Министерство образования Республики Беларусь

Учреждение образование

«Белорусский государственный технологический университет»

Кафедра программной инженерии

«Администрирование и безопасность Интернет-систем»

**Отчет к лабораторной работе №2:**

«Сетевые операционные системы»

Выполнил:

Студент 4 курса 6 группы ФИТ

Хлыстов Глеб Георгиевич

Проверила:

Сазонова Дарья Владимировна

Минск 2023

# **Оболочка и ее дочерние процессы**

В системе UNIX существует много уровней программ, начиная с ядра и заканчивая конкретным приложением или командой.

* Процесс, происходящий при выполнении команды $ps в командной оболочке UNIX-подобной операционной системы, такой как Linux состоит из следующих этапов:
* Оболочка ищет исполняемый файл ps в директориях, указанных в переменной окружения PATH, например, /bin/ps.
* Если ps найден, оболочка создает новый дочерний процесс для выполнения ps.
* Ядро операционной системы выделяет память и загружает в нее ps.
* Новый процесс ps выполняет свою задачу, в данном случае, отображает информацию о процессах.
* По завершении работы ps, ядро освобождает память и завершает процесс.

Результат выполнения ps выводится в командной строке.

Команды вводятся в оболочке, которая управляет созданием и управлением процессами. В результате выводится перечень выполняющихся в данный момент команд: программа оболочки и программа ps. Результат выполнения команды представлен на рисунке 1.1.

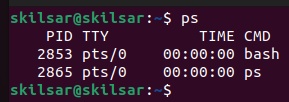


Рисунок 1.1 – Результат выполнения команды ps на ОС Ubuntu

Далее для демонстрации создания дочернего процесса была выполнена команда sleep 10 & для создания дочернего процесса независимо от родительского процесса или оболочки. Далее с помощью команды ps выводим список выполняющихся на данный момент команд. Результат выполнения данных команд представлен на рисунке 1.2.

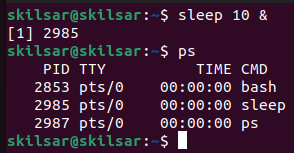


Рисунок 1.2 – Результат выполнения команд sleep и ps на ОС Ubuntu

# **Интерпретация команд**

**Команда** — это исполняемая команда UNIX, программа, утилита или программа оболочки.

**Аргументы** передаются исполняемому модулю. Большинство программ утилит UNIX предполагают, что аргументы будут иметь следующую форму:

ОПЦИИ имена\_файлов

Одно из действий, выполняемых shell для ядра — исключение необязательной информации. Так, применительно к компьютеру одним из типов необязательной информации являются пробелы; следовательно, важно знать, что делает оболочка, встречая пробел. Пробелами являются символы пробелов, горизонтальной табуляции и символы новой строки. Рассмотрим следующий пример:

$echo part A part В part С part A part В part С

В данном случае оболочка интерпретирует командную строку как команду echo, имеющую шесть аргументов, и удаляет пробелы между ними.

Например, при печати заголовков отчета и при необходимости сохранить пробелы, нужно было бы заключить данные в символы кавычек, как в следующем примере:

$echo 'part A part В part С part A part B part С'

Символ одинарных кавычек мешает оболочке “заглядывать” внутрь кавычек. Теперь оболочка интерпретирует эту строку как команду echo с единственным аргументом, который в данном случае является строкой символов, содержащей пробелы.

Результат выполнения приведённых команд представлен на рисунке 2.1.

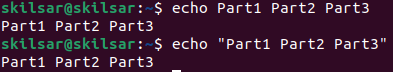


Рисунок 2.1 – Результат выполнения команды echo

# **Выполнение практических заданий**

Midnight Commander - один из файловых менеджеров с текстовым интерфейсом типа Norton Commander для UNIX-подобных операционных систем, а также Microsoft Windows и Windows NT, OS/2, macOS.

