**Инструкция по выполнению домашнего задания**

1. Сделайте копию Шаблона для домашнего задания себе на Google Диск.
2. В названии файла введите корректное название лекции и вашу фамилию и имя.
3. Зайдите в «Настройки доступа» и выберите доступ «Просматривать могут все в Интернете, у кого есть эта ссылка».
4. Скопируйте текст задания в свой документ.
5. Выполните домашнее задание.
6. Для проверки домашнего задания преподавателем отправьте ссылку на ваш документ в личном кабинете.
7. Все вопросы задавайте в разделе «Вопросы по заданию» у вас в личном кабинете или в комментариях к вашему решению при сдаче домашнего задания.

# **Задание 1**

Многозадачность в современных системах реализована через переключение между процессами с помощью планировщика. Какие операции при этом замедляют работу компьютера?

*Приведите ответ в свободной форме со своим комментарием.*

# **Решение 1**

Во время переключения между процессами происходит расход тактов ядра на выгрузку данных о текущем процессе в память и загрузка данных нового процессора из памяти в кэш. Тратиться время на переключение контекста. Это замедляет общее время на выполнение процесса.

# **Задание 2**

В каких случаях используется планировщик SCHED\_DEADLINE

*Приведите ответ в свободной форме*

# **Решение 2**

SCHED\_DEADLINE используется в процесса реального времение, для тех процессов где критично выполнение в определенные сроки. промежуток времени. В приоритет выполняется то процесс у которого ближайшее время исполнения.

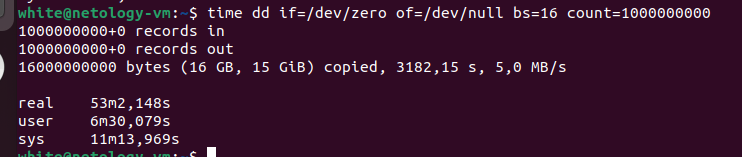
# **Задание 3**

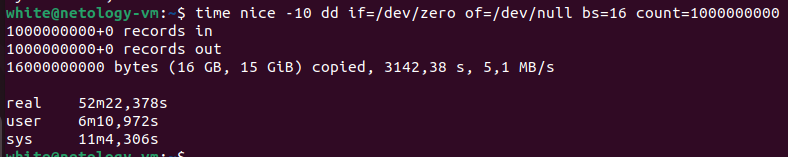
Запустите следующий код, имитирующий нагрузку типа ввод / вывод, с значениями nice 19, 10, 0 и измерьте время исполнения с помощью утилиты time. Объясните получившееся различие во времени исполнения для разных запусков.

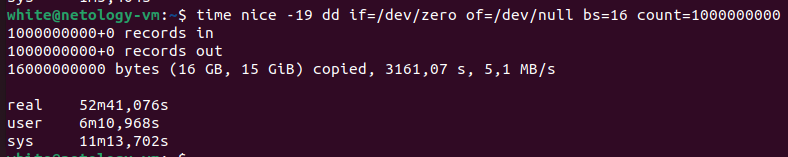
dd if=/dev/zero of=/dev/null bs=16 count=10000000

*Ответ приведите в виде снимка экрана с комментариями в свободной форме*

# **Решение 3**

****

****

****

Реальное время должно увеличивается по мере увеличение значение NICE, чем выше значение nice тем менее приоритетнее задача, а значит процесс все время стоит в ожидание пропуская задачи с более высоким приоритетом.

Также увеличивается время sys, что значит что процесс чаще/больше прерывался и тратилось время на выгрузку загрузку данных из памяти.

Время user должно увеличивается, что значит что процесс сам по себе одинаковый и на него процессор истратил одинаковое время.

