

Module 4 Linux Essentials with Bash (Task 4.10)

4.10.1 Написать скрипт, который выводит приглашения для ввода данных до тех пор, пока не будет введено quit.

endless_input.sh (stop-слово объявляю в переменной, но можно и параметром передать; ничего не сказано о том, что делать с введенными данными, я поставил вывод на экран в качестве действия по умолчанию) здесь и далее в скрипах для экономии места обрезаю "шапку" скрипта с шебангом и установочными set - командами.

```
~/ssh (zsh) 1 ~ (zsh)
to_quit="quit"
while read -p "Enter Your`s Arguments or \"${to_quit}\": " line
do
    if [ $line = $to_quit ]; then break
    else
        echo "$line" # Здесь может быть любое действие с введенной строкой
    fi
done
bruh@wibob-X61:~/devops$ ./endless_input.sh quit
Enter Your`s Arguments or "quit": than
than
Enter Your`s Arguments or "quit": that
that
Enter Your`s Arguments or "quit": your's
your's
Enter Your`s Arguments or "quit": quit
bruh@wibob-X61:~/devops$
```

4.10.2 Написать скрипт, который каждую минуту записывает в файл текущее время и дату и количество процессов. При этом файл должен создаваться в директории /home/user/tmp независимо от пользователя и системы. Запустить его в фоновом режиме.

proc_qlogger.sh. В переменных в начале указаны: настройка удаления лога при новом запуске и имя лог-файла. Запуск в фоне
./proc_qlogger.sh &
или если нужно чтоб работал после логаута
nohup ./proc_qlogger.sh &

```
way_to_store="0" # "1" to clear old log, "0" - add to tail
log_file=~/.tmp/proc_q.log

[[ -f "$log_file" && $way_to_store -eq 1 ]] && rm -f $log_file
[[ ! -d ~/.tmp ]] && mkdir ~/.tmp

while true
do
    printf "%-10s%-10s%-10s" "$(date "+%D %T")" "$(ps aux | wc -l)" >> $log_file && echo >> $log_file
    sleep 59
done
```

```
bruh@wibob-X61:~/devops/scripts_4_10$ cat ~/.tmp/proc_q.log
05/05/20 03:33:14 219
05/05/20 03:34:13 219
05/05/20 03:34:15 219
05/05/20 03:34:17 219
05/05/20 03:40:36 221
05/05/20 03:41:36 221
05/05/20 03:42:36 221
05/05/20 03:43:36 220
05/05/20 03:44:36 220
```

4.10.3 Перевести скрипт из п.2 из фонового режима в приоритетный, потом в фоновый с приостановкой, возобновить его работу в фоновом режиме получить сведения о процессе и завершить работу скрипта передачей соответствующего сигнала.

информация о запущенных программах jobs , чтоб получить информацию о процессе ps -aux | grep имя_скрипта
 переход из фона в приоритет fg номер_программы
 