

# **Лабораторная работа №8**

**НКАбд-02-23**

Выборнов Дмитрий Валерьевич

# 1 Цель работы

Целью этой лабораторной работы является ознакомление с инструментами поиска файлов и фильтрации текстовых данных и приобретение практических навыков: по управлению процессами (и заданиями), по проверке использования диска и обслуживанию файловых систем.

## **2 Задание**

1. Изучение теоретического материала.
2. Выполнение основной части лабораторной работы.
3. Ответы на контрольные вопросы.

### 3 Теоретическое введение

В системе по умолчанию открыто три специальных потока: – `stdin` — стандартный поток ввода (по умолчанию: клавиатура), файловый дескриптор 0; – `stdout` — стандартный поток вывода (по умолчанию: консоль), файловый дескриптор 1; – `stderr` — стандартный поток вывод сообщений об ошибках (по умолчанию: консоль), файловый дескриптор 2. Большинство используемых в консоли команд и программ записывают результаты своей работы в стандартный поток вывода `stdout`. Например, команда `ls` выводит в стандартный поток вывода (консоль) список файлов в текущей директории. Потоки вывода и ввода можно перенаправлять на другие файлы или устройства. Проще всего это делается с помощью символов `>`, `>>`, `<`, `<<`. Конвейер (`pipe`) служит для объединения простых команд или утилит в цепочки, в которых результат работы предыдущей команды передаётся последующей. Команда `find` используется для поиска и отображения на экран имён файлов, соответствующих заданной строке символов. Найти в текстовом файле указанную строку символов позволяет команда `grep`. Команда `df` показывает размер каждого смонтированного раздела диска. Команда `du` показывает число килобайт, используемое каждым файлом или каталогом. Любую выполняющуюся в консоли команду или внешнюю программу можно запустить в фоновом режиме. Для этого следует в конце имени команды указать знак амперсанда `&`. Любой команде, выполняемой в системе, присваивается идентификатор процесса (`process ID`). Получить информацию о процессе и управлять им, пользуясь идентификатором процесса, можно из любого окна командного интерпретатора. Команда `ps` используется

для получения информации о процессах. Для получения информации о процессах, управляемых вами и запущенных (работающих или остановленных) на вашем терминале, используйте опцию `aux`.

## 4 Выполнение лабораторной работы

1. Т. к. я использую WSL, у меня уже был выполнен вход в систему.
2. Сначала я создаю файл `file.txt`, после чего я добавляю в него названия содержимого каталогов `~` и `/etc` при помощи `ls`.
3. Я нахожу все файлы с расширением `.conf`, собираю их в файл `conf.txt` и вывожу его содержимое на экран.
4. Я нахожу все файлы, которые находятся в моём домашнем каталоге и начинаются с символа `s`, используя `grep`. Также это можно сделать при помощи команды `find` с несколькими опциями.
5. При помощи `ls` и `grep` я вывожу все файлы, начинающиеся с `h`, и сортирую их при помощи `sort`.
6. Я запускаю описанный процесс в фоновом режиме, поставив в конце `&`.
7. После выполнения предыдущего шага, я удаляю каталог `logfile`.
8. Запускаю редактор `gedit` в фоновом режиме.
9. Определяю PID `gedit` при помощи `ps` и `grep`.
10. Использую PID, найденный в предыдущем шаге, чтобы остановить `gedit`.
11. Просматриваю информацию о `df` и `du` и выполняю эти команды с некоторыми опциями.
12. Воспользовавшись командой `man`, я понял, что для того, чтобы найти названия всех директорий в моём домашнем каталоге нужно использовать опции `-type d` и `-maxdepth 1`.

## 5 Выводы

Выполнив эту лабораторную работу, я ознакомился с инструментами поиска файлов и фильтрации текстовых данных и приобрёл практические навыки по управлению процессами и заданиями, проверке использования диска и обслуживанию файловых систем.

## 6 Контрольные вопросы.

1. По умолчанию в системе открыто 3 специальных потока: `stdin` (стандартный ввод), `stdout` (стандартный выход) и `stderr` (стандартный канал для ошибок).
2. Полностью перезаписывает содержимое файла, а `>` - добавляет к уже существующему.
3. Конвейер - цепочка команд, где результат одной команды передаётся следующей.
4. Процесс - это запущенная единица программы.
5. PID - идентификатор процесса, а GID - идентификатор группы.
6. Задачи - запущенные фоном программы, которыми можно управлять при помощи команды `jobs`.
7. `top` Показывает программы, работающее на данный момент, а `htop` - делает то же самое, при этом позволяя пользоваться мышкой для взаимодействия.
8. Для поиска файлов используется команда `find`, (`find` путь -опции). Так, поиск всех файлов в корневом каталоге, которые содержат в своём названии слово `test`, выглядит так: `find ~ -name test`
9. Да, файл можно найти по его содержимому. Для этого нужно использовать конвейер с командами `find` и `grep`.



10. Объём оставшейся свободной памяти можно определить при помощи команды `df`.
11. Объём своего домашнего каталога можно при помощи `df`, найдя его в списке, который выводит на экран эта команда.
12. Удалить зависший процесс можно при помощи команды `kill`.