



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»

НАПРАВЛЕНИЕ ПОДГОТОВКИ \_\_\_\_\_ «09.03.04 Программная инженерия»

## ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №7

Название: \_\_\_\_\_ Ввод вывод

Дисциплина: \_\_\_\_\_ Операционные системы

Студент	<u>ИУ7-66Б</u>	_____	<u>А.Д. Ковель</u>
	Группа	Подпись, дата	И. О. Фамилия

Преподаватель	_____	<u>Н. Ю. Рязанова</u>
	Подпись, дата	И. О. Фамилия

Москва, 2023 г.

# ИСПОЛЬЗУЕМЫЕ СТРУКТУРЫ

Листинг 1 – 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 2 – 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 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;
```

# СХЕМЫ АЛГОРИТМОВ

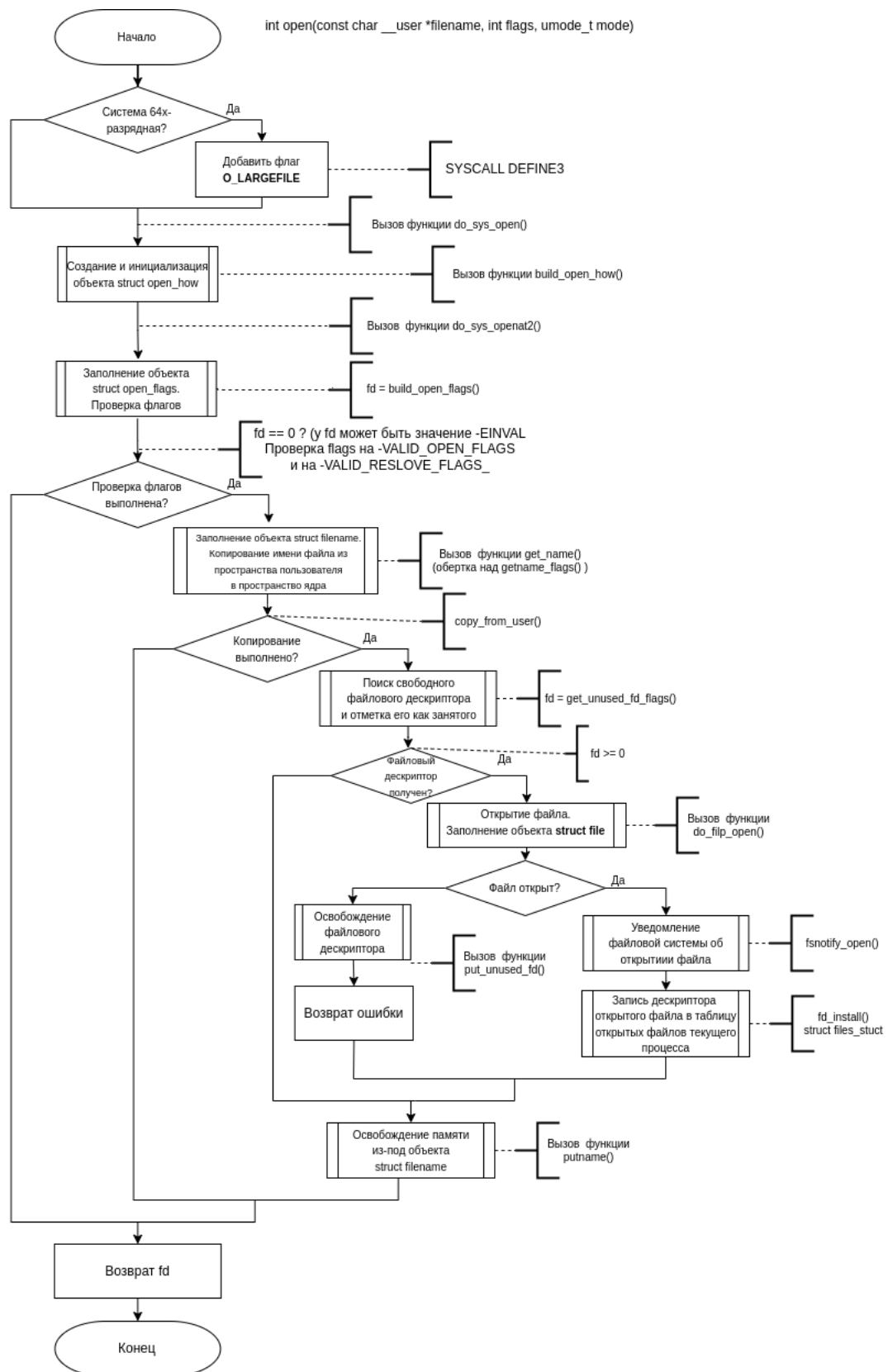


Рисунок 1 – Схема алгоритма работы системного вызова open()

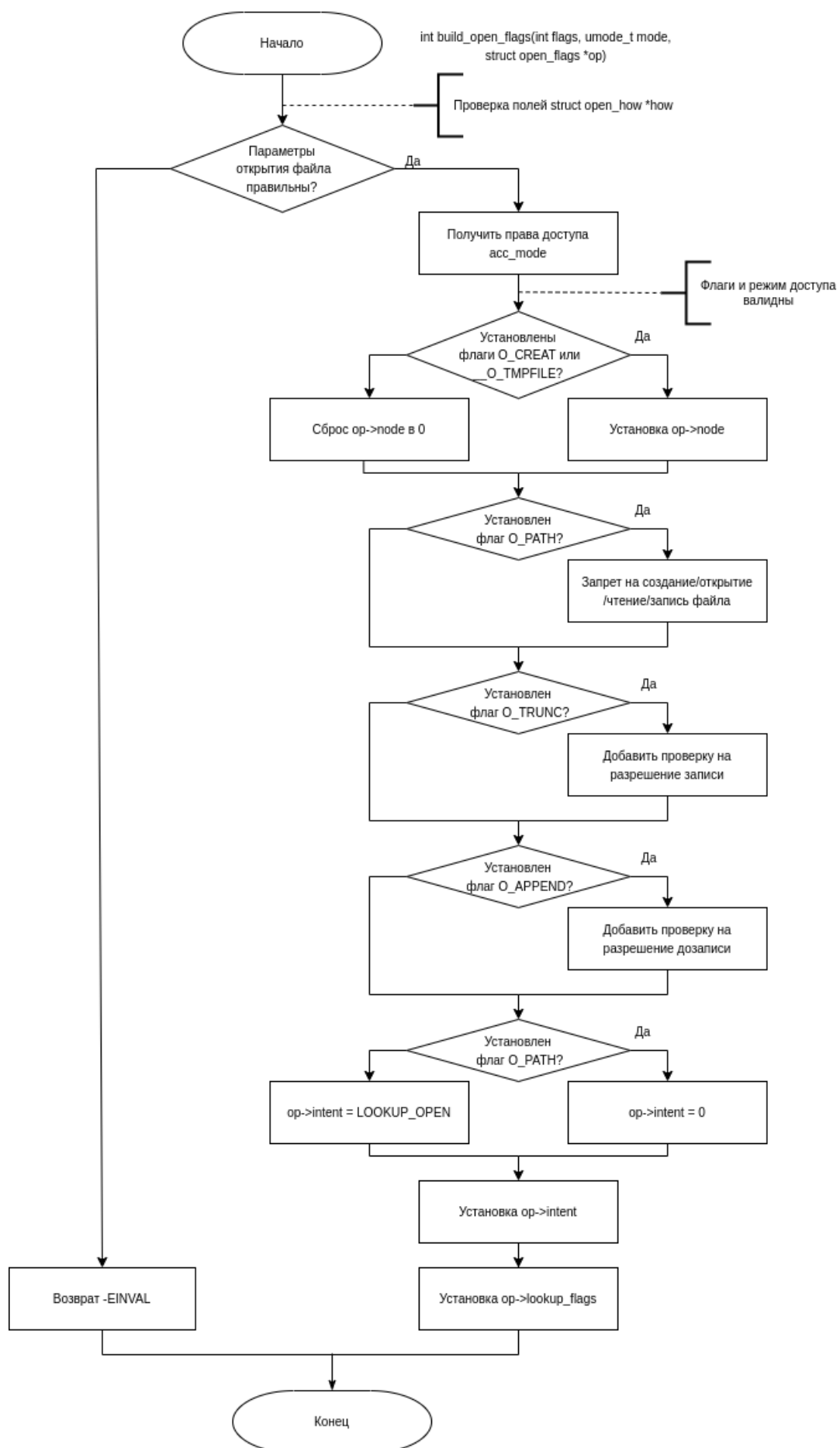


Рисунок 2 – Схема алгоритма работы функции build\_open\_flags()

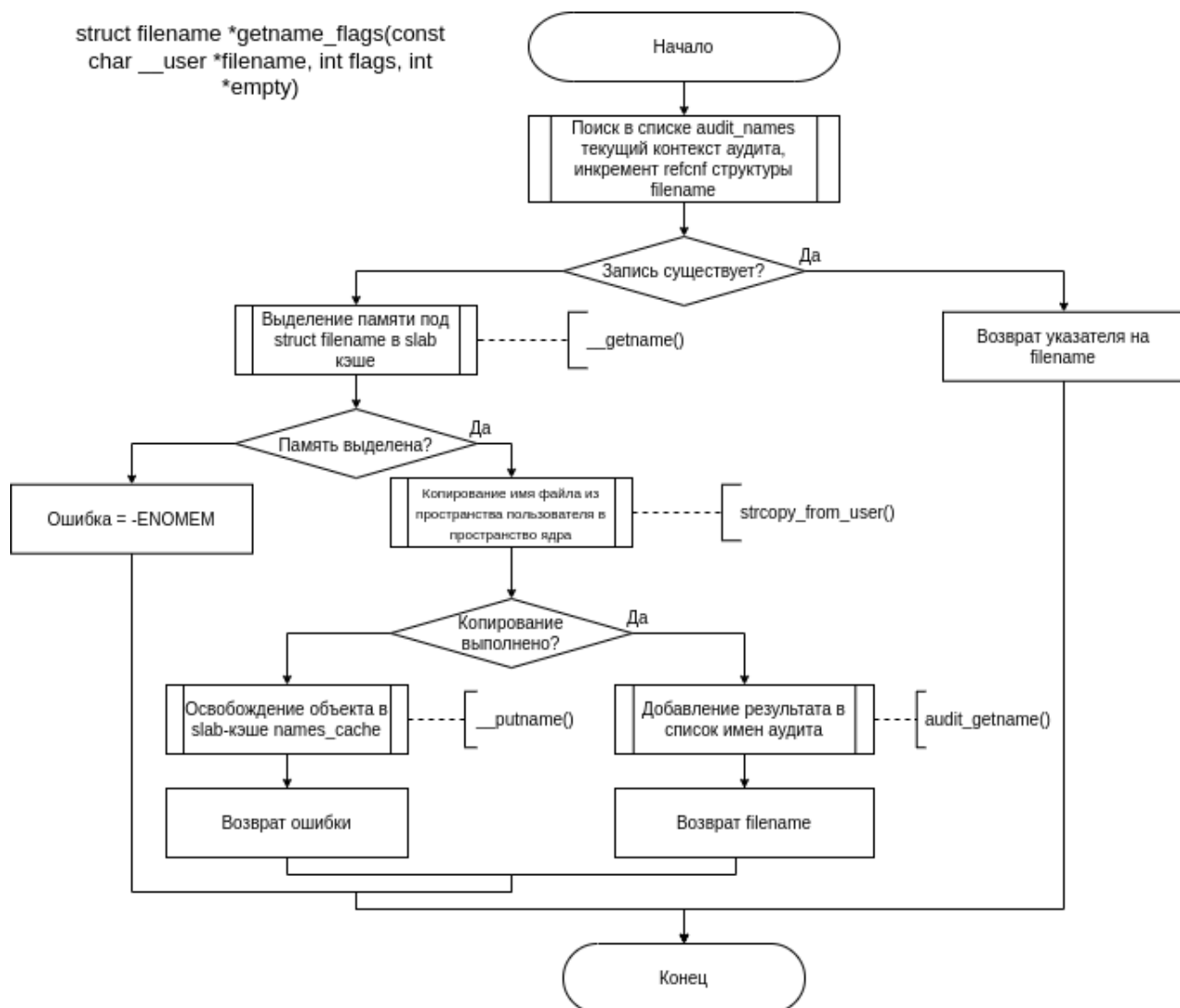


Рисунок 3 – Схема алгоритма работы функции getname\_flags()

