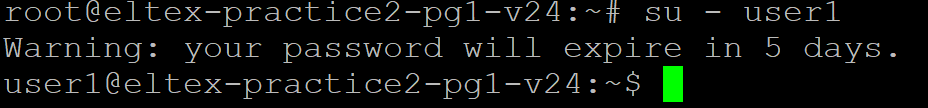
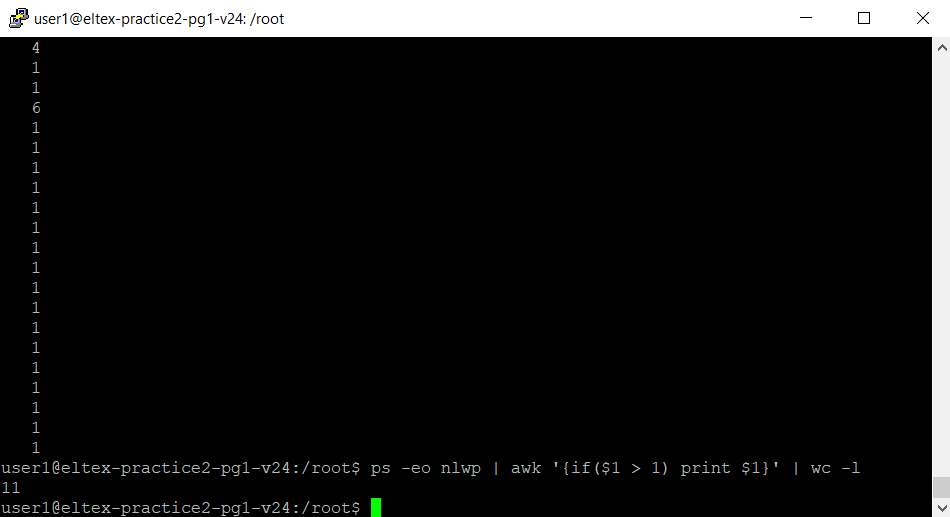
1. Войдите под пользователем user1 из практики 2 (su - user1)



2. Подсчитайте количество процессов, имеющих несколько потоков

Выполнения



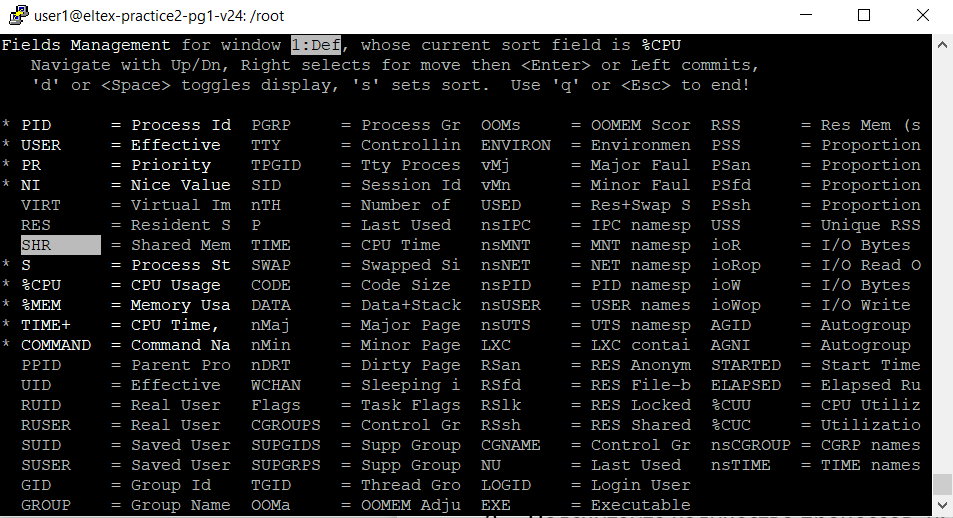
3. Запустите top и настройте вывод полей с информацией о процессе