Для запуска файлового менеджера Midnight Commander в операционной системе Ubuntu необходимо выполнить команду $ mc. Результат выполнения данной команды представлен на рисунке 3.1.

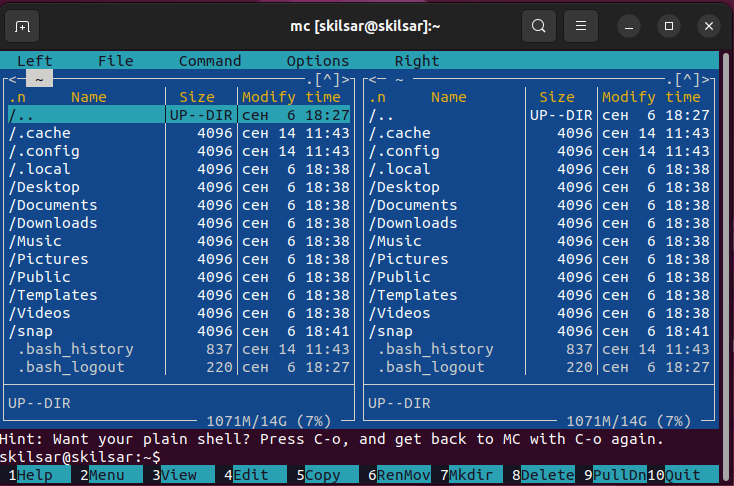


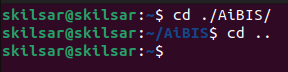
Рисунок 3.1 – Файловой менеджер Midnight Commander, ОС Ubuntu

Далее будут представлены результаты выполнения 50 популярных команд UNIX, приведенных в Приложении 1 к данной лабораторной работе, на примере ОС Linux Ubuntu.

* 1. cat – вывод содержимого в поток ввода консоли.



* 1. cd – навигация по файловой системе используя shell.



* 1. cp – копирование файлов или папок



* 1. find – поиск файлов



* 1. grep – поиск в файлах текста используя регулярные выражения



* 1. ls – список файлов в каталоге



* 1. more – чтение файлов



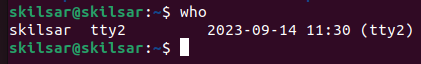
* 1. rm – удаление файлов и директорий



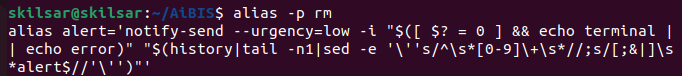
* 1. vi – редактор vim



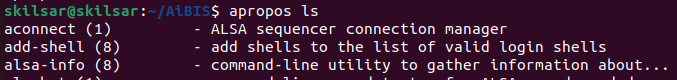
* 1. who – дата входа пользователя



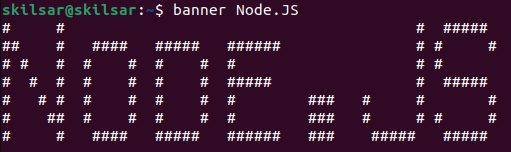
* 1. adduser – создание пользователя
  2. alias – пседонимы для комманд



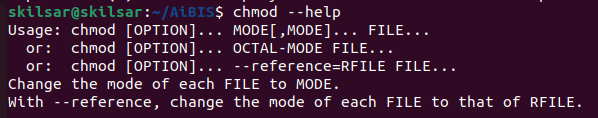
* 1. apropos – поиск по руководствам man аналогична (man -k *<параметр>)*



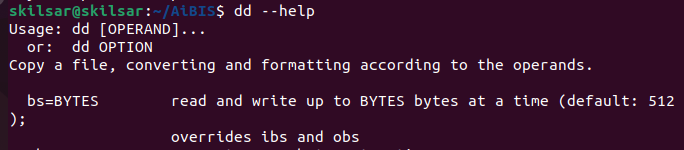
* 1. banner



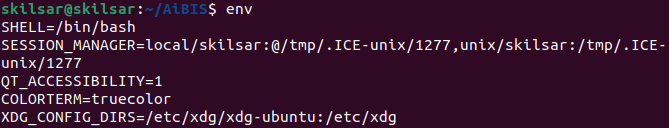
* 1. bg – создание фоновых процессов
  2. chgrp – изменение прав на файл
  3. chmod – изменение прав на файл или директорий



