Отчёт по лабораторной работе №6

Управление процессами

Лабси Мохаммед

Содержание

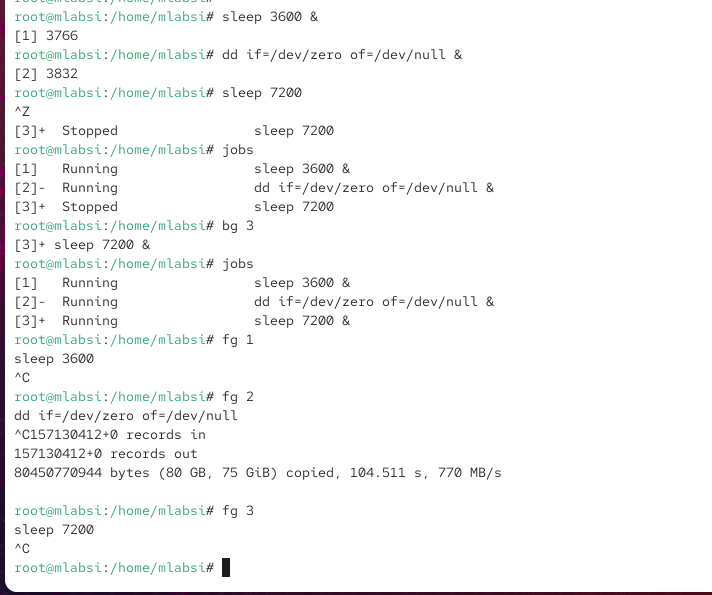
# 1 Цель работы

Получить навыки управления процессами операционной системы.

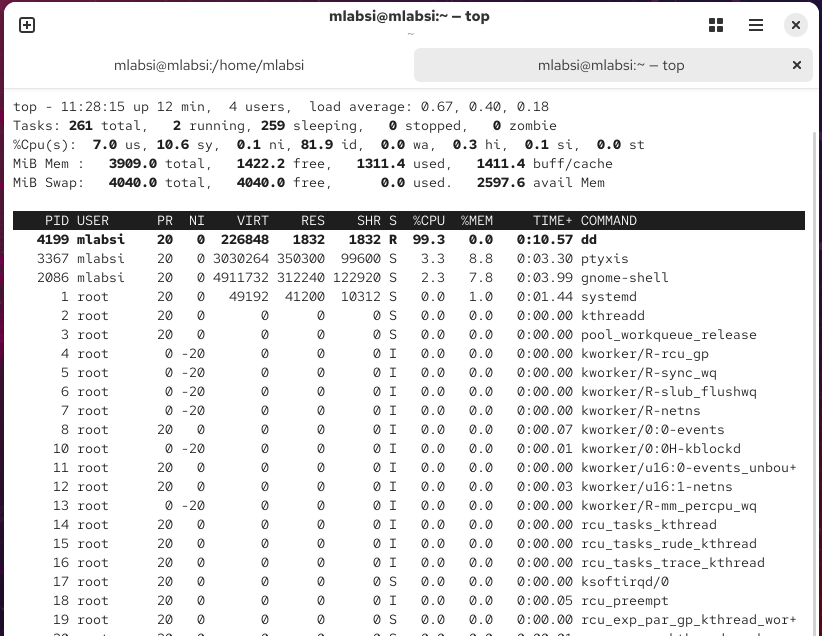
# 2 Ход выполнения

## 2.1 Управление заданиями

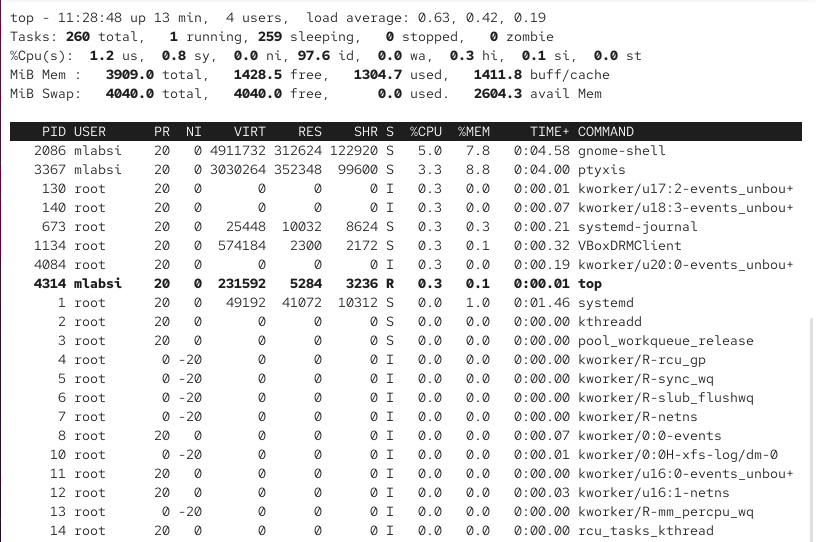
1. Получены права администратора с помощью команды **su -**.  
   После ввода пароля открыт сеанс суперпользователя **root**.
2. Запущены три процесса: два в фоновом режиме (**sleep 3600 &**, **dd if=/dev/zero of=/dev/null &**) и один без символа **&** (**sleep 7200**), из-за чего оболочка ожидала завершения последнего задания.  
   Для остановки активного процесса использована комбинация **Ctrl + Z**, что перевело его в состояние *Stopped*.