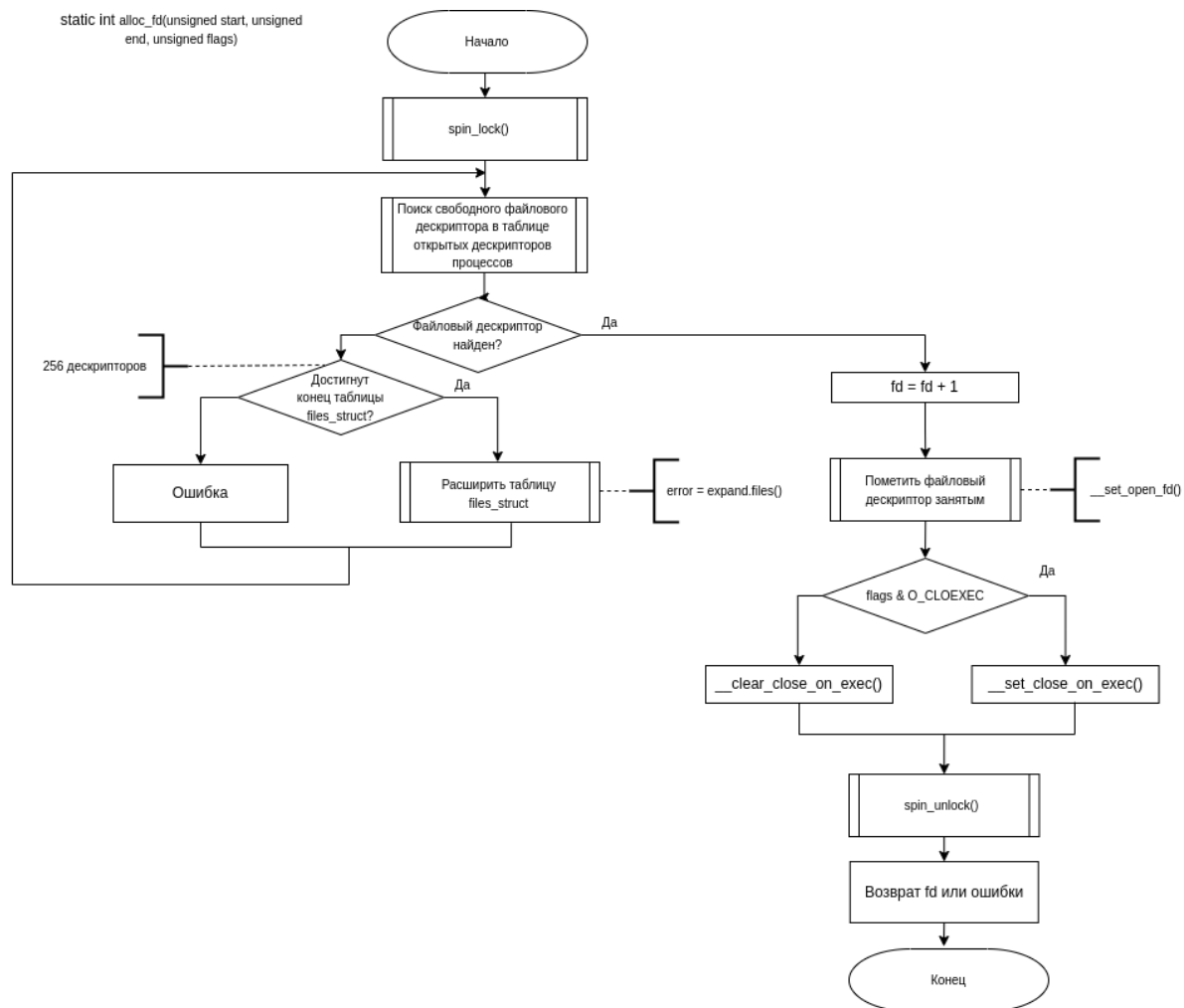


Рисунок 4 – Схема алгоритма работы функции alloc\_fd()

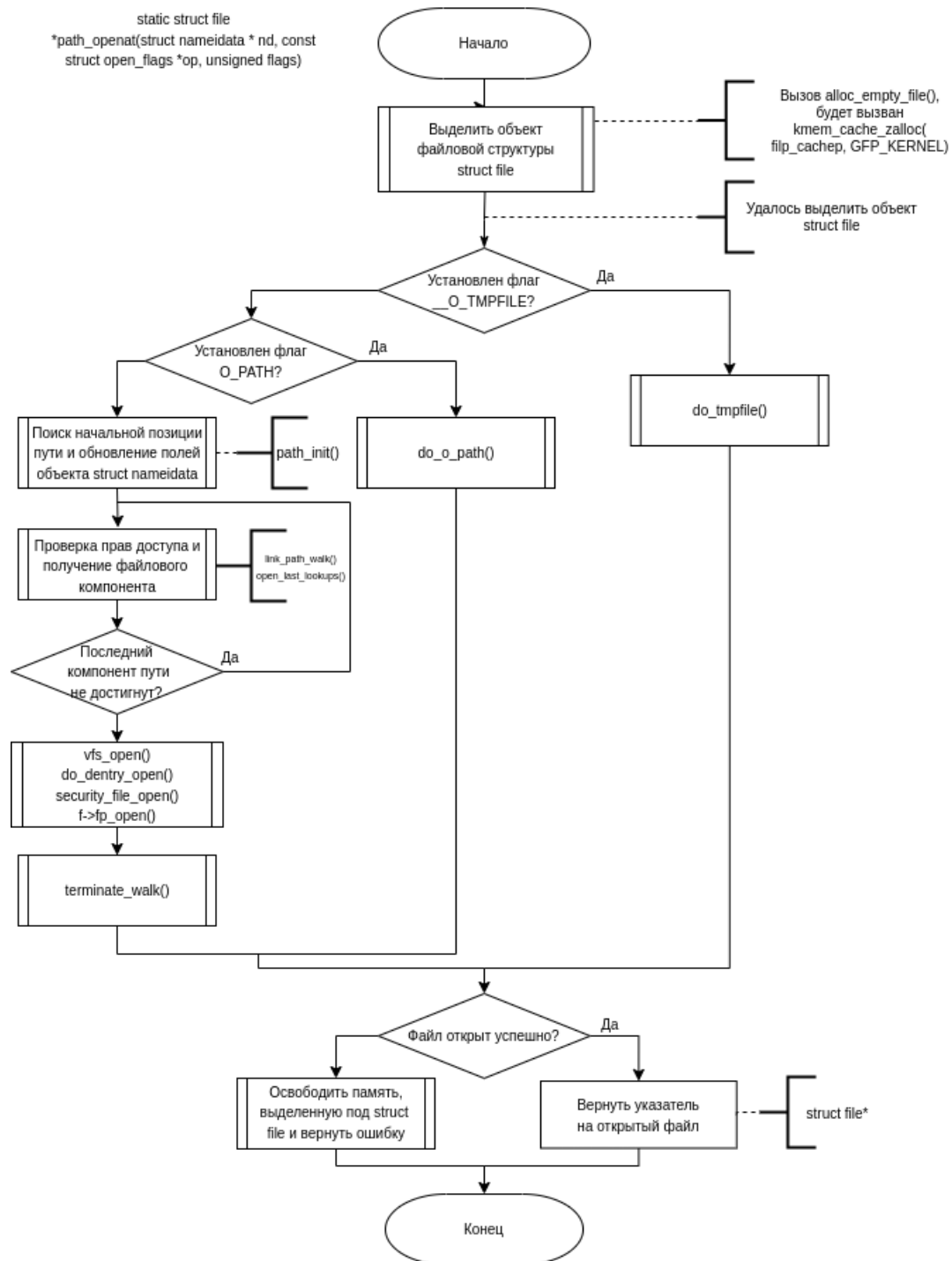


Рисунок 5 – Схема алгоритма функции path\_openat()

```
struct file *do_filp_open(int dfd, struct filename
*pathname, const struct open_flags *op)
```