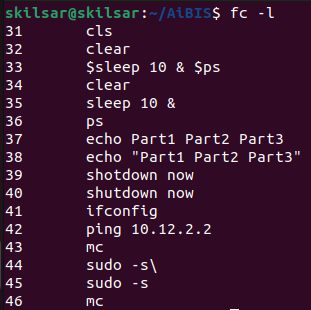
* 1. chown – назначение владельца файла
  2. chroot - позволяет изменить системное окружение для выполняемых процессов
  3. dd – преобразование форматов



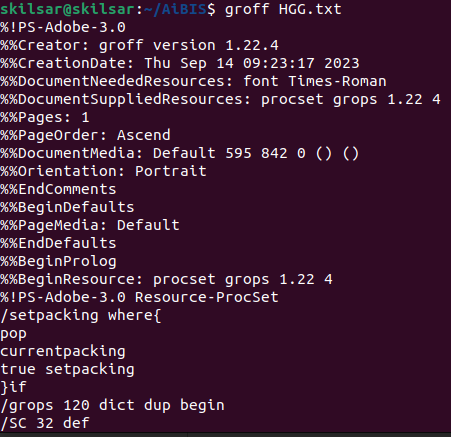
* 1. env – позволяет проссматривать переменные среды



* 1. fc – список последних комманд



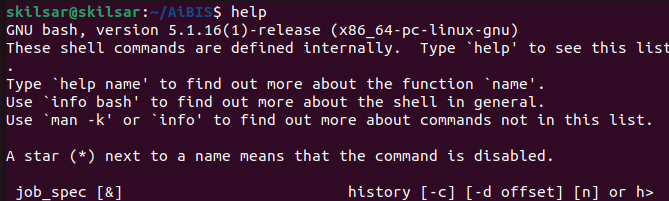
* 1. fg – управление фоновыми процессами
  2. groff – утилита для ввода и вывода содержимого файлов



* 1. gzip – утилита для сжатия и распоковки файлов



* 1. halt – завершение системы
  2. help – справка



* 1. hostname – имя машины



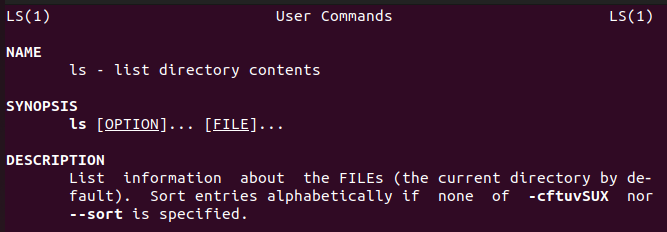
* 1. kill – завершение процесса
  2. less – просмотр содержимого файла с возможностью прокрутки



* 1. login – вызов утилиты для смены пользователя
  2. logout – выход из учетной записи
  3. lpc – управление построчной печатью



* 1. lpd – фоновый процесс построчной печати
  2. lpr – команда построчной печати
  3. make – утилита для перекомпиляции
  4. man – утилита для отображения руководства



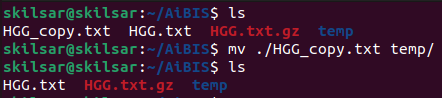
* 1. mesg – утилита для обмена сообщениями
  2. mkdir – создание директории



