**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»**

**Тема** Системный вызов open()

**Студент** Климов И.С.

**Группа** ИУ7-62Б

**Оценка (баллы)**

**Преподаватели** Рязанова Н.Ю.

Москва — 2022 г.

## **Используемые структуры**

Используемая операционная система – Linux 5.13.0

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** iname[];

};

Структура **open\_flags**

**struct** open\_flags {

**int** open\_flag;

**umode\_t** mode;

**int** acc\_mode;

**int** intent;

**int** lookup\_flags;

};

Структура **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.
* *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 алгоритмов**

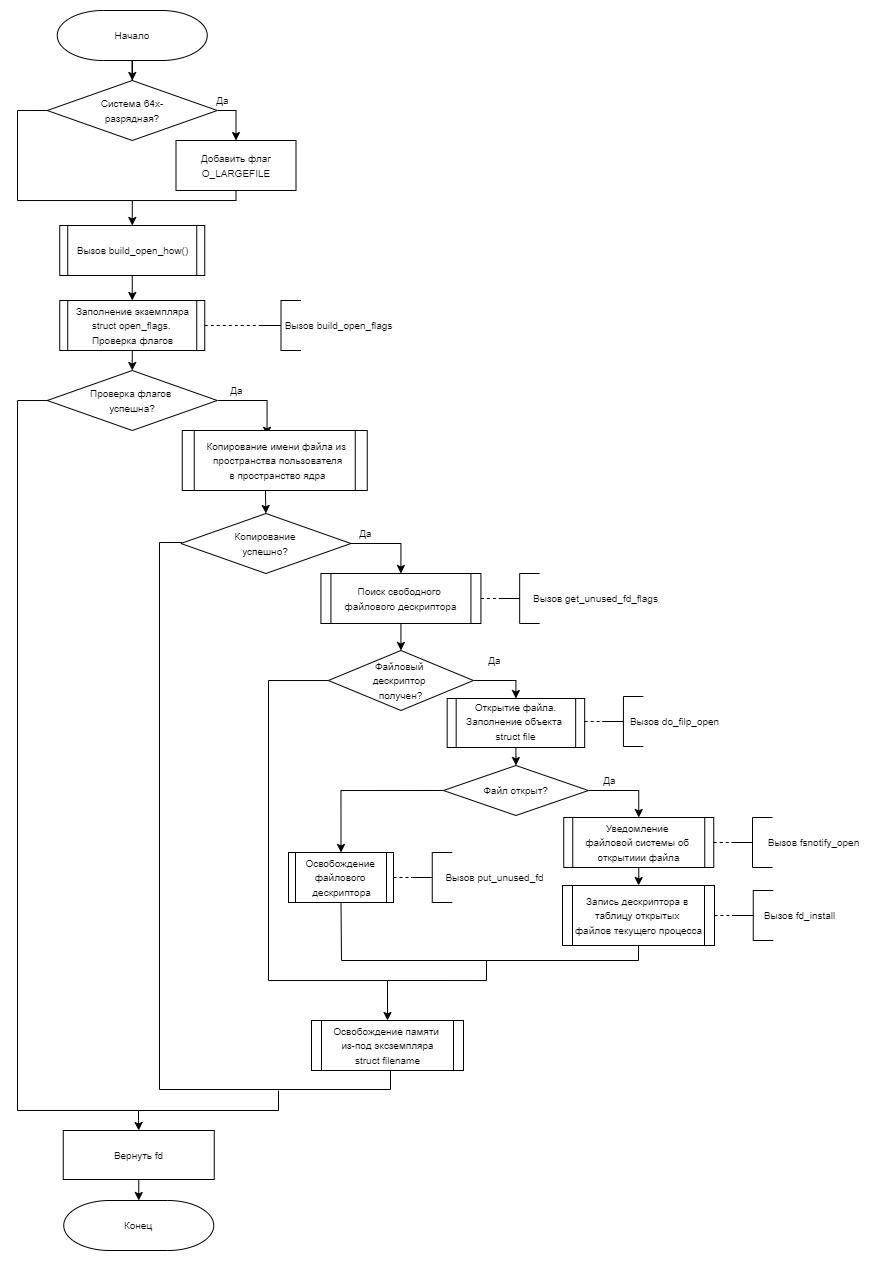


Рисунок 1 – Алгоритм работы функции open