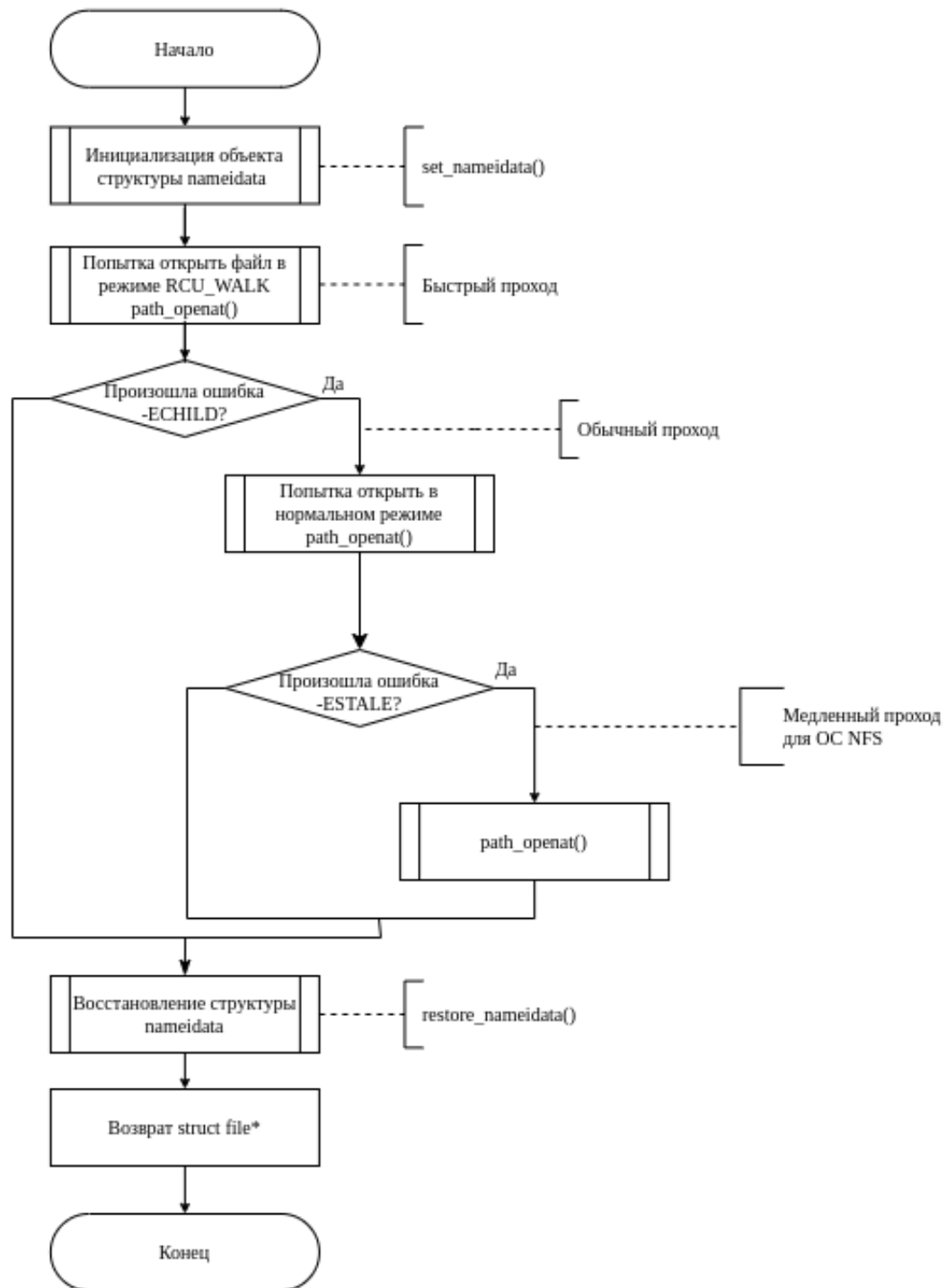


Рисунок 6 – Схема алгоритмов функций, работающих с nameidata (do filp open)



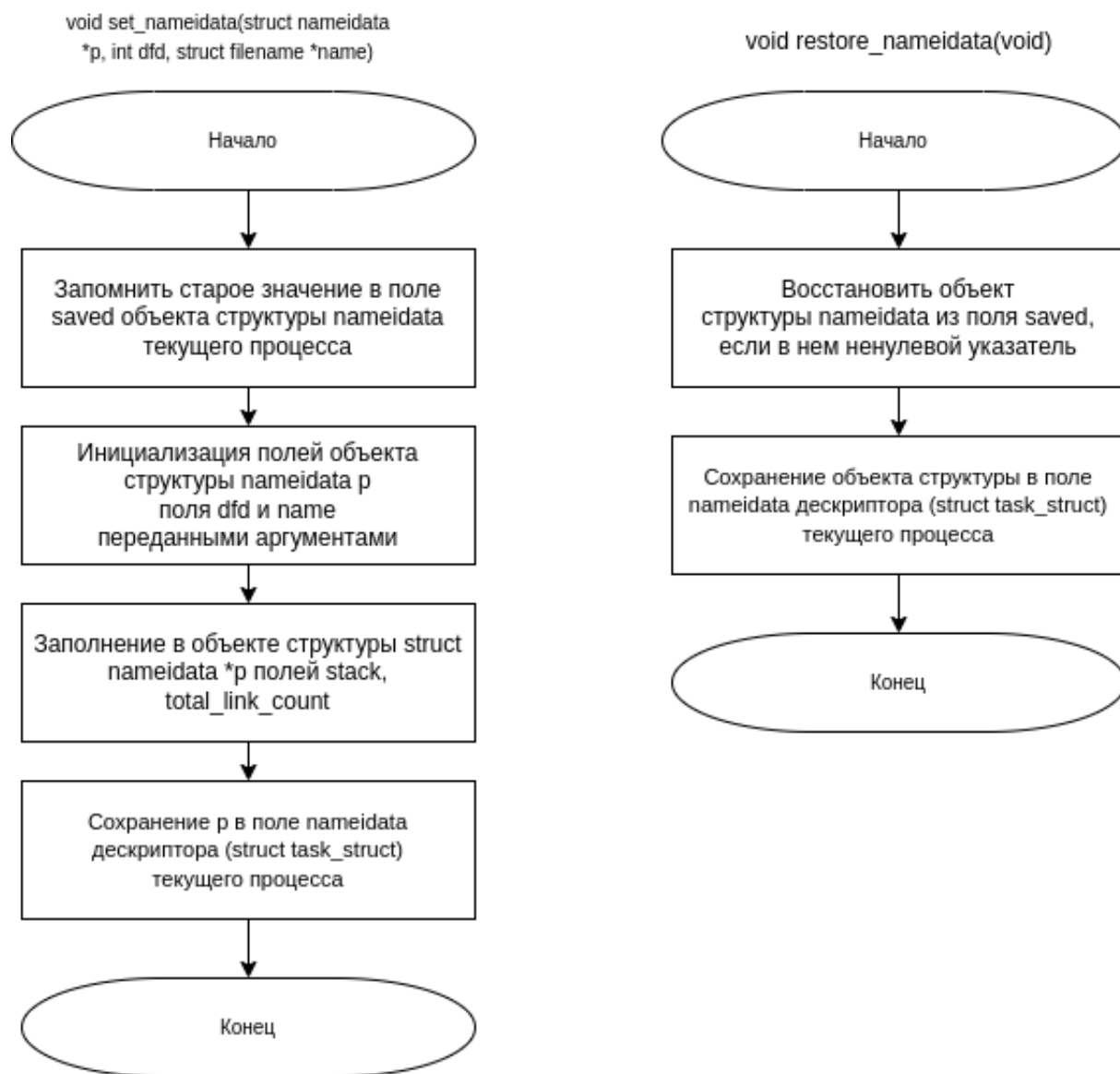


Рисунок 7 – Схема алгоритмов функций, работающих с nameidata

LOOKUP\_RCU — флаг для открытия файла в режиме RCU\_walk (Допускает возможность одновременного доступа).

