



# Отчёт по лабораторной работе № 1 по курсу ОС

Студент группы 206 Мигалев Р.П., № по списку 11

Контакты www, e-mail, icq, skype \_\_\_\_\_

Работа выполнена: « \_\_\_\_\_ » 201 \_\_\_\_ г.

Преподаватель: \_\_\_\_\_ каф.806 \_\_\_\_\_

Входной контроль знаний с оценкой \_\_\_\_\_

Отчёт сдан « \_\_\_\_\_ » 201 \_\_\_\_ г., итоговая оценка \_\_\_\_\_

Подпись преподавателя \_\_\_\_\_

1. Тема: Работа с системными вызовами

2. Цель работы: Приобретение практических навыков диагностики работы ПО

☐ 3. Задание ( вариант № ): Написать программу, демонстрирующую работу с различными системными вызовами

4. Оборудование(лабораторное):

ЭВМ \_\_\_\_\_, процессор \_\_\_\_\_, имя узла сети \_\_\_\_\_ с ОП \_\_\_\_\_ Мб,

НМД \_\_\_\_\_ Мб. Терминал \_\_\_\_\_ адрес \_\_\_\_\_ . Принтер \_\_\_\_\_

Другие устройства \_\_\_\_\_

*Оборудование ПЭВМ студента, если использовалось:*

Процессор \_\_\_\_\_ с ОП \_\_\_\_\_ Мб, НМД \_\_\_\_\_ Мб. Монитор \_\_\_\_\_

Другие устройства \_\_\_\_\_

5. Программное обеспечение(лабораторное):

Операционная система семейства \_\_\_\_\_, наименование \_\_\_\_\_ версия \_\_\_\_\_

интерпретатор команд \_\_\_\_\_ версия \_\_\_\_\_

Система программирования \_\_\_\_\_ версия \_\_\_\_\_

Редактор текстов \_\_\_\_\_ версия \_\_\_\_\_

Утилиты операционной системы \_\_\_\_\_

Прикладные системы и программы \_\_\_\_\_

Местонахождение и имена файлов программ и данных \_\_\_\_\_

*Программное обеспечение ЭВМ студента, если использовалось:*

Операционная система семейства Linux, наименование Kubuntu версия 16.04

интерпретатор команд gcc версия 5.4.0

Система программирования \_\_\_\_\_ версия \_\_\_\_\_

Редактор текстов \_\_\_\_\_ версия \_\_\_\_\_

Утилиты операционной системы \_\_\_\_\_

Прикладные системы и программы \_\_\_\_\_

Местонахождение и имена файлов программ и данных на домашнем компьютере \_\_\_\_\_

6. **Идея, метод, алгоритм** решения задачи (в формах: словесной, псевдокода, графической [блок-схема, диаграмма, рисунок, таблица] или формальные спецификации с пред- и постусловиями)

Рассмотреть различные системные вызовы (обращения прикладной программы к ядру операционной системы для выполнения какой-либо операции) ОС Linux.

Для выполнения межпроцессорной операции или операции, требующей доступа к оборудованию, программа обращается к ядру, которое, в зависимости от полномочий вызывающего процесса, исполняет либо отказывает в таком вызове.

Архитектура современных процессоров предусматривает использование защищенного режима с несколькими уровнями привилегий: приложения ограничены своим адресным пространством, следовательно не могут получить доступ или модифицировать другие приложения или саму ОС.

Таким образом, процессор обеспечивает безопасность выполнения программ, "общаясь" с ними через системные вызовы и предоставляя доступные ресурсы.

7. **Сценарий выполнения работы** [план работы, первоначальный текст программы в черновике (можно на отдельном листе) и тесты либо соображения по тестированию].

(по порядку появления в коде программы)

1. `int exit(int status)` - завершает текущий процесс
2. `int getpid(void)` - получить идентификатор процесса
3. `int chdir(const char * filename)` - сменить рабочую директорию
4. `int creat(const char * pathname, int mode)` - создать файл или девайс
5. `int open(const char * filename, int flags, int mode)` - открыть (и, возможно, создать) файл или девайс
6. `ssize_t write(unsigned int fd, const char * buf, size_t count)` - записать в файловый дескриптор
7. `off_t lseek(unsigned int fd, off_t offset, unsigned int origin)` - сменить отступ указателя на текущее положение в файле.
8. `ssize_t read(unsigned int fd, char * buf, size_t count)` - читать из файлового дескриптора
9. `int fstat(unsigned int fd, struct stat * buf)` - возвращает информацию об открытом файле, на который указывает файловый дескриптор.
10. `int close(unsigned int fd)` - закрыть файловый дескриптор
11. `int time(int * floc)` - получить системное время в секундах
12. `int dup(int oldfd)` - создает копию файлового дескриптора

Пункты 1-7 отчета составляются **строго до** начала лабораторной работы.

Допущен к выполнению работы. Подпись преподавателя \_\_\_\_\_

- 8. Распечатка протокола** (подклеить листинг окончательного варианта программы с тестовыми примерами, подписанный преподавателем).

[https://github.com/darkloned/labs/tree/master/C2S3/Operating\\_Systems/lab1](https://github.com/darkloned/labs/tree/master/C2S3/Operating_Systems/lab1)

9. **Дневник отладки** должен содержать дату и время сеансов отладки и основные события (ошибки в сценарии и программе, нестандартные ситуации) и краткие комментарии к ним. В дневнике отладки приводятся сведения об использовании других ЭВМ, существенном участии преподавателя и других лиц в написании и отладке программы.

№	Лаб. или дом.	Дата	Время	Событие	Действие по исправлению	Примечание

10. **Замечания автора** по существу работы \_\_\_\_\_

#### 11. Выводы

В результате выполнения данной лабораторной работы было получено общее представление о взаимодействии программного и аппаратного обеспечения на низком уровне посредством системных вызовов - обращений к ядру ОС. Была создана и отлажена тестовая программа, демонстрирующая функционал набора подобных вызовов.

Недочёты при выполнении задания могут быть устранены следующим образом: \_\_\_\_\_

Подпись студента \_\_\_\_\_