* 
* Рис. 1: Запуск фоновых процессов и приостановка одного из них

1. Выполнена команда **jobs**, чтобы просмотреть список запущенных заданий.  
   Отображены три процесса: первые два — в состоянии **Running**, третье — **Stopped**.
2. Для продолжения выполнения третьего задания в фоновом режиме использована команда **bg 3**.  
   Проверка через **jobs** подтвердила, что все процессы теперь работают в фоне.
3. Задание 1 (**sleep 3600**) было выведено на передний план с помощью **fg 1** и прервано комбинацией **Ctrl + C**.  
   Аналогичным образом остановлены задания 2 и 3.
4. В другом терминале, под пользователем **mlabsi**, запущен процесс **dd if=/dev/zero of=/dev/null &**.  
   После закрытия терминала с помощью команды **exit** процесс продолжил выполняться.

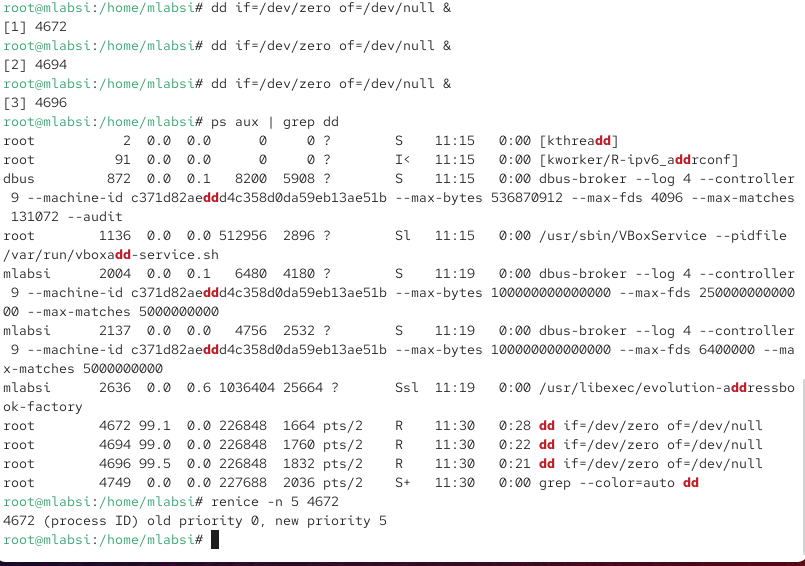
* 
* Рис. 2: Работа фоновых процессов dd и просмотр через top

1. Вызвана команда **top**, где видно, что процесс **dd** активно использует ресурсы CPU (более 99 %).  
   После нажатия **k** и ввода PID процесса задание было завершено.

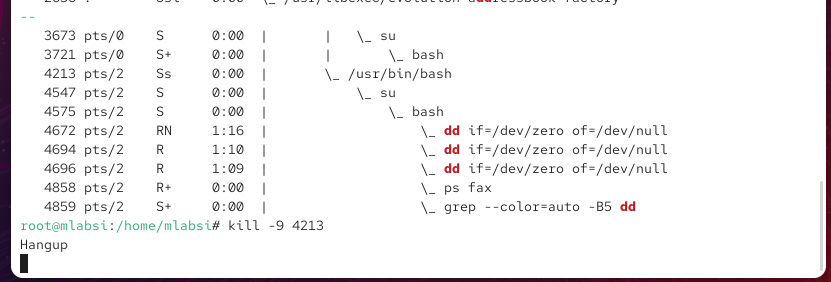
* 
* Рис. 3: Завершение процесса dd через top

## 2.2 Управление процессами

1. Получены права администратора с помощью **su -**.  
   Запущено три процесса **dd**: **dd if=/dev/zero of=/dev/null &**, **dd if=/dev/zero of=/dev/null &**, **dd if=/dev/zero of=/dev/null &**.
2. Для просмотра списка активных процессов использована команда **ps aux | grep dd**.  
   В результате выведены строки с тремя процессами **dd** и их PID.

