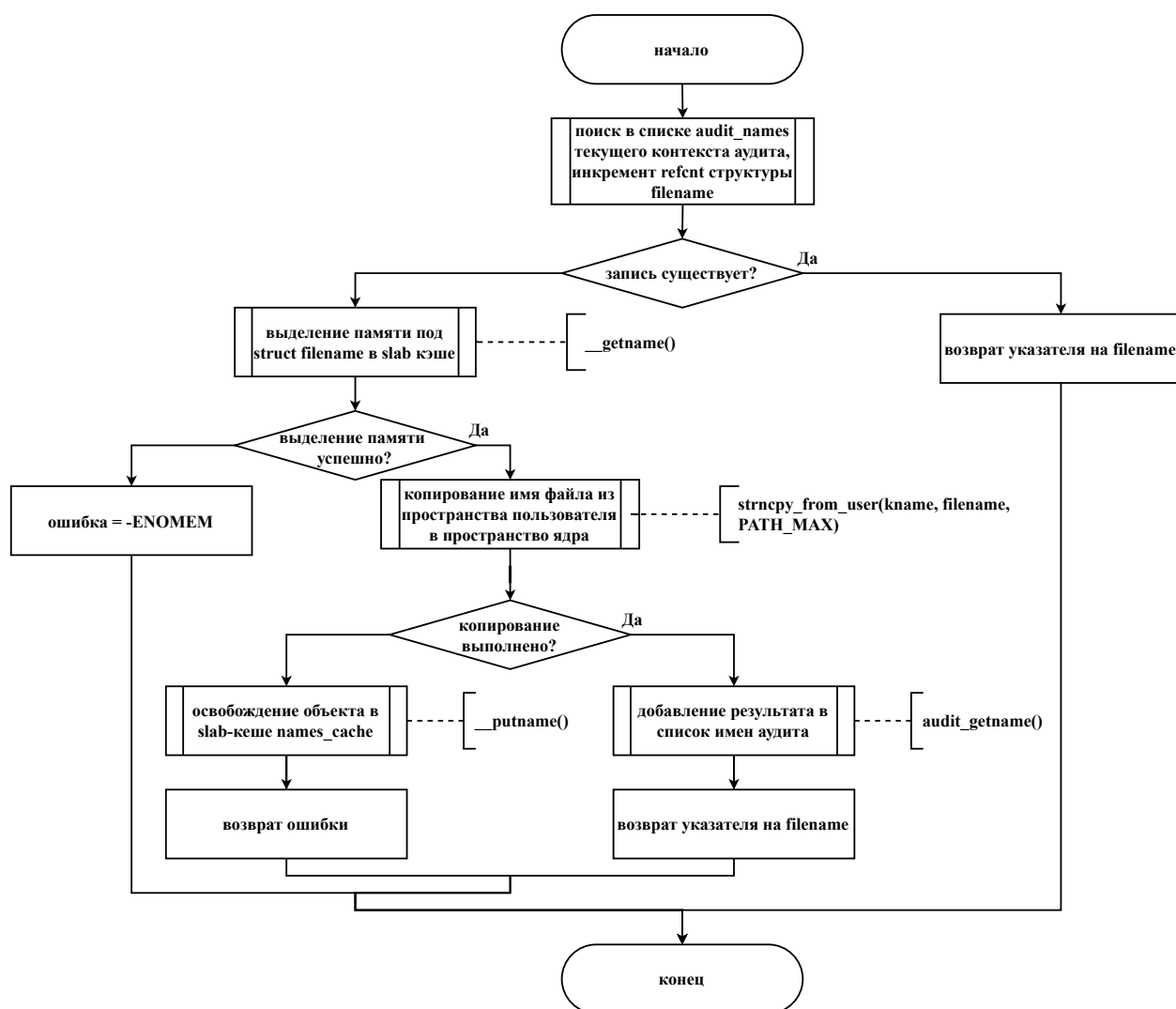


Рисунок 4 – `getname_flags()`

