

# Лабораторная работа 1

## Знакомство с операционной системой UNIX

### Порядок выполнения работы

1. Загрузить операционную систему (ОС) семейства UNIX. Запустить эмулятор терминала (Konsole, GNOME Terminal и т. п.) или перейти в 1-й терминал по нажатию клавиш <Ctrl>+<Alt>+<F1>.
2. Выполнить команду, выводящую на экран полное имя текущей рабочей директории (обычно, при запуске терминала, это домашняя директория пользователя).
3. Вывести содержимое текущей директории, включая скрытые объекты (файлы и директории), с указанием параметров объектов (тип, права доступа, количество ссылок, имя владельца, имя группы, размер (в байтах), временной штамп и имя объекта).
4. Выполнить команду, которая позволит изменить текущую рабочую директорию на корневую. Затем, вернуть в качестве рабочей директории домашнюю. В обоих случаях указывать абсолютный (полный) путь к директориям. В правильности смены директорий убедиться, выводя каждый раз на экран полное имя текущей рабочей директории.
5. Выполнить команду `cd ../..`, а затем `cd` без параметров, просматривая после выполнения каждой команды полное имя текущей директории. Сравнить с результатами выполнения п. 4, сделав соответствующие выводы, в том числе о значении директории, имя которой состоит из двух точек.
6. Выполнить команду `cd .` (параметром команды `cd` является точка). Как изменилась рабочая директория? Сделать вывод о значении директории с именем, состоящим из одной точки.
7. Используя команду конкатенации файлов, в текущей (домашней) директории создать файл с именем `file1`, в который в качестве содержимого записать строку 'Nihil sine labore!', введенную с клавиатуры. Убедиться в правильности, выполнив команду, выводящую содержимое файла `file1` на экран.
8. Используя механизм перенаправления данных, в текущей директории создать `file2`, содержащий имена (без дополнительной информации о типе, правах доступа и т. п.) нескрытых объектов текущей директории. Убедиться в правильности, выполнив команду, выводящую содержимое файла `file2` на экран.
9. Создать в текущей директории файл с именем `file3` и содержимым, являющимся результатом конкатенации информации из файлов `file1` и `file2`. Убедиться в правильности, выполнив команду, выводящую содержимое файла `file3` на экран.
10. Внутри текущей директории создать директории с именами `dir1`, `dir2` и `dir3`. Убедиться в правильности создания, выведя содержимое текущей директории на экран.
11. Создать копию файла `file1` с именем `file4`. Скопировать файлы `file1`, `file2`, `file3`, `file4` в директорию `dir1` одной командой без использования шаблона имен. Убедиться в успешности копирования, выведя содержимое директории `dir1` на экран (текущую рабочую директорию не изменять).

12. Переместить файлы `file1`, `file2`, `file3`, `file4` из текущей директории в директорию `dir2` одной командой с использованием шаблона имен. Переименовать перемещенный файл `file2` в `abc.txt`. Убедиться в успешности перемещения, выведя содержимое директории `dir2` на экран (текущую рабочую директорию не изменять).

13. Скопировать директории `dir1` и `dir2` вместе с их содержимым внутрь директории `dir3`. Убедиться в успешности копирования, выведя содержимое директории `dir3` на экран (текущую рабочую директорию не изменять).

14. Выполнить команду, позволяющую удалить файлы `file1`, `file3` из директории `dir1`, используя при этом шаблон имен. Убедиться в успешности удаления, выведя содержимое директории `dir1` на экран (текущую рабочую директорию не изменять).

15. Выполнить команду, которая удалит все содержимое директории `dir2`. Убедиться в успешности удаления, выведя содержимое директории `dir2` на экран (текущую рабочую директорию не изменять).

16. Используя шаблон имен, удалить директории `dir1`, `dir2` и `dir3` из текущей. Убедиться в правильности создания, выведя содержимое текущей директории на экран.

17. Используя команду `id` вывести информацию о текущем пользователе: его имя, идентификатор, названия и идентификаторы группы по умолчанию и дополнительных групп. Аналогичную информацию вывести для пользователя `root`, выполнив команду `id root`.

18. В текущей директории создать файл с именем `text`, содержащий строку 'Experientia est optima magistra.'. Выполнить команду, позволяющую получить для созданного файла информацию о правах доступа.

