Отчёт по лабораторной работе №6

Управление процессами

Амина Аджигалиева

Содержание

# 1 Цель работы

Получить навыки управления процессами операционной системы.

# 2 Ход выполнения работы

## 2.1 Управление заданиями

Сначала я получила права суперпользователя с помощью команды su.  
После этого были запущены несколько процессов:

* команда sleep 3600 & запустила задачу в фоновом режиме,
* команда dd if=/dev/zero of=/dev/null & создала ещё один фоновый процесс,
* команда sleep 7200 была запущена без символа &, поэтому процесс занял терминал.

Для остановки выполнения последнего процесса я использовала сочетание клавиш **Ctrl+Z**.  
Затем с помощью команды jobs проверила список активных заданий.

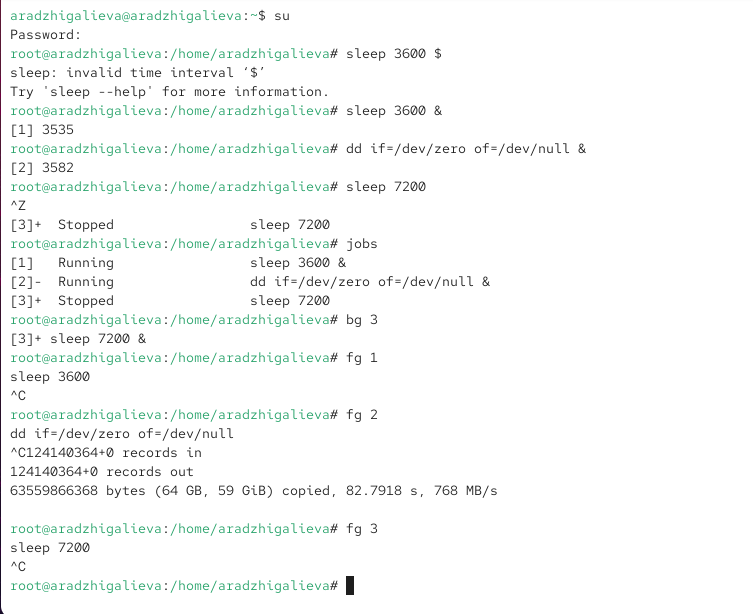


Рис. 1: Запуск и просмотр фоновых заданий

Далее я перевела задание №3 в фоновый режим с помощью команды bg 3.  
После этого выполнила последовательное переключение процессов на передний план:

* fg 1 — процесс был остановлен с помощью **Ctrl+C**,
* fg 2 — также был завершён через **Ctrl+C**,
* fg 3 — аналогично остановлен.

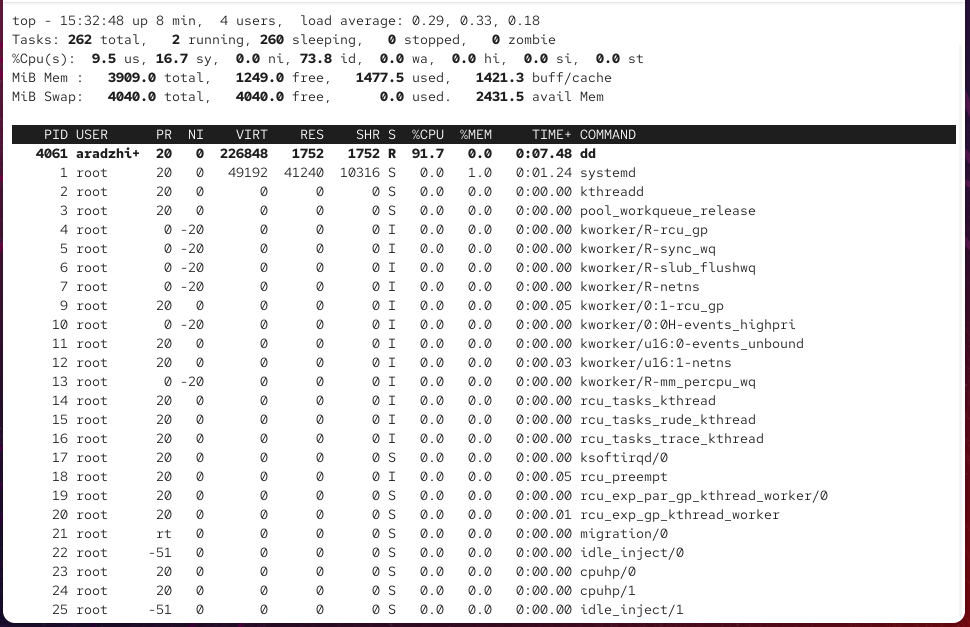


Рис. 2: Управление заданиями с помощью fg и bg

В другом терминале под своей учётной записью я снова запустила процесс dd if=/dev/zero of=/dev/null &.  
Затем закрыла сессию командой exit.  
После этого в новом терминале открыла монитор процессов top и убедилась, что процесс dd продолжает выполняться.