* 
* Рис. 4: Просмотр процессов dd

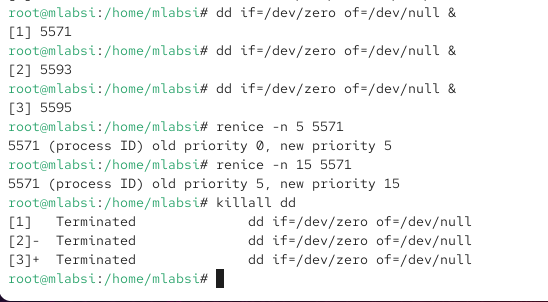
1. Приоритет одного из процессов изменён командой **renice -n 5 4672**, что увеличило значение **nice** и уменьшило его приоритет при планировании.
2. Для отображения иерархии запущенных процессов применена команда **ps fax | grep -B5 dd**, показавшая родительские оболочки и их дочерние процессы **dd**.
3. Определён PID родительского процесса, из которого были запущены задания **dd**, и выполнена команда **kill -9 4213**.  
   После этого оболочка **root** была закрыта, а все дочерние процессы **dd** завершены.

* 
* Рис. 5: Изменение приоритета и завершение родительского процесса dd

## 2.3 Самостоятельная работа

### 2.3.1 Задание 1

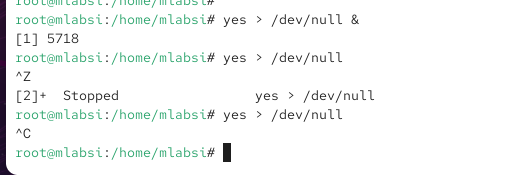
1. Запущена команда **dd if=/dev/zero of=/dev/null** трижды в фоновом режиме.  
   Каждый процесс получил свой идентификатор PID.

* 
* Рис. 6: Запуск трёх фоновых процессов dd

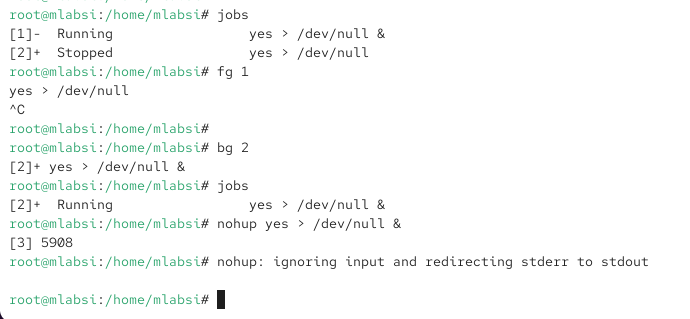
1. Для одного из процессов изменён приоритет с помощью команды **renice -n 5 5571**,  
   что установило значение приоритета **5**.
2. После этого тот же процесс был перенастроен с приоритетом **15**.  
   При изменении приоритета видно, что увеличение значения *nice* снижает приоритет выполнения процесса (чем больше число, тем ниже приоритет).
3. Все три процесса **dd** завершены одновременно с помощью команды **killall dd**.

### 2.3.2 Задание 2

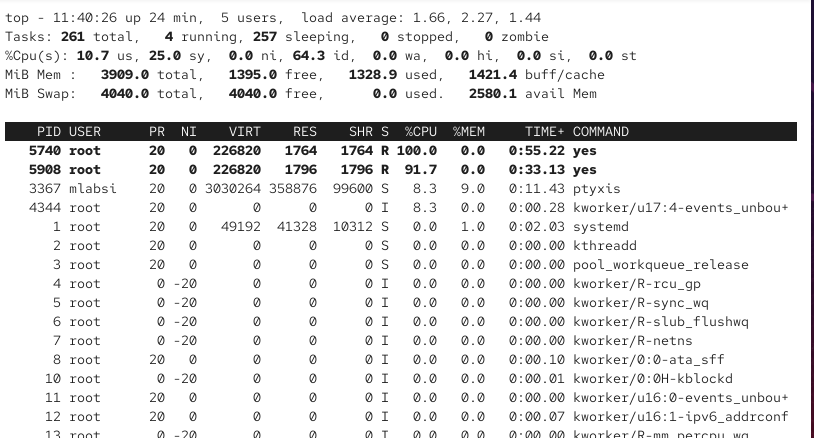
1. Программа **yes** была запущена в фоновом режиме с перенаправлением вывода в **/dev/null**,  
   что позволило избежать загрузки терминала.