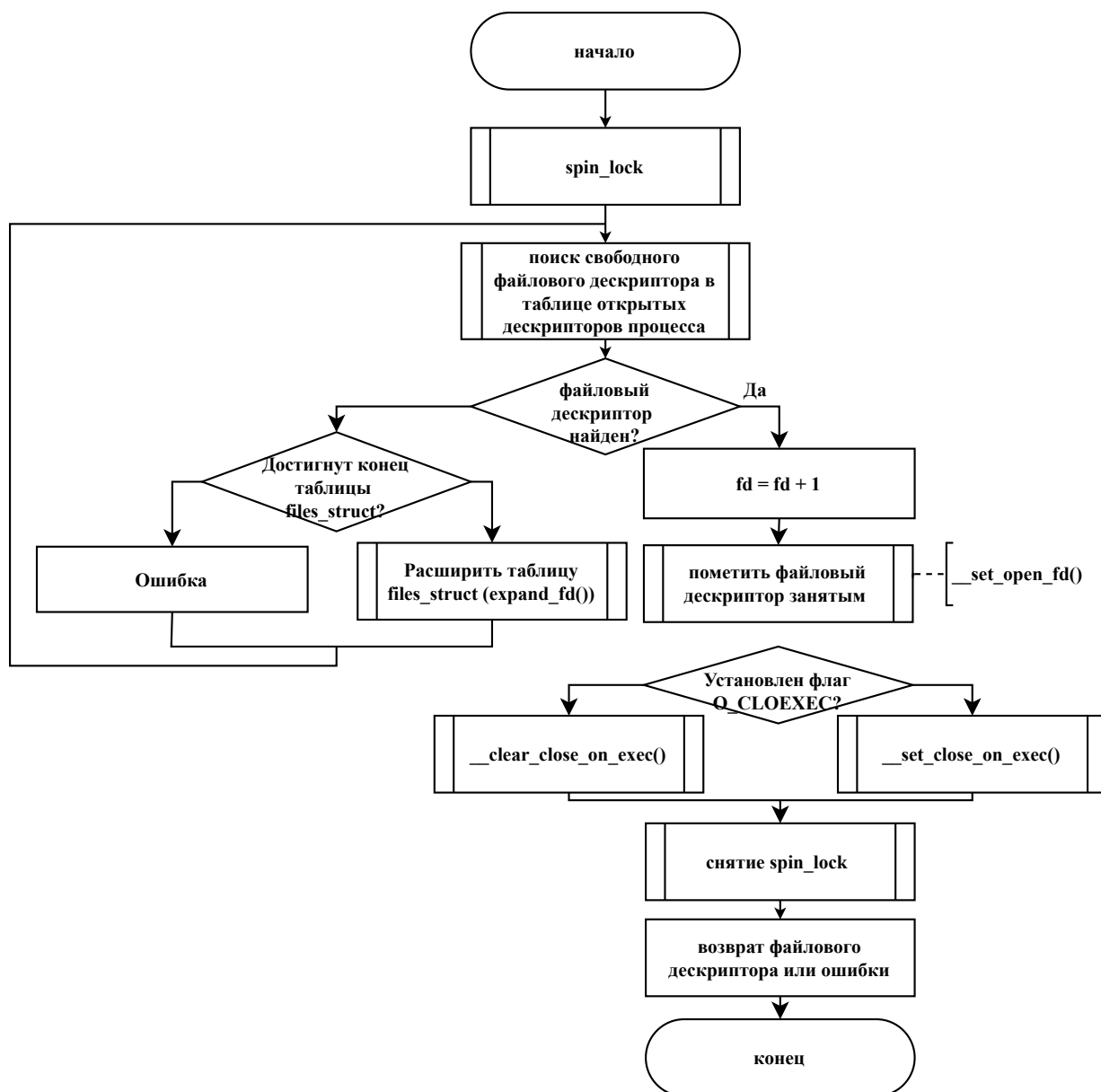


Рисунок 5 – alloc\_fd()