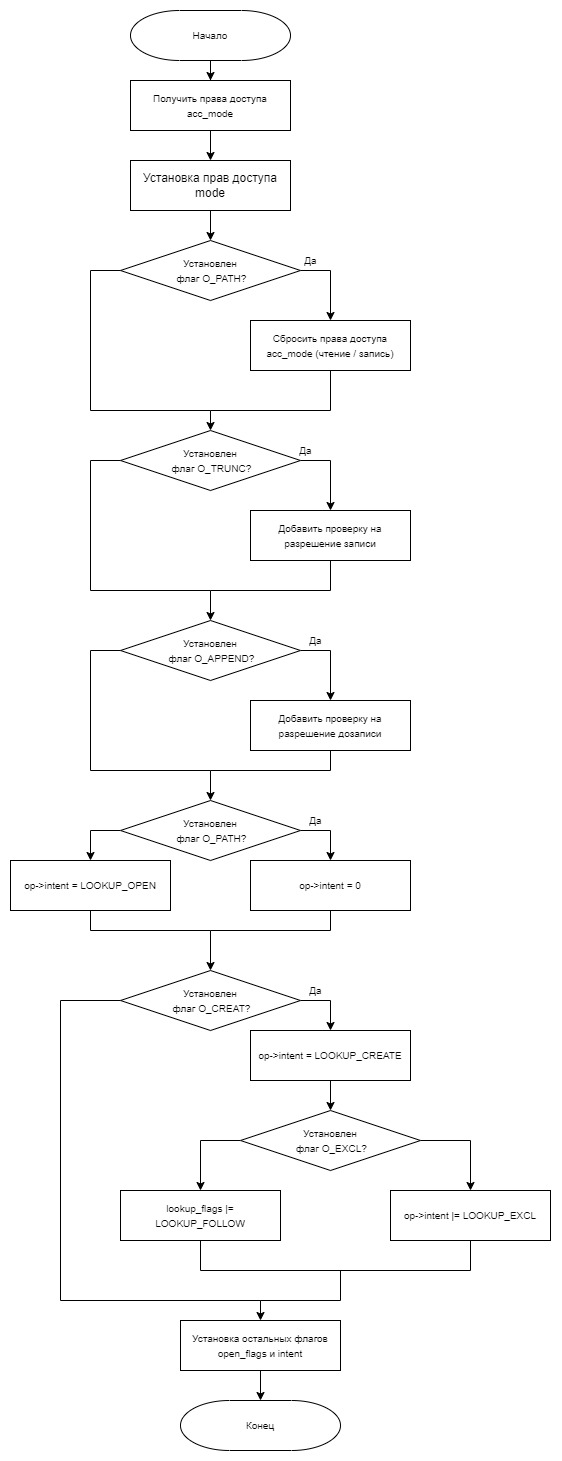


Рисунок 2 – Алгоритм работы функции build\_open\_flags

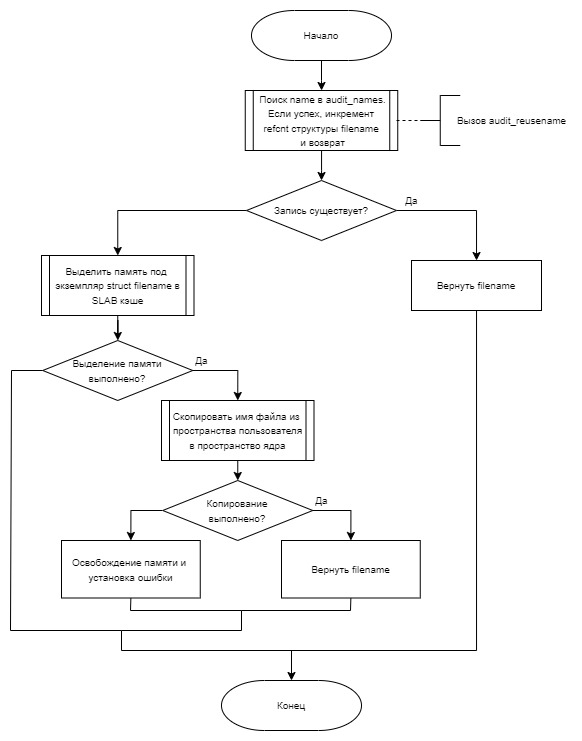


Рисунок 3 – Алгоритм работы функции getname\_flags

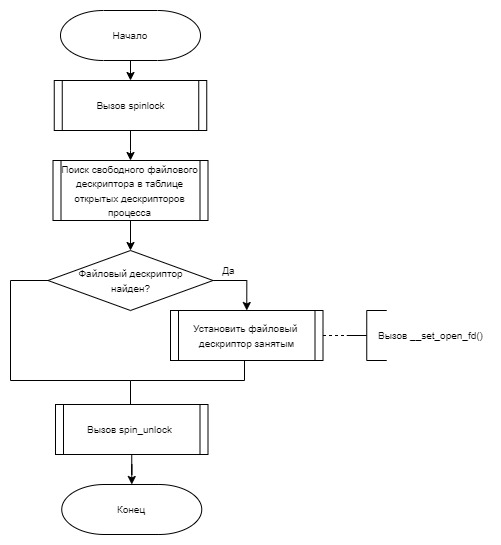


Рисунок 4 – Алгоритм работы функции alloc\_fd

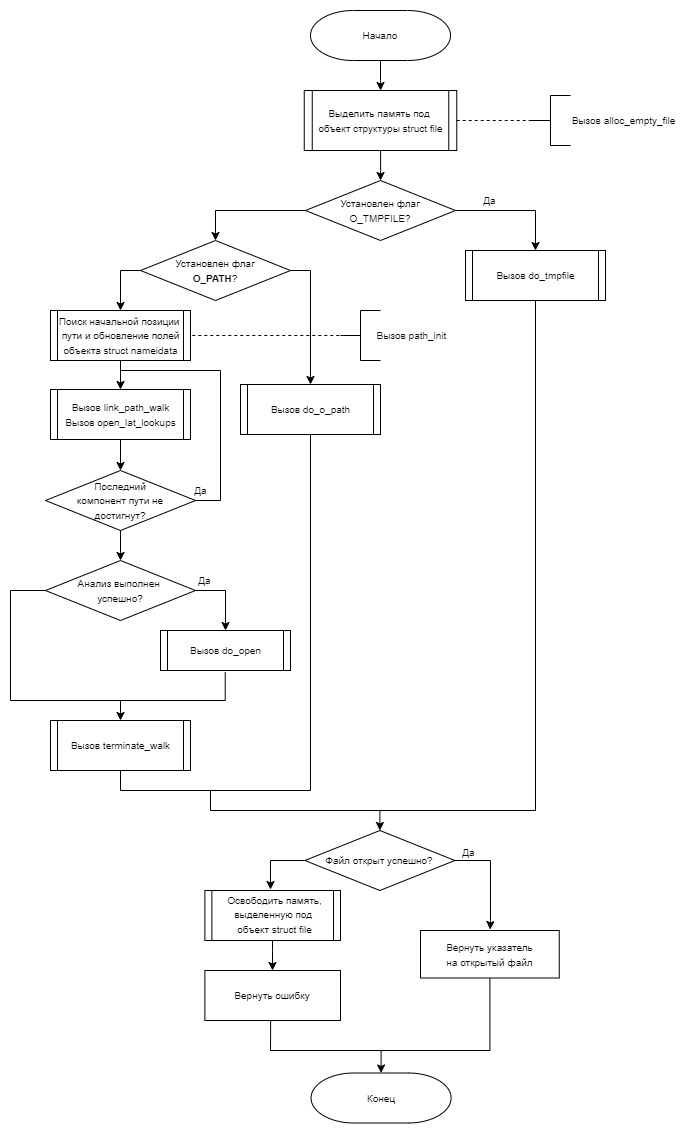


Рисунок 5 – Алгоритм работы функции path\_openat