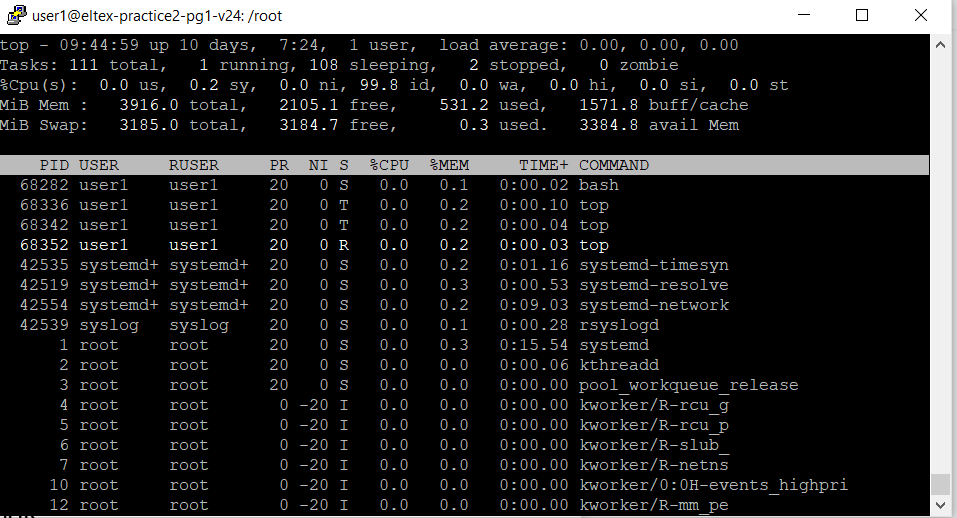
следующим образом:

• удалите поля VIRT, RES, SHR;

• добавьте поле RUSER и сделайте так, чтобы это поле было

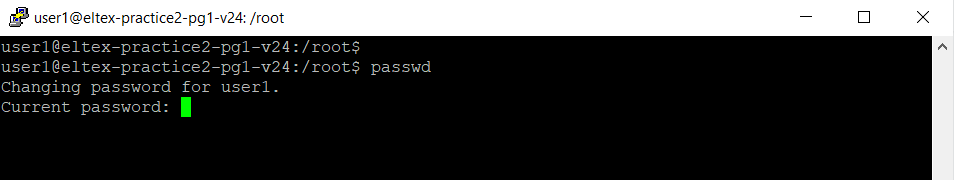
показано после поля USER;





4. В другом терминальном окне выполните команду passwd и оставьте ее в

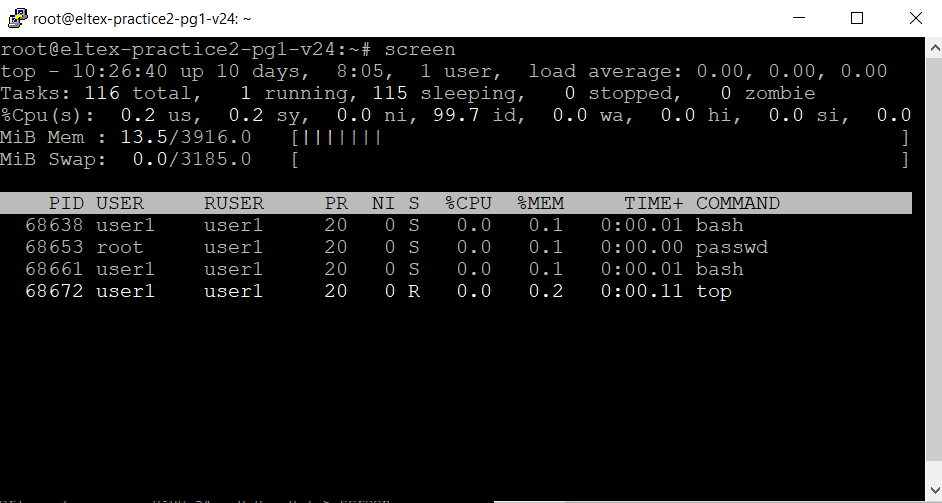
состоянии запроса текущего пароля



5. Перейдите в терминальное окно с top и выполните следующие действия:

• выведите все процессы, для которых реальным пользователем

является пользователь, которым вы вошли в сеанс;