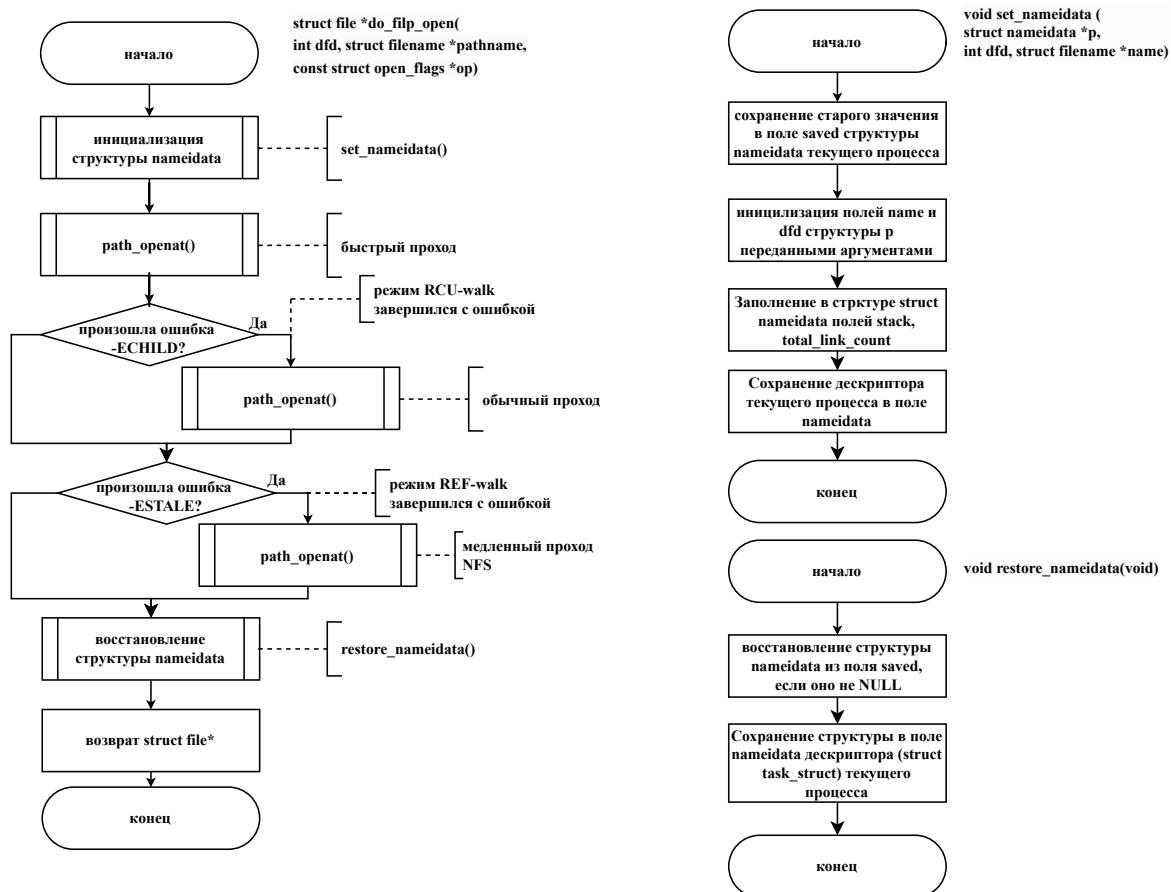


Рисунок 6 – `do_filp_open()`

`LOOKUP_RCU` — флаг используется в VFS для указания, что операция должна выполняться с использованием RCU (Read-Copy-Update).

`LOOKUP_REVAL` — флаг для работы с NFS, указывает, что необходимо выполнить повторную проверку.

`O_APPEND` — может привести к изменению файлов в файловых системах NFS, если несколько процессов одновременно добавляют данные в файл. Это связано с тем, что в NFS не работает флаг `O_APPEND`, поэтому доступ к файлу осуществляется в монопольном режиме, что невозможно без состояния гонки.