LOOKUP\_REVAL — флаг для ФС NFS O\_APPEND.

O\_APPEND может проводить к потере данных файлов в ФС NFS, если одновременно добавл. данные нескольких процессов. Нельзя избежать ускорение гонки.

NFS не поддерживает добавление в файл, потому клиентское ядро имитирует такое поведение.

```
static struct dentry *lookup_open( struct
nameidata *ndm struct file *file, const
struct open_flags *op, bool got_write)
```

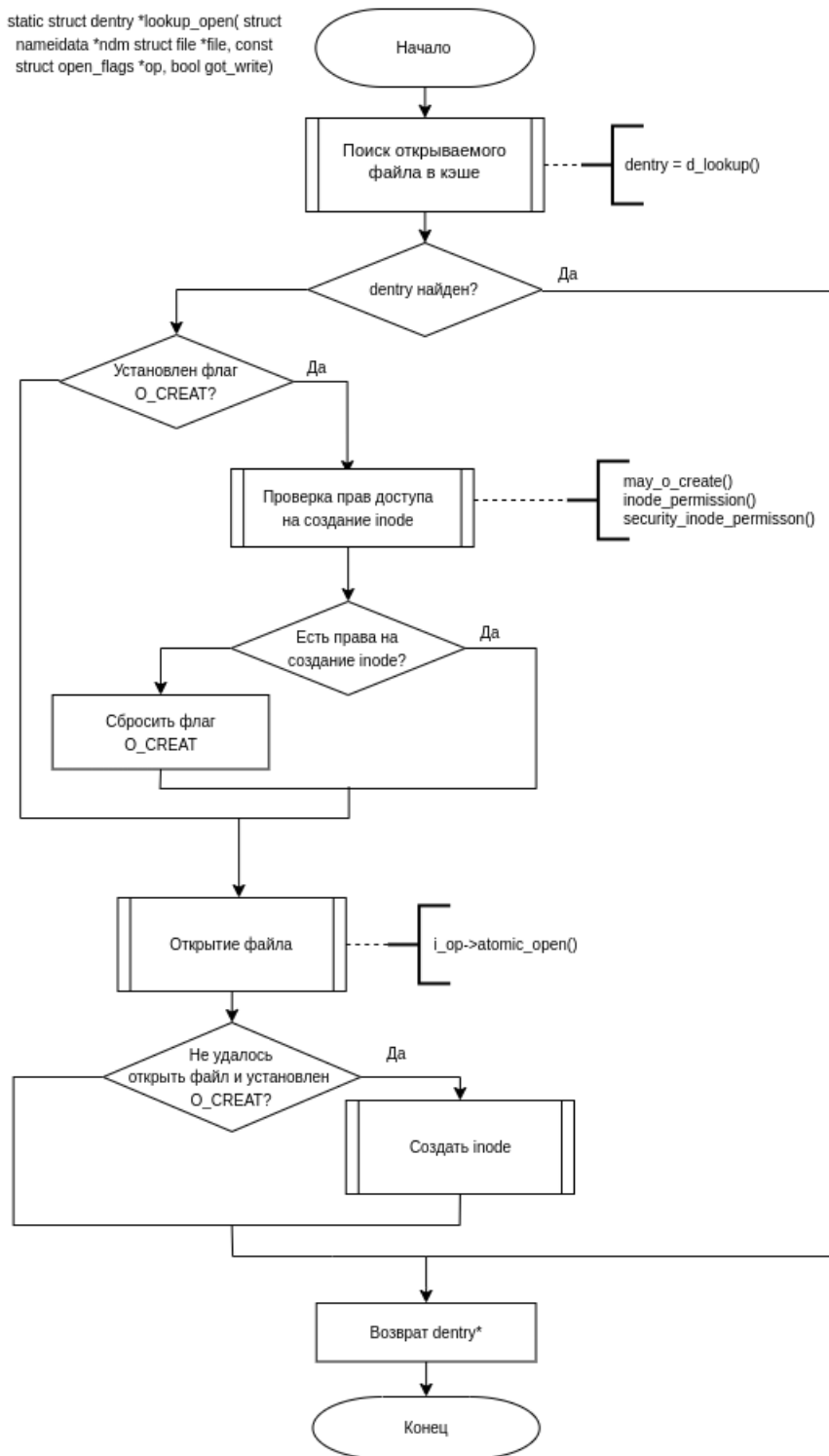


Рисунок 8 – Схема алгоритма функции open\_last\_lookups()

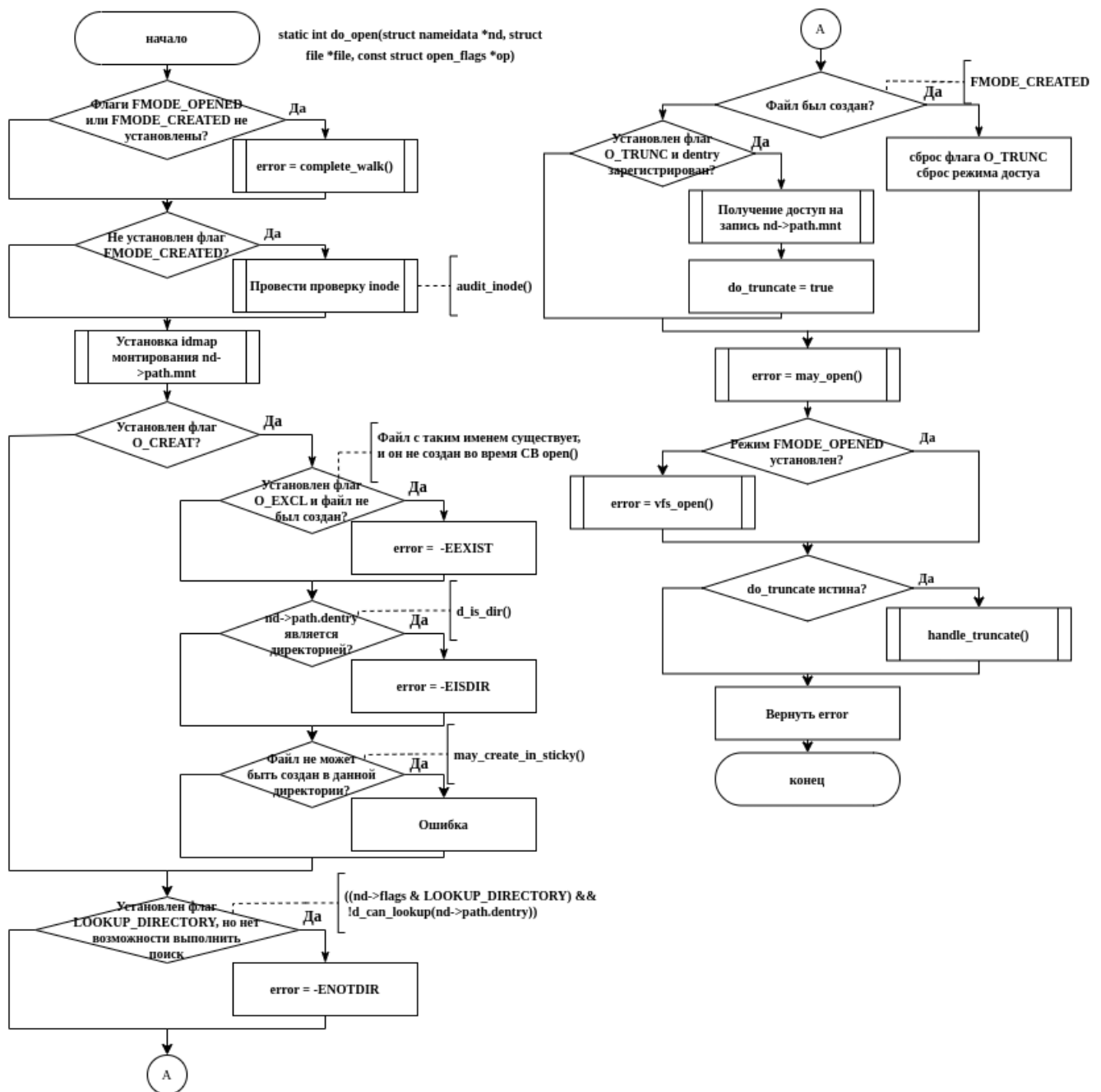


Рисунок 9 – Схема алгоритма функции do\_open()

```
static struct dentry *lookup_open( struct
nameidata *ndm, struct file *file, const