**НО, сколько ни пытался воспроизвести не получилось))) тестировал на виртуальной машине и на ноутбуке.**

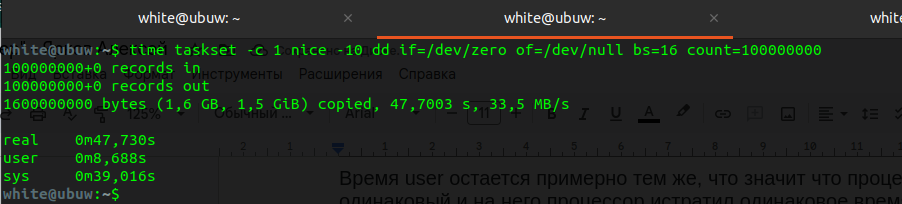
**Задание 4**

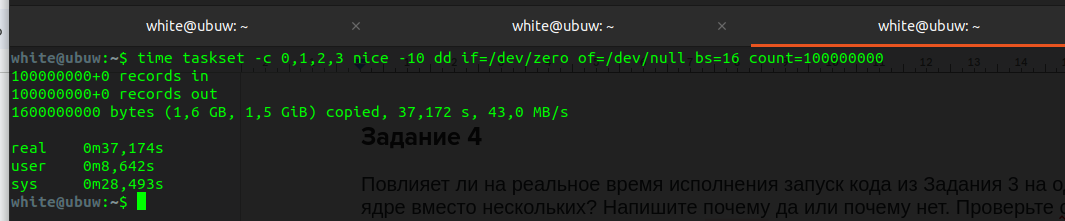
Повлияет ли на реальное время исполнения запуск кода из Задания 3 на одном ядре вместо нескольких? Напишите почему да или почему нет. Проверьте своё предположение с помощью утилиты taskset

Ответ приведите в виде снимка экрана с комментариями в свободной форме

# **Решение 4**

Реальное время уменьшается, так как к задаче привлекается большее количество ядер, но увеличение непропорциональное, так как все равно расходуется время на переключение контекста.





**Задание 5\***

При каких условиях лучше увеличить time slice планирования, а когда лучше уменьшить?

Приведите ответ в свободной форме со своим комментарием.

# **Решение 5**

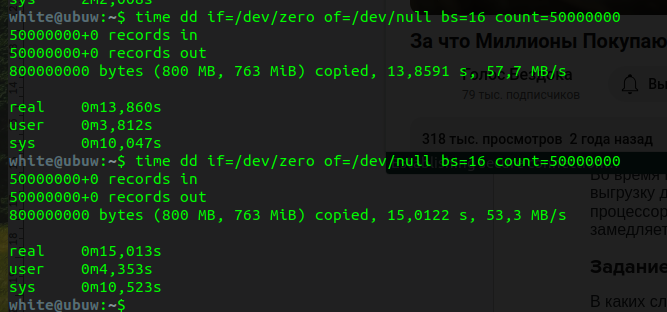
time slice лучше увеличивать в случае использование компьютера как сервера. Когда необходимо сосредоточиться на времени выполнение конкретных процессов, например сервера БД. При этом пострадает отзывчивость системы.И соответственно time slivce уменьшаем в случае использования как рабочей станции пользователя для повышения отзывчивости.

**Задание 6\***

Запустите код из Задания 3 с приоритетом по умолчанию. Одновременно запустите веб-браузер с разными приоритетами. Создайте условия, чтобы получить различную отзывчивость браузера при использовании разных приоритетов. Запишите условия и опишите результаты.

Приведите ответ в свободной форме со своим комментарием.

# **Решение 6**



на скрине при втором запуске dd, параллельно был запущен браузер с youtube, что повлияло на время выполнение команды и увеличило его на 2 секунды, время user увеличилось, это то время которое процесс провел в режим пользователя.

### **Задание 7\***

В каких ситуациях планировщик ввода / вывода noop может быть производительней cfq?

*Приведите ответ в свободной форме со своим комментарием.*

# **Решение 6**

Планировщик noop следует использовать в системах с повышенной нагрузкой на CPU, так как данный планировщик использует минимальное количество команд CPU на одну операцию ввода/вывода. И если в системе повышенный спрос на CPU, планировщик noop будет более производительным. Вся работа уходит на внешний контроллер систем ввода/вывода