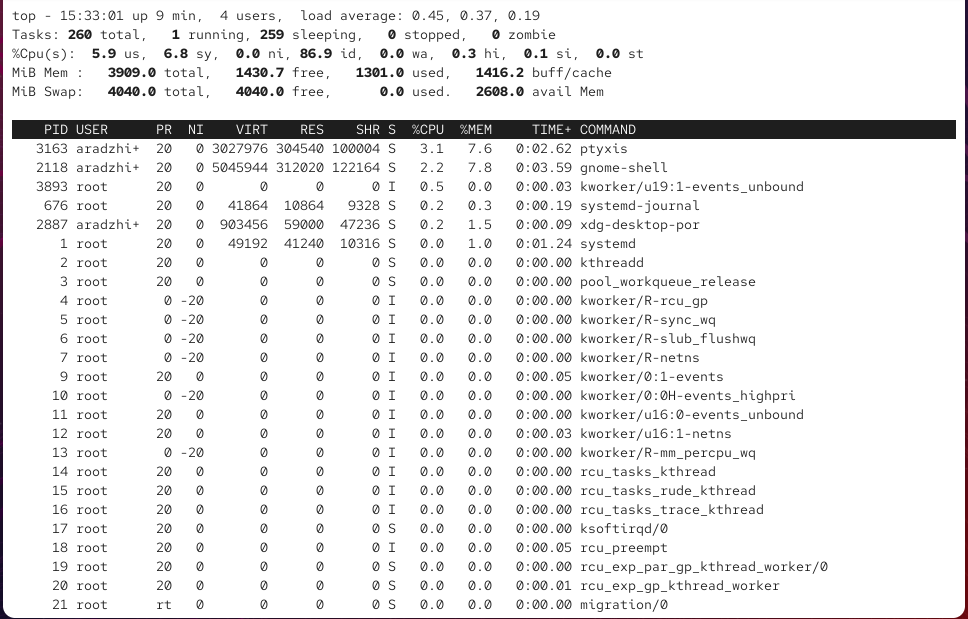


Рис. 3: Мониторинг процесса dd с помощью top

Для завершения процесса я снова запустила top, использовала клавишу k и вручную завершила задание dd. После этого вышла из top с помощью клавиши **q**.

## 2.2 Управление процессами

Сначала я получила права администратора с помощью команды su.  
Затем последовательно запустила три процесса:

* dd if=/dev/zero of=/dev/null &
* dd if=/dev/zero of=/dev/null &
* dd if=/dev/zero of=/dev/null &

После этого я выполнила команду ps aux | grep dd, чтобы просмотреть все процессы, содержащие строку dd.  
В выводе отобразились три запущенных процесса с их PID.

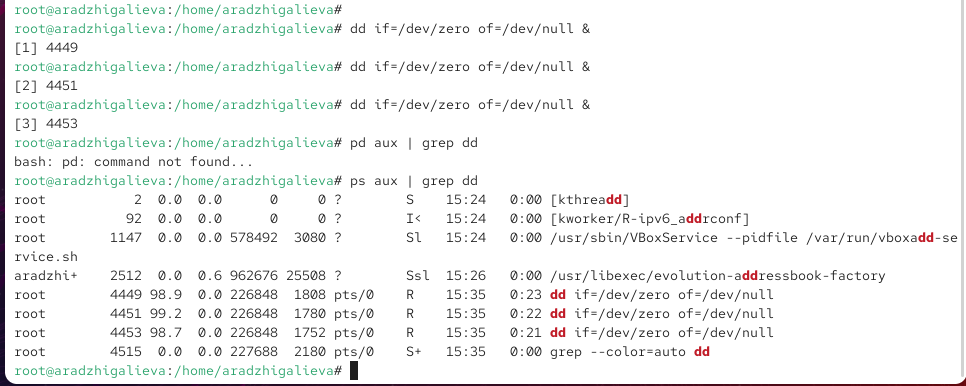


Рис. 4: Просмотр процессов dd через ps aux

Затем я изменила приоритет одного из процессов dd с помощью команды renice -n 5 <PID>.  
Для анализа иерархии процессов я использовала команду ps fax | grep -B5 dd.  
Это позволило увидеть не только процессы dd, но и их родительскую оболочку.