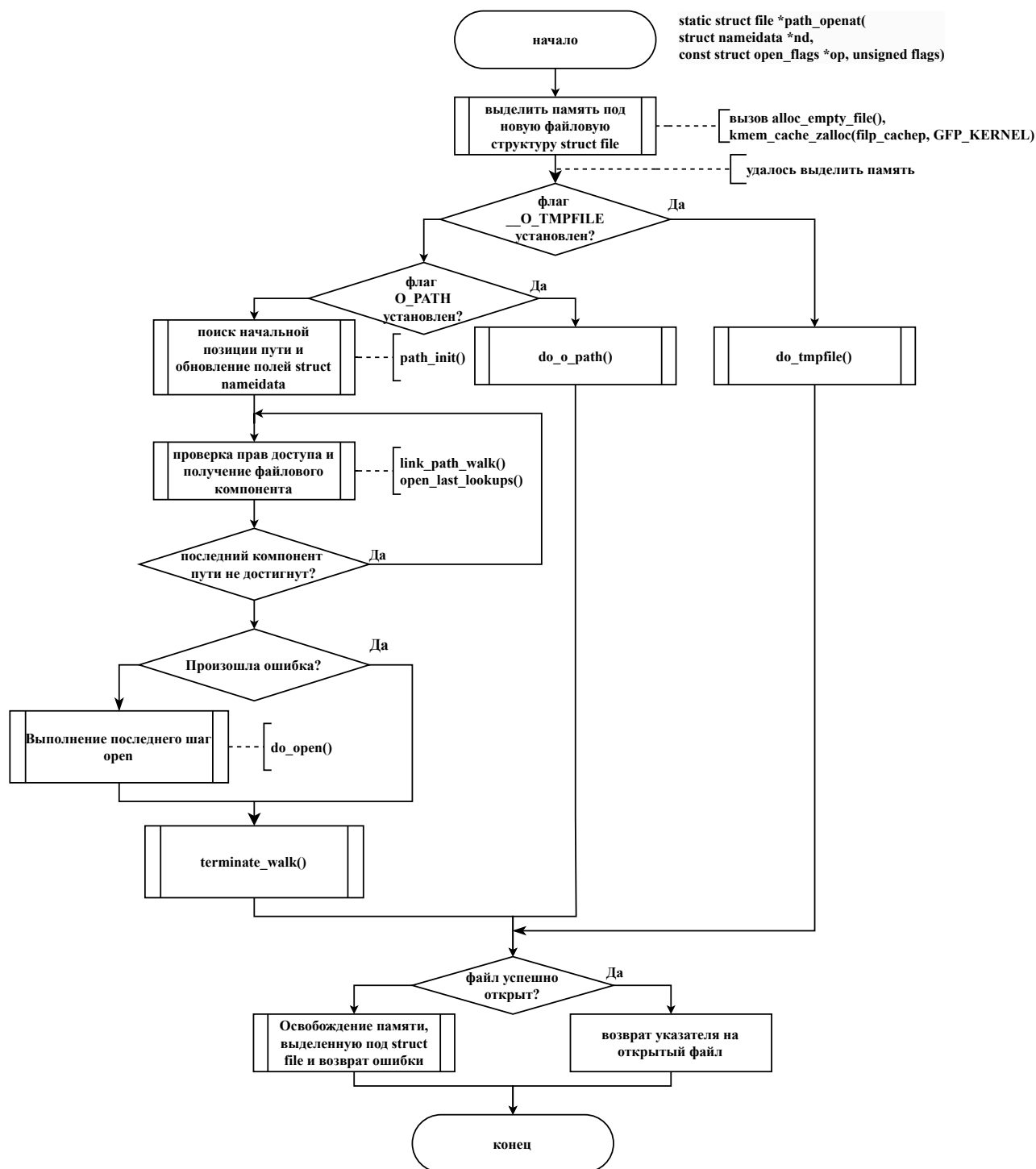


Рисунок 7 – path\_openat()