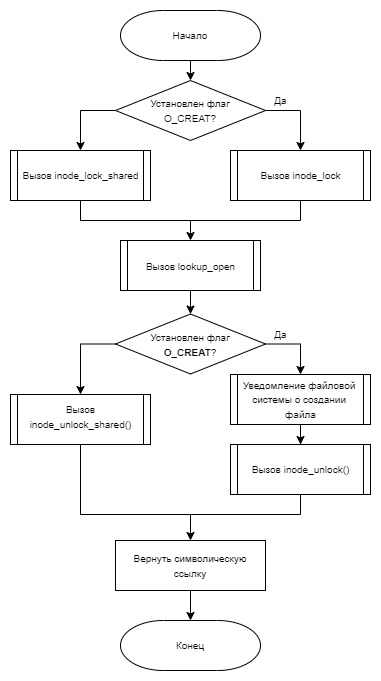


Рисунок 6 – Алгоритм работы функции open\_last\_lookups

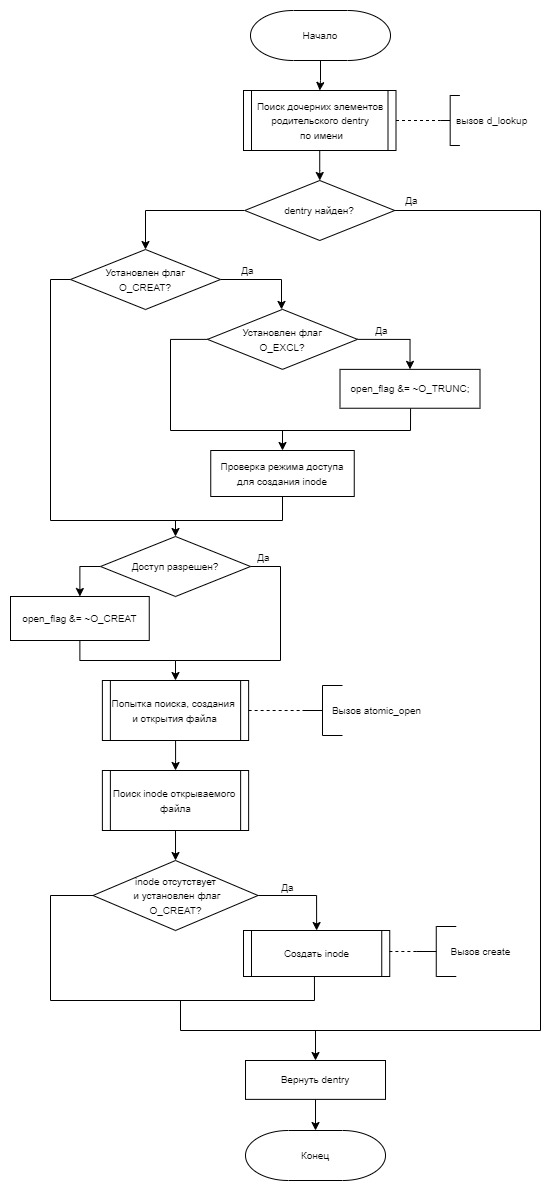


Рисунок 7 – Алгоритм работы функции lookup\_open

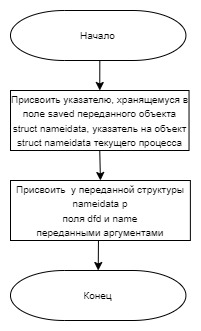


Рисунок 8 – Алгоритм работы функции set\_nameidata

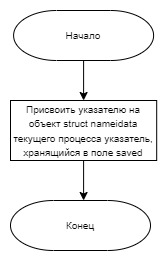


Рисунок 9 – Алгоритм работы функции restore\_nameidata