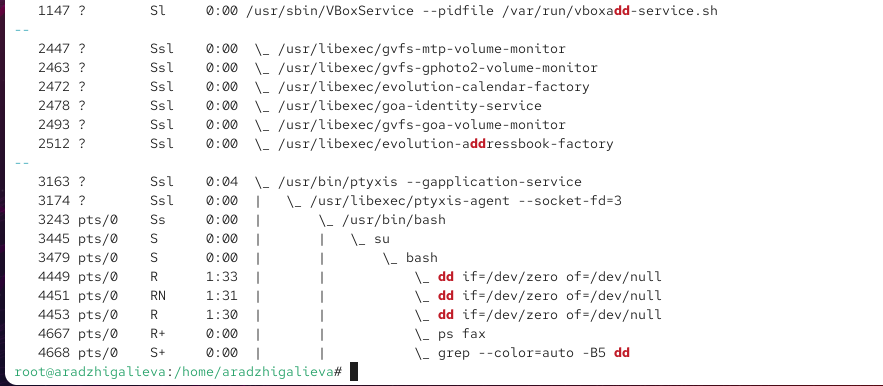


Рис. 5: Отображение иерархии процессов через ps fax

После определения PID оболочки, из которой были запущены процессы dd, я применила команду kill -9 <PID>.  
В результате оболочка завершила работу, а вместе с ней были остановлены все дочерние процессы dd.

## 2.3 Задание 1. Управление приоритетами процессов

Сначала я запустила три процесса с помощью команды dd if=/dev/zero of=/dev/null &, чтобы они выполнялись в фоновом режиме.

Далее изменила приоритет одного из процессов с помощью команды renice -n -5 <PID>.  
После этого снова применила renice, но уже с параметром -15.

Разница заключается в том, что чем меньше значение nice (и больше отрицательное), тем выше приоритет процесса.  
Таким образом, процесс с -15 получает больше процессорного времени по сравнению с тем же процессом при значении -5.

В завершение я остановила все процессы dd. Сначала команда killall dd была выполнена без прав суперпользователя и вернула сообщение об ошибке «Operation not permitted».  
После получения полномочий администратора через su команда killall dd успешно завершила все запущенные процессы dd.

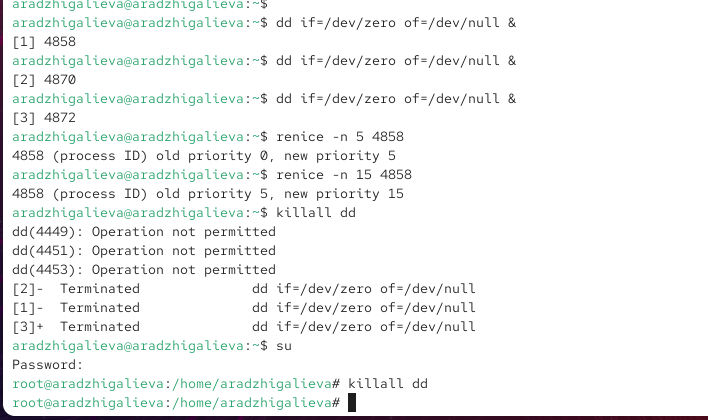


Рис. 6: Управление приоритетами и завершение процессов dd

# 3 Ход выполнения работы

## 3.1 Задание 2. Управление процессами yes

Сначала я запустила программу yes в фоновом режиме с перенаправлением вывода в /dev/null.  
Затем второй процесс yes был запущен на переднем плане и остановлен с помощью **Ctrl+Z**, после чего переведён в фоновый режим.

Проверка состояний через jobs показала наличие двух процессов: один в статусе **Running**, другой в статусе **Stopped**.