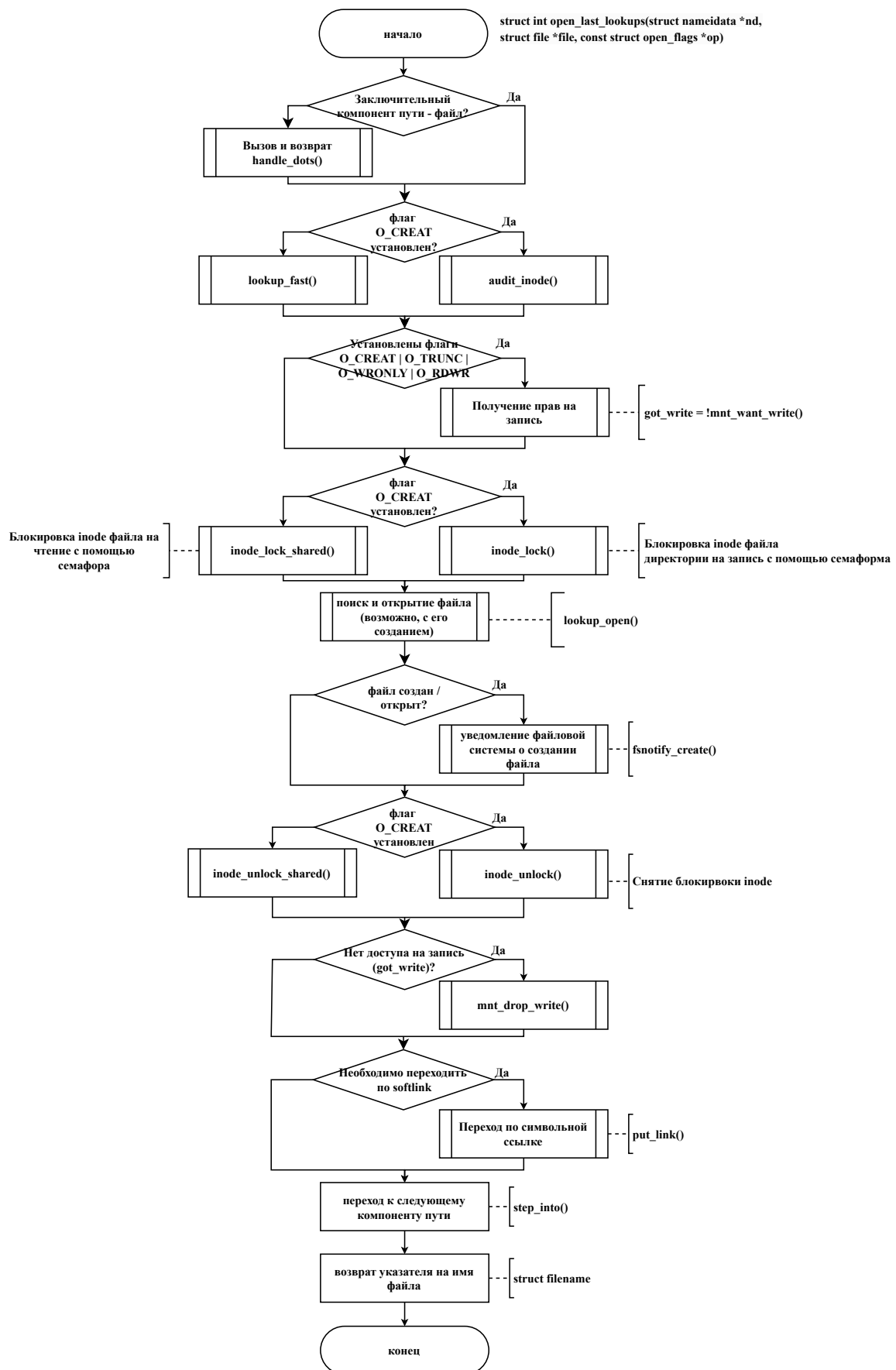


Рисунок 8 – open\_last\_lookups()