19. Вывести содержимое файла `text` на экран. Выполнить команду `chmod u-r text`. Попытайтесь вывести на экран информацию из файла. Сделать вывод о результате выполнения команды `chmod u-r text`, предварительно получив для файла `text` информацию об измененных правах доступа.

20. Выполнить команду `chmod 642 text`. Для файла `text` вывести на экран информацию о правах доступа. Сделать вывод о том, как при выполнении `chmod 642 text` изменились права доступа для каждой из трех категорий пользователей. Какие действия теперь могут и не могут совершать пользователи каждой категории с файлом?

21. Создать файл `lab1_1.c`, содержащий текст программы выводящей на экран при помощи системных вызовов `getuid()` и `getgid()` идентификаторы пользователя и группы по умолчанию для пользователя. Разрабатывать текст программы возможно с использованием удобных графических текстовых редакторов (`gedit`, `kwrite` и т. п.).

22. Скомпилировать текст разработанной программы в исполняемый файл с именем `lab1_1`.

23. Запустив программу из файла `lab1_1` на выполнение, получить идентификатор и группу текущего пользователя.

24. Используя команду `su`, выполнить программу из файла `lab1_1` от имени пользователя `root`, таким образом, получить идентификатор и группу пользователя `root`.

### ***Содержание отчета***

1. Команды, которые позволяют выполнить действия, описанные в пп. 2—20 выполнения работы, а также результаты выполнения, подтверждаемые копиями экранов терминала.
2. Текст программы по п. 21 выполнения работы..
3. Результаты запусков разработанной программы по пп. 23, 24 выполнения работы, подтверждаемые копиями экранов терминала.
4. Вывод.

## **Лабораторная работа 2**

### **Процессы в операционной системе UNIX**

#### ***Порядок выполнения работы***

1. Изучить теоретические сведения.
2. Откомпилировать программы, текст которых приведен в методических указаниях. Проверить правильность их работы.
3. Разработать программы, предназначенные для самостоятельной подготовки. Откомпилировать, проверить правильность их работы.

#### ***Содержание отчета***

1. Текст программы, печатающей значения PID и PPID для текущего процесса. Копия экрана, подтверждающего правильность выполнения программы.
2. Текст программы 03-1.c создания нового процесса с одинаковой работой процессов ребенка и родителя. Копия экрана, подтверждающего правильность выполнения программы.
3. Текст программы создания нового процесса с различной работой процессов ребенка и родителя. Копия экрана, подтверждающего правильность выполнения программы.
4. Текст программы, распечатывающей аргументы командной строки и параметры среды. Копия экрана, подтверждающего правильность выполнения программы.
5. Тест программы 03-2.c, изменяющая пользовательский контекст процесса. Копия экрана, подтверждающего правильность выполнения программы.
6. Текст программы для изменения пользовательского контекста в порожденном процессе. Копия экрана, подтверждающего правильность выполнения программы.
7. Вывод.

# **Лабораторная работа 3**

## **Организация взаимодействия процессов через *pipe* и FIFO в UNIX**

### ***Порядок выполнения работы***

1. Изучить теоретические сведения.
2. Откомпилировать программы, текст которых приведен в методических указаниях. Проверить правильность их работы.
3. Разработать программы, предназначенные для самостоятельной подготовки. Откомпилировать, проверить правильность их работы.

### ***Содержание отчета***

1. Текст программы 05-1.c для записи информации в файл. Копия экрана, подтверждающего правильность выполнения программы.
2. Текст программы для чтения информации из файла. Копия экрана, подтверждающего правильность выполнения программы.
3. Текст программы 05-2.c, иллюстрирующей работу с *pipe* в рамках одного процесса. Копия экрана, подтверждающего правильность выполнения программы.
4. Текст программы 05-3.c, осуществляющей однонаправленную связь через *pipe* между процессом-родителем и процессом-ребенком. Копия экрана, подтверждающего правильность выполнения программы.
5. Текст программы для связи через *pipe* между собой двух родственных процессов, исполняющих разные программы. Копия экрана, подтверждающего правильность выполнения программы.
6. Текст программы, осуществляющей двунаправленную связь через *pipe* между процессом-родителем и процессом-ребенком. Копия экрана, подтверждающего правильность выполнения программы.
7. Текст программы для определения размера *pipe* для операционной системы. Копия экрана, подтверждающего правильность выполнения программы.
8. Текст программы 05-4.c, осуществляющей однонаправленную связь через FIFO между процессом-родителем и процессом-ребенком. Копия экрана, подтверждающего правильность выполнения программы.
9. Два текста программ, одна из которых пишет информацию в FIFO, а вторая – читает из него, так чтобы между ними не было ярко выраженных родственных связей. Копия экрана, подтверждающего правильность выполнения программы.
10. Вывод.