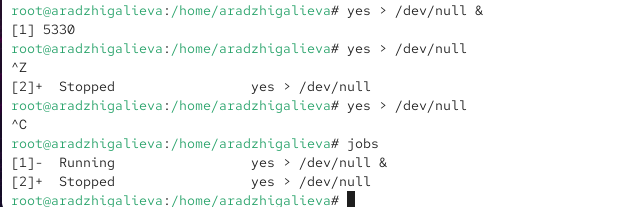


Рис. 7: Запуск и остановка процессов yes

Далее я использовала команду fg 1, чтобы вернуть первый процесс на передний план и завершить его с помощью **Ctrl+C**.  
Процесс со статусом **Stopped** был переведён в фоновый режим с помощью bg 2. После этого оба процесса находились в состоянии **Running**.

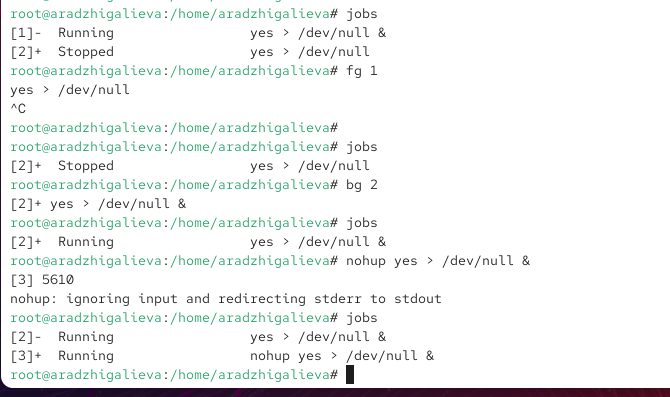


Рис. 8: Перевод процессов между фоном и передним планом

Для того чтобы процесс продолжал выполняться даже после выхода из терминала, я запустила его с использованием nohup yes > /dev/null &.  
Проверка через jobs подтвердила, что процесс выполняется.

Далее я открыла монитор процессов top и убедилась, что процессы yes продолжают работать и загружают процессор.

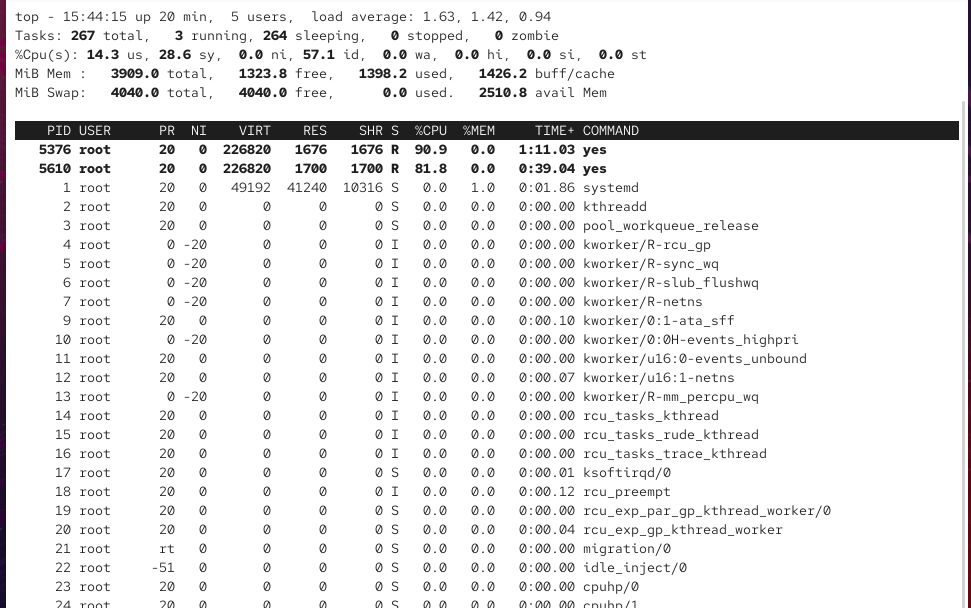


Рис. 9: Мониторинг процессов yes через top

После этого я запустила ещё три процесса yes и завершила два из них разными способами:  
один процесс был остановлен по его PID, а другой — по идентификатору задания.  
Также я протестировала отправку сигнала **SIGHUP** процессам, запущенным как обычным способом, так и через nohup.