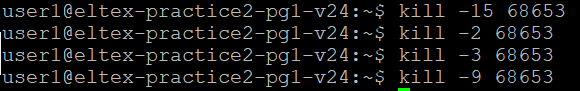


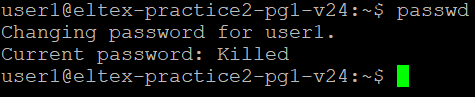
• найдите процесс, запущенный командой passwd;



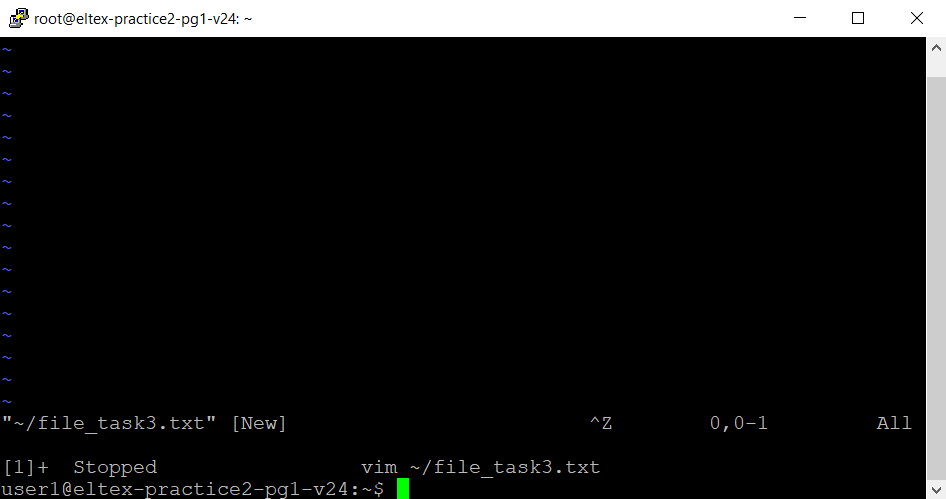
• отправьте этому процессу сигналы 15 (SIGTERM), 2 (SIGINT), 3

(SIGQUIT), 9(SIGKILL)

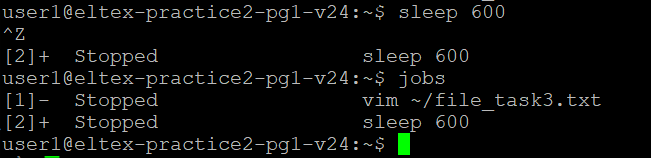




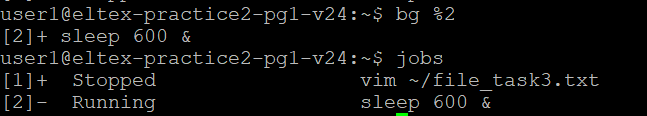
6. Выполните команду vim ~/file\_task3.txt и нажмите Ctrl-Z



7. Выполните команду sleep 600, нажмите Ctrl-Z и выполните команду jobs



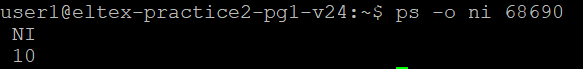
8. Последнее задание (sleep 600) сделайте фоновым