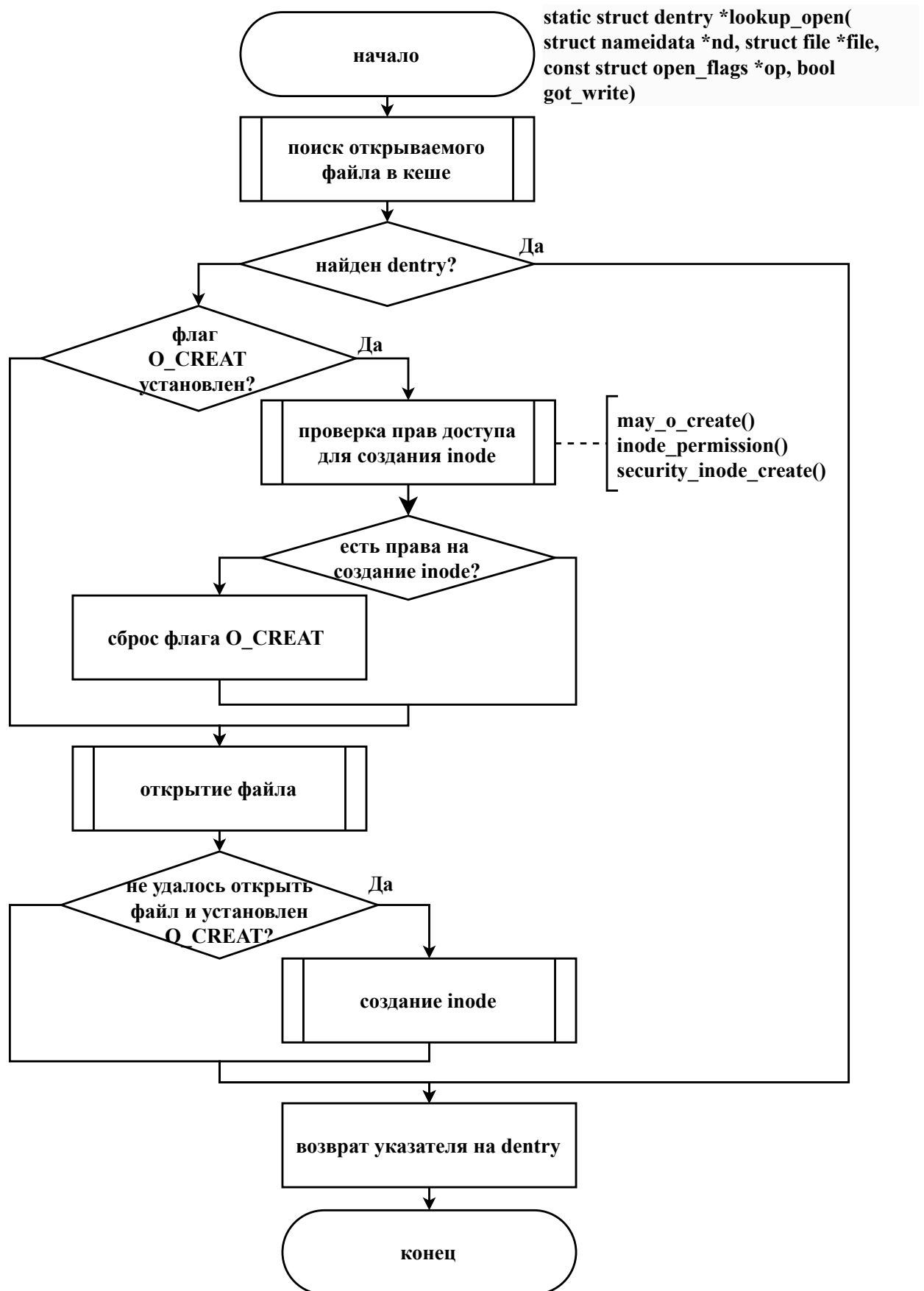


Рисунок 9 – lookup\_open()