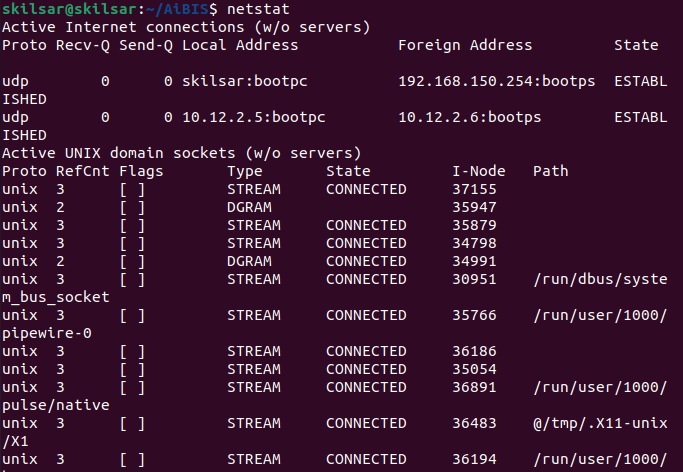
* 1. mkfs – монтирование файловой системы
  2. mkswap – установка размера области свопинга
  3. more – просмотр содержимого текстового файла



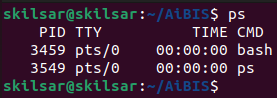
* 1. mount – подключение файловой системы
  2. mv – перемещение директорий или файлов



* 1. netstat – информация о сетевых соединениях



* 1. passwd – установка нового пароля
  2. ps – вывод список процессов пользователя



* 1. pwd – имя текущего каталага



* 1. su – вход в режим суперпользователя



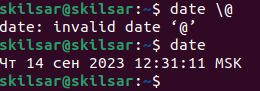
* 1. swapoff – отключение свопинга
  2. swapon – включение свопинга

# **Выполнение упражнений**

1. Объясните, что произойдет в случае ввода команды:

$ date \@

Выводом будет invalid parameter @. Без использования \@ будет выведено текущее время.



2. Большинство интерпретаторов shell интерпретируют символ # как символ начала комментария и игнорируют весь текст от символа # до конца строки. Учитывая это, объясните приведенный ниже диалог. Предполагается, что для стирания также используется символ #.

$date

Mon Sep 26 12:39:56 EET 2006

$# date

Пустота

$\#date

$\\#date

date : not found

3. Попробуйте выполнить команду:

$ls /usr/games

а, затем что-либо из предложенного ею.

Остановите выполнение запущенной программы, одним из известных Вам способов.

Объясните, что выполнила система в ответ на ваши команды.

Была выбрана игра kmines – аналог сапёра из ОС Windows. Для закрытия использовался shortcut – ctrl + c.

4. В чем состоит разница между следующими командами:

$ls junk

Попытка вывода содержимого папки под названием ‘junk’

$ls /

Ожидания дополнительного ввода

$ls

Вывод содержимого текущего директория

$ls \*

Вывод содержимого текущего директория с форматированием

$ls’\*’

Попытка вывода содержимого папки под названием ‘\*’

$echo junk

Вывод строки ‘junk’

$echo /

Ожидания дополнительного ввода

$echo

Вывод пустой строки

$echo \*

Вывод всех файлов в текущей директории

$echo ‘\*’

Вывод строки ‘\*’

$‘\*’

Несуществующая команда

5. Объясните, почему команда

$ls > ls.out

Включает ls.out в список имен []?

6. Объясните, результат выполнения команды:

$wc temp > temp

Получение метаинформации о файле temp и запись его в файл temp

7. Что произойдет, если вы ошибетесь в имени команды, задав

$who > temp

Ничего

8. Объясните разницу между командами

$who | sort

и

$who > sort

Одна ждёт выполнения первой команды и начинает после её окончания выполнение второй. Во втором случае перенаправление потока вывода у исполнения одной команды в поток ввода у другой

# **Вывод**

В данной лабораторной работе я ознакомился с работой в UNIX-подобных операционных системах. Ознакомились с примерами команд и списком наиболее используемых и выполнила их для ознакомления в shell.