9. Измените число NICE у задания (sleep 600), сделав его равным 10

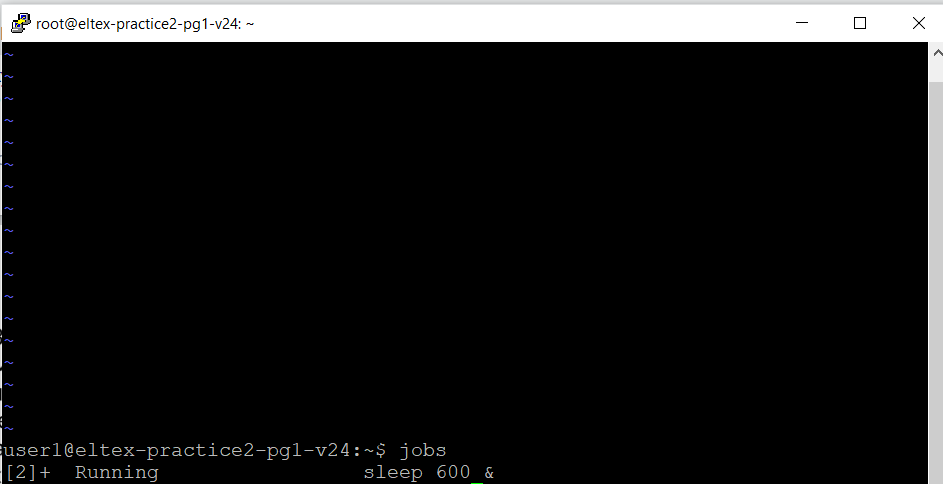


10. Проверьте, что число NICE у этого задания изменилось

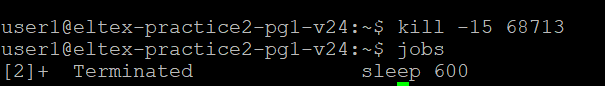


11. Сделайте задание vim ~/file\_ task3.txt активным и выйдите из редактора

fg %1



12. Отправьте сигнал 15 (SIGTERM) заданию sleep 600 и выполните команду jobs

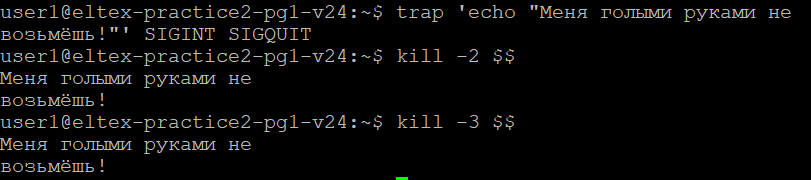


13. Создайте перехватчик сигналов SIGINT и SIGQUIT внутри командного

интерпретатора, который выводит сообщение «Меня голыми руками не

возьмёшь!» (используйте встроенную команду trap) и отправьте сигналы

самому себе



1. Создайте скрипт на языке bash с именем template\_task.sh, делающий

следующее:

• При запуске проверят, что имя скрипта не совпадает с template\_task.sh,

если совпадает - выходит с уведомлением «я бригадир, сам не работаю»

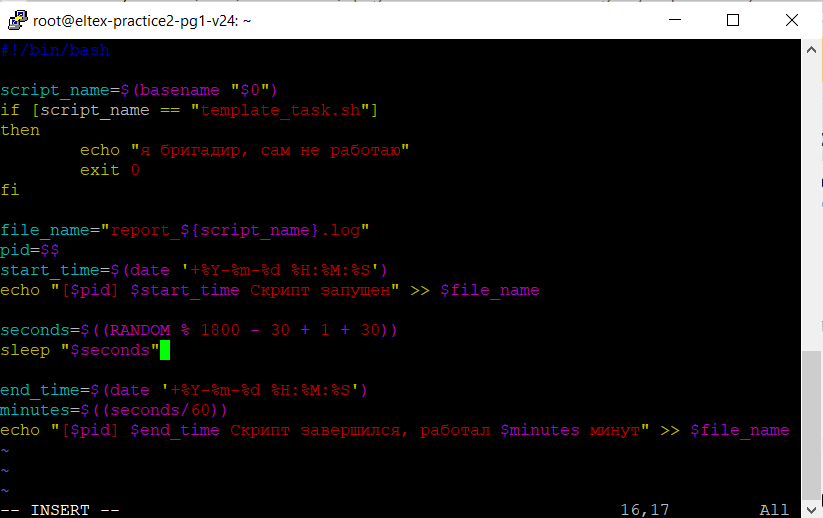
• При запуске дописывает в файл report\_имя\_скрипта\_без\_полного\_пути.log

в рабочем каталоге информацию: [PID] ДАТА ВРЕМЯ Скрипт запущен

• Генерирует случайное число от 30 до 1800 и ждет такое количество секунд

• Дописывает в файл report\_имя\_скрипта\_без\_полного\_пути.log сообщение:

[PID] ДАТА ВРЕМЯ Скрипт завершился, работал N минут



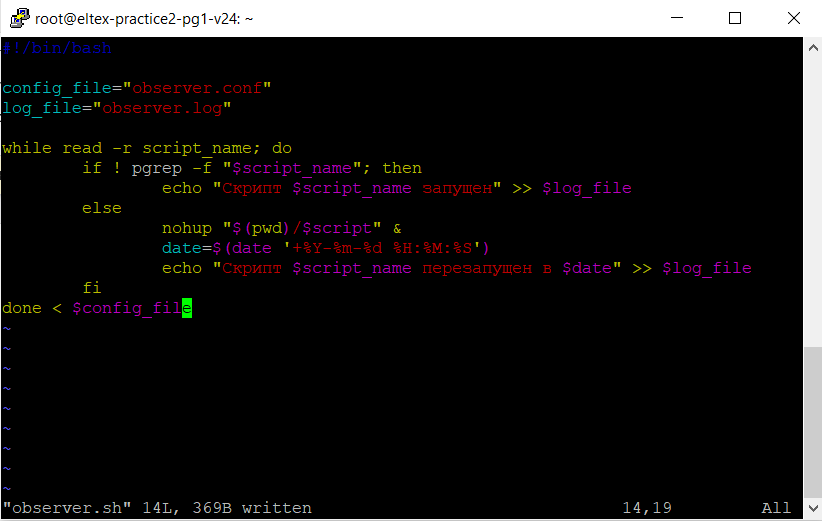
2. Создайте скрипт на языке bash с именем observer.sh, читающий файл

конфигурации со списком скриптов observer.conf, проверяющим их наличие в

списке работающих процессов поиском в /proc и запускающих их в

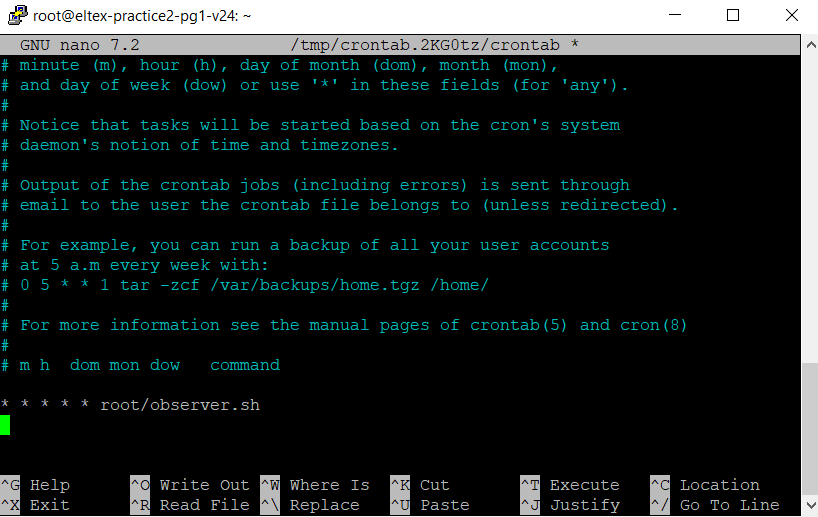
отключенном от терминала режиме (nohup) в случае отсутствия в нем.

Информация о перезапуске дописывайте в файл observer.log



3. Настройте запуск observer.sh посредством cron по расписанию – 1 раз в

минуту

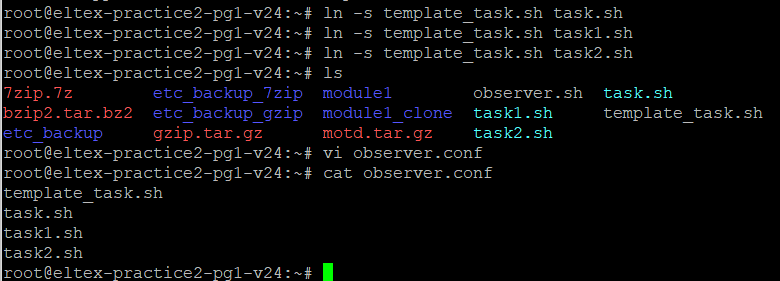


4. Создайте несколько символьных ссылок на файл template\_task.sh с

различными именами (рабочие задачи), добавьте в файл конфигурации

observer.conf соответствующие записи об этих задачах, включая исходный

файл template\_task.sh



5. Соберите статистику работы в виде набора файлов report\_\*.log,

observer.log, приложите их вместе с исходными текстами скриптов в

качестве отчета в виде сжатого архива tar. Не забудьте остановить

процесс, удалив задачу в cron!