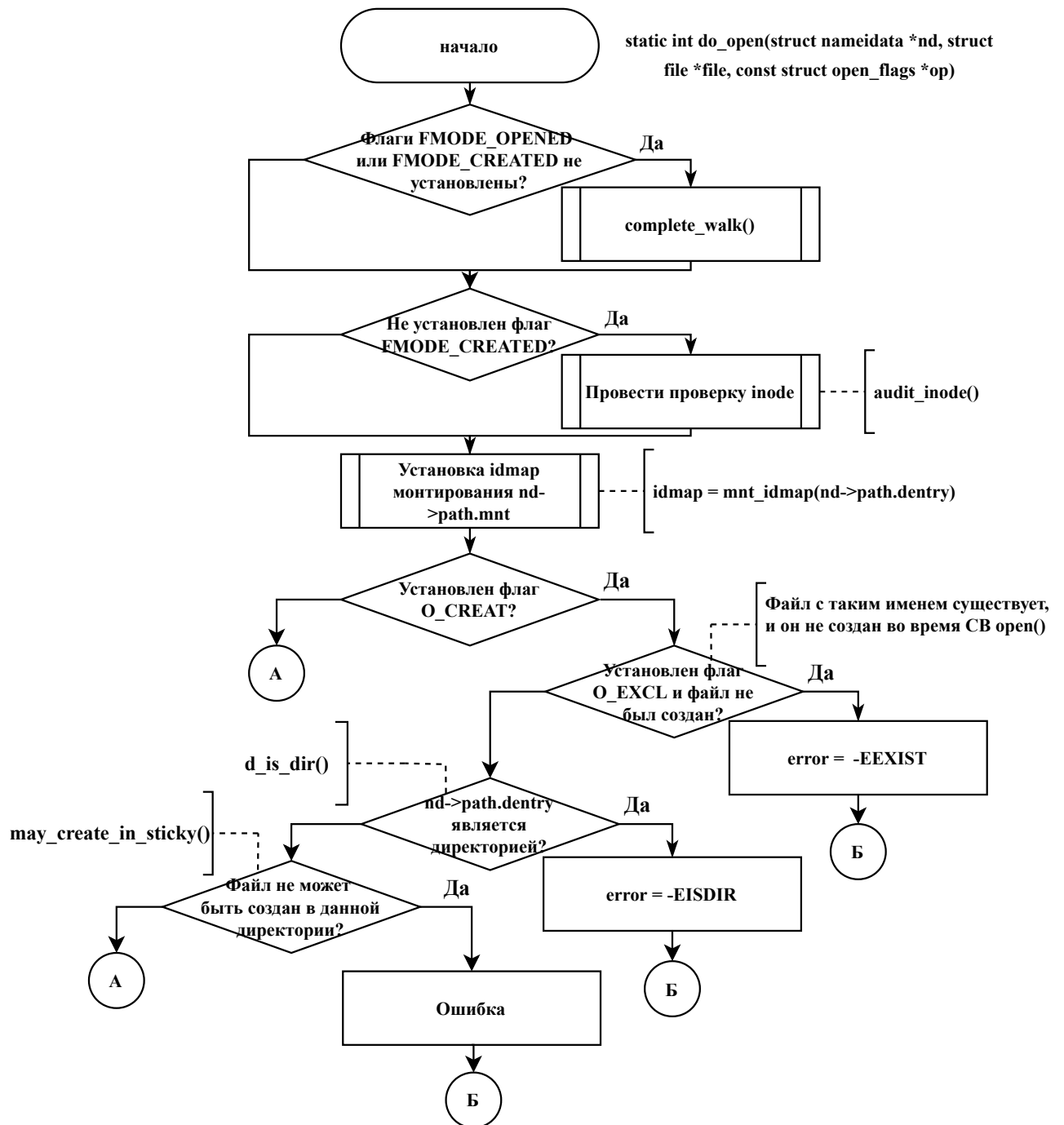


Рисунок 10 – do\_open()