## **Лабораторная работа 4**

### **Средства System V IPC. Организация работы с разделяемой памятью в UNIX. Понятие нитей исполнения (thread)**

#### ***Порядок выполнения работы***

1. Изучить теоретические сведения.
2. Откомпилировать программы, текст которых приведен в методических указаниях. Проверить правильность их работы.
3. Разработать программы, предназначенные для самостоятельной подготовки. Откомпилировать, проверить правильность их работы.

#### ***Содержание отчета***

1. Тексты программ 06-1a.c и 06-1b.c для иллюстрации работы с разделяемой памятью. Копия экрана, подтверждающего правильность выполнения программ.
2. Результаты выполнения в терминале команд `ipcs` и `ipcrm`.
3. Тексты программ, осуществляющие взаимодействие через разделяемую память. Первая программа должна создавать сегмент разделяемой памяти и копировать туда собственный исходный текст, вторая программа должна брать оттуда этот текст, печатать его на экране и удалять сегмент разделяемой памяти из системы. Копия экрана, подтверждающего правильность выполнения программы.
4. Текст программы 06-2.c, в которой работают две нити исполнения. Копия экрана, подтверждающего правильность выполнения программы.
5. Текст программы, в которой работают три нити исполнения. Копия экрана, подтверждающего правильность выполнения программы.
6. Тексты программ 06-3a.c и 06-3b.c для иллюстрации некорректной работы с разделяемой памятью. Копия экрана, подтверждающего правильность выполнения программы.
7. Тексты модифицированных программ для корректной работы с разделяемой памятью при помощи алгоритма Петерсона. Копия экрана, подтверждающего правильность выполнения программы.
8. Вывод.

## **Лабораторная работа 5**

### **Семафоры в UNIX как средство синхронизации процессов**

#### ***Порядок выполнения работы***

1. Изучить теоретические сведения.
2. Откомпилировать программы, текст которых приведен в методических указаниях. Проверить правильность их работы.
3. Разработать программы, предназначенные для самостоятельной подготовки. Откомпилировать, проверить правильность их работы.

#### ***Содержание отчета***

1. Тексты программы 08-1a.c и 08-1b.c для иллюстрации работы с семафорами. Копия экрана, подтверждающего правильность выполнения программ.
2. Тексты программ 08-1a.c и 08-1b.c, измененных так, чтобы первая программа могла работать без блокировки после не менее 5 запусков второй программы. Копия экрана, подтверждающего правильность выполнения программ.
3. Результаты выполнения в терминале команд `ipcs` и `ipcrm` при удалении семафоров.
4. Тексты модифицированных программ 06-3a.c и 06-3b.c, в которых обеспечены с помощью семафоров взаимоисключения для их правильной работы. Копия экрана, подтверждающего правильность выполнения программ.
5. Текст программы, использующей семафоры для синхронизации двусторонней поочередной связи процесса-родителя и процесса-ребенка через `pipe`. Копия экрана, подтверждающего правильность выполнения программ.
6. Вывод.

## **Лабораторная работа 6**

### **Сообщения как средства связи и средства синхронизации процессов**

#### ***Порядок выполнения работы***

1. Изучить теоретические сведения.
2. Откомпилировать программы, текст которых приведен в методических указаниях. Проверить правильность их работы.
3. Разработать программы, предназначенные для самостоятельной подготовки. Откомпилировать, проверить правильность их работы.

#### ***Содержание отчета***

1. Тексты программы 09-1a.c и 09-1b.c для однонаправленной передачи текстовой информации через очередь сообщений. Копия экрана, подтверждающего правильность выполнения программ.
2. Тексты модифицированных программ 09-1a.c и 09-1b.c для однонаправленной передачи нетекстовых сообщений. Копия экрана, подтверждающего правильность выполнения программ.
3. Тексты программ, осуществляющих двустороннюю связь через одну очередь сообщений. Копия экрана, подтверждающего правильность выполнения программ.
4. Тексты программ сервера и клиентов для предложенной схемы мультиплексирования сообщений. Копия экрана, подтверждающего правильность выполнения программ.
5. Тексты программ, реализующих семафоры через очереди сообщений. Копия экрана, подтверждающего правильность выполнения программ.