struct open_flags *op, bool got_write)
```

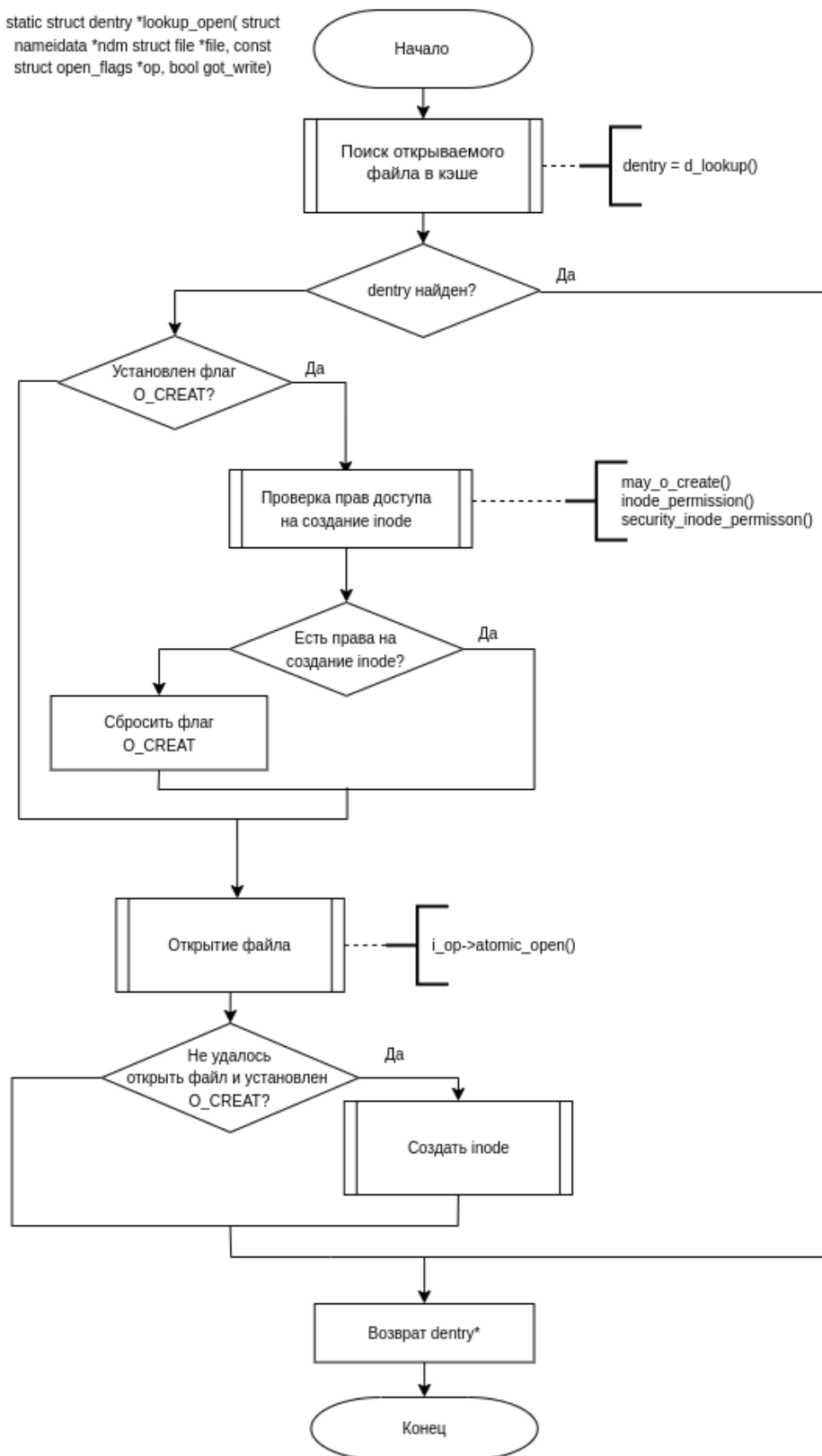


Рисунок 10 – Схема алгоритма функции open\_lookup()

static int open\_last\_lookups(struct nameidata \*nd, struct file \*file, const struct open\_plugs \*op)

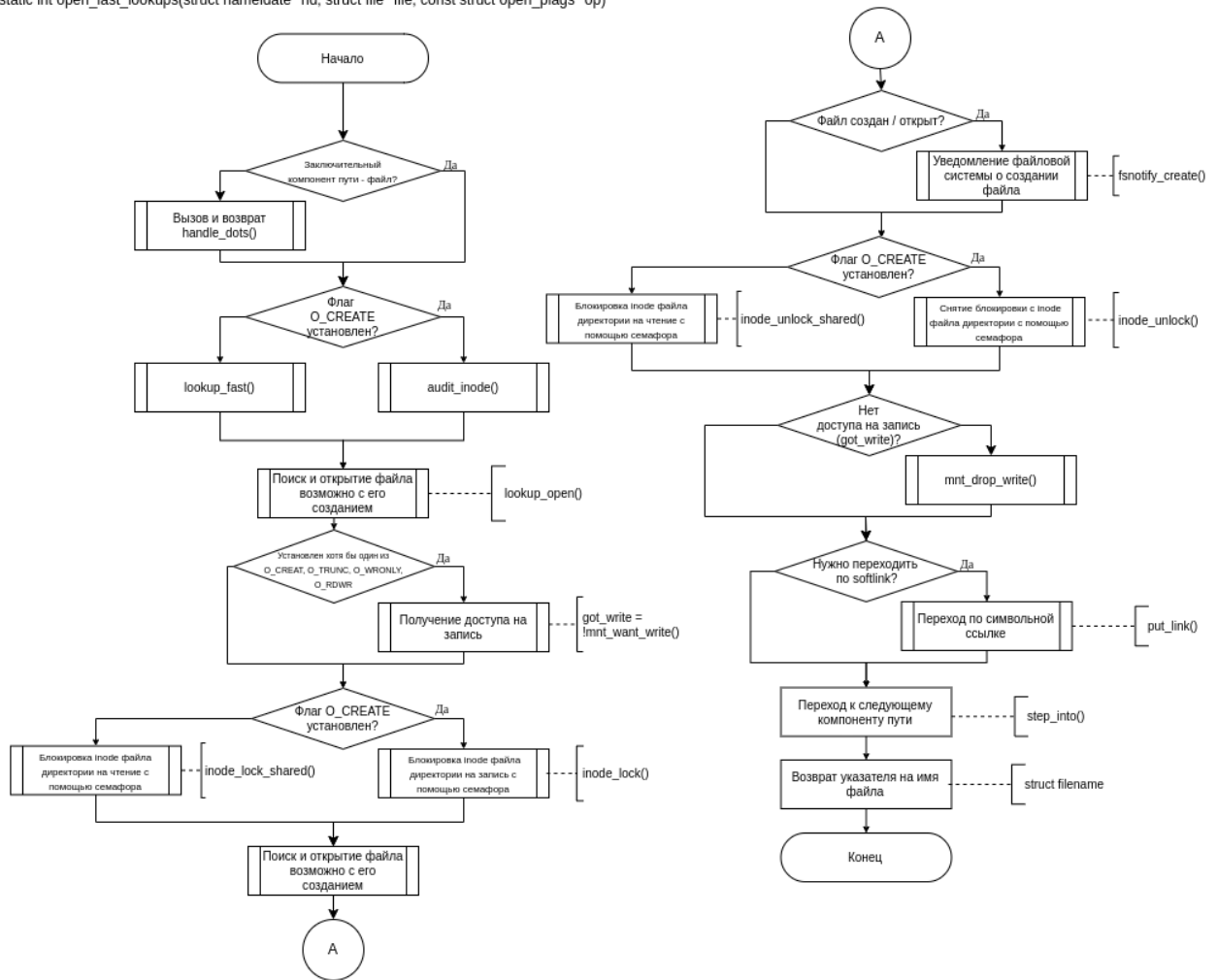


Рисунок 11 – Схема алгоритма функции last\_lookup()