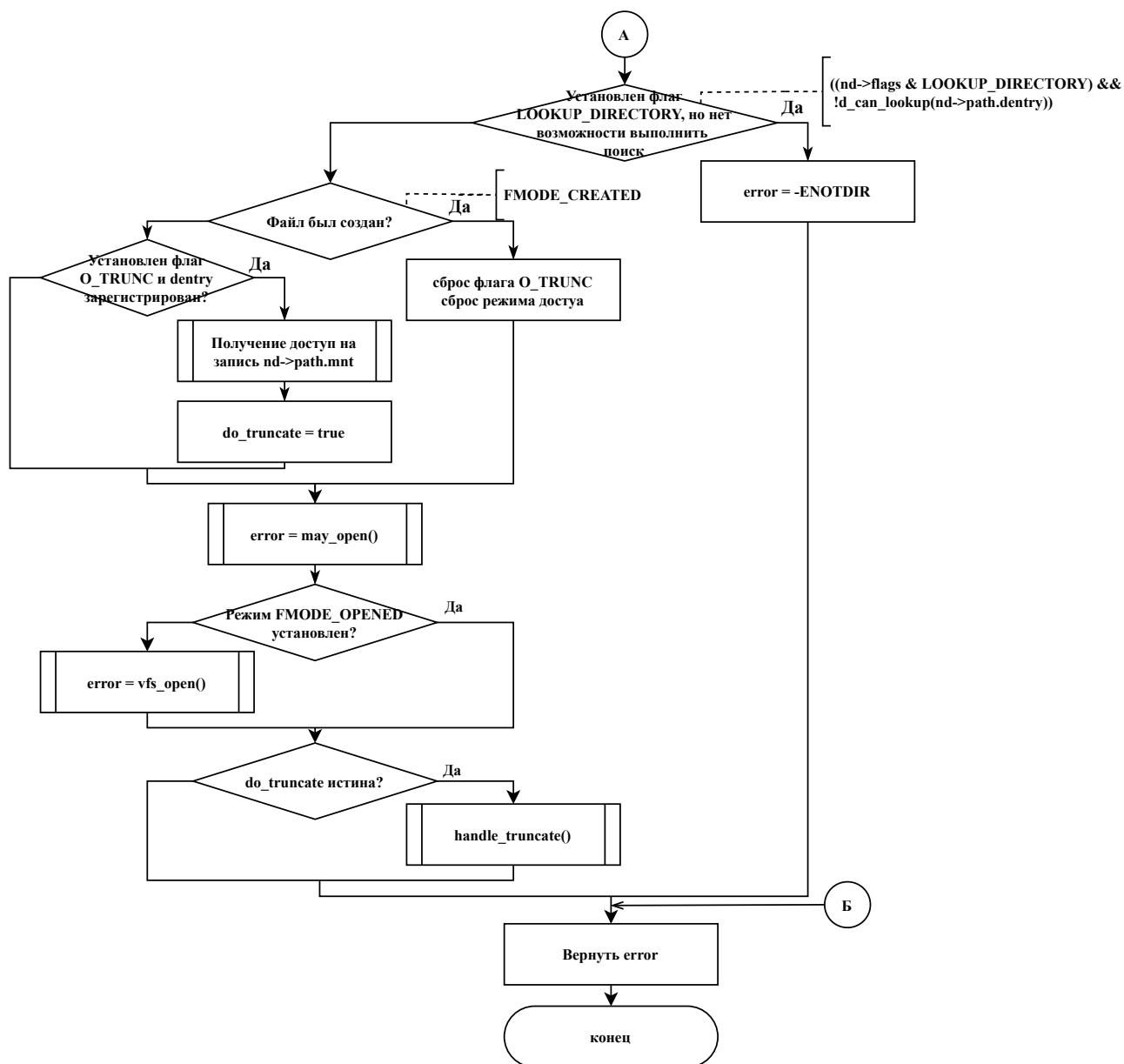


Рисунок 11 – do\_open()