* 
* Рис. 7: Запуск yes в фоновом режиме

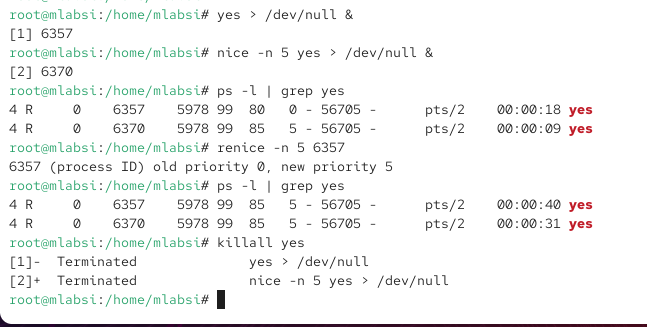
1. Затем программа **yes** была запущена на переднем плане с теми же параметрами.  
   После приостановки с помощью **Ctrl + Z** она была вновь запущена и впоследствии остановлена комбинацией **Ctrl + C**.
2. Состояния заданий проверены с помощью команды **jobs** — одно задание выполнялось в фоне, другое находилось в состоянии *Stopped*.  
   Задание на переднем плане завершено через **fg 1** и **Ctrl + C**.
3. Второе задание переведено в фоновый режим с помощью команды **bg 2**, после чего в списке **jobs** отображено, что оно выполняется.  
   Далее запущен процесс **nohup yes > /dev/null &**, чтобы он продолжил работу даже после закрытия терминала.

* 
* Рис. 8: Работа процессов yes и использование nohup

1. Проверка через **top** показала активные процессы **yes**, интенсивно использующие ресурсы CPU.  
   PID процессов **5740** и **5908** соответствовали запущенным заданиям.

* 
* Рис. 9: Просмотр процессов yes через top

1. Запущено два процесса **yes** — один обычный, другой с повышенным приоритетом, заданным с помощью **nice -n 5 yes > /dev/null &**.  
   Командой **ps -l | grep yes** подтверждено различие их приоритетов.  
   Один из процессов имел приоритет **0**, другой — **5**.
2. Приоритет процесса с PID **6357** был изменён утилитой **renice -n 5 6357**,  
   после чего оба процесса завершены командой **killall yes**.

* 
* Рис. 10: Изменение приоритета и завершение процессов yes

# 3 Контрольные вопросы

1. **Какая команда даёт обзор всех текущих заданий оболочки?**  
   Для просмотра списка всех заданий используется команда **jobs**.  
   Она отображает состояние заданий, их идентификаторы и режим работы (*Running* или *Stopped*).
2. **Как остановить текущее задание оболочки, чтобы продолжить его выполнение в фоновом режиме?**  
   Текущее задание можно приостановить комбинацией клавиш **Ctrl + Z**,  
   а затем возобновить его выполнение в фоновом режиме командой **bg**.
3. **Какую комбинацию клавиш можно использовать для отмены текущего задания оболочки?**  
   Для завершения текущего задания используется комбинация **Ctrl + C**.
4. **Необходимо отменить одно из начатых заданий. Доступ к оболочке, в которой в данный момент работает пользователь, невозможен. Что можно сделать, чтобы отменить задание?**  
   Можно воспользоваться командой **kill** с указанием PID процесса или использовать **killall** с именем команды,  
   например: **kill -9**  или **killall** .
5. **Какая команда используется для отображения отношений между родительскими и дочерними процессами?**  
   Для отображения иерархии процессов используется команда **ps fax**.  
   Она показывает дерево процессов и их взаимосвязи.
6. **Какая команда позволит изменить приоритет процесса с идентификатором 1234 на более высокий?**  
   Для повышения приоритета используется команда **renice -n -5 1234**,  
   где отрицательное значение *nice* повышает приоритет выполнения процесса.
7. **В системе в настоящее время запущено 20 процессов dd. Как проще всего остановить их все сразу?**  
   Все процессы **dd** можно завершить одной командой **killall dd**.
8. **Какая команда позволяет остановить команду с именем mycommand?**  
   Для этого используется команда **killall mycommand**.
9. **Какая команда используется в top, чтобы убить процесс?**  
   В интерактивной утилите **top** для завершения процесса используется клавиша **k**,  
   после чего вводится PID нужного процесса.
10. **Как запустить команду с достаточно высоким приоритетом, не рискуя, что не хватит ресурсов для других процессов?**  
    Для этого применяется команда **nice** с положительным значением, например:  
    **nice -n 10**  — она запускает процесс с пониженным приоритетом,  
    позволяя системе распределять ресурсы между задачами более равномерно. # Заключение

В ходе работы были освоены основные команды и приёмы управления заданиями и процессами в Linux, включая изменение приоритета, перевод процессов между фоновым и передним режимами и их завершение.