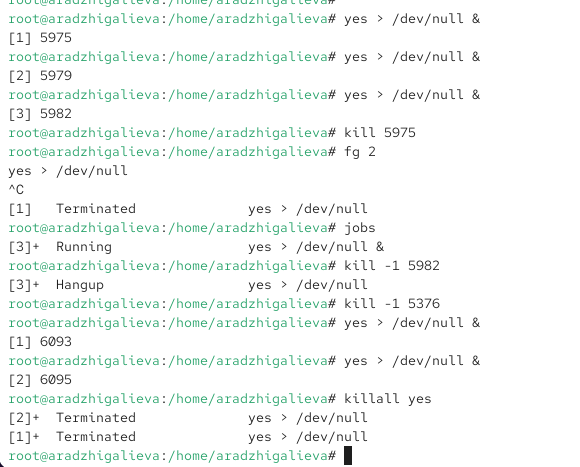


Рис. 10: Завершение процессов по PID и job ID

В конце я запустила несколько процессов yes в фоновом режиме с подавлением вывода и завершила их все одновременно с помощью killall yes.  
Также я применила nice, чтобы запустить процесс с приоритетом выше на 5, и сравнила приоритеты через ps -l.  
Затем изменила приоритет уже запущенного процесса с помощью renice.  
Это позволило на практике убедиться в различиях между абсолютными и относительными приоритетами процессов.

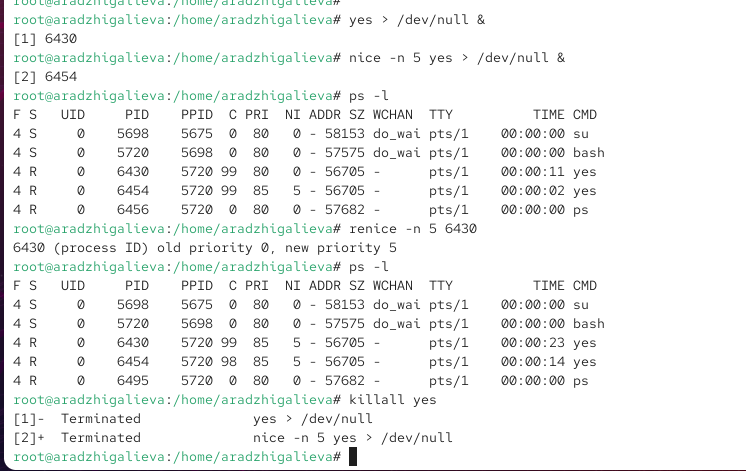


Рис. 11: Изменение приоритета

# 4 Контрольные вопросы

1. **Какая команда даёт обзор всех текущих заданий оболочки?**  
   Для просмотра используется команда jobs.
2. **Как остановить текущее задание оболочки, чтобы продолжить его выполнение в фоновом режиме?**  
   Нужно нажать **Ctrl+Z**, а затем выполнить команду bg.
3. **Какую комбинацию клавиш можно использовать для отмены текущего задания оболочки?**  
   Для завершения 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 <PID> или killall <имя\_процесса> в другой оболочке.
5. **Какая команда используется для отображения отношений между родительскими и дочерними процессами?**  
   Для этого подходит команда ps fax.
6. **Какая команда позволит изменить приоритет процесса с идентификатором 1234 на более высокий?**  
   Следует выполнить: renice -n -5 -p 1234.
7. **В системе в настоящее время запущено 20 процессов dd. Как проще всего остановить их все сразу?**  
   Использовать команду killall dd.
8. **Какая команда позволяет остановить команду с именем mycommand?**  
   Для этого используется killall mycommand.
9. **Какая команда используется в top, чтобы убить процесс?**  
   Внутри top необходимо нажать **k**, ввести PID и подтвердить.
10. **Как запустить команду с достаточно высоким приоритетом, не рискуя, что не хватит ресурсов для других процессов?**  
    Запуск осуществляется с помощью nice. Например: nice -n 10 команда — чтобы процесс имел более низкий приоритет по сравнению с другими.

# 5 Заключение

В ходе выполнения лабораторной работы я освоила базовые приёмы управления заданиями и процессами в Linux.  
Я научилась запускать процессы в фоновом и переднем режиме, приостанавливать и возобновлять их работу, изменять приоритет с помощью nice и renice, а также завершать процессы через kill, killall и утилиту top.  
Кроме того, я закрепила навыки анализа иерархии процессов с использованием команды ps и убедилась в том, что завершение родительского процесса приводит к остановке всех его дочерних задач.  
Полученные знания позволяют более эффективно администрировать систему, управлять её ресурсами и обеспечивать стабильную работу в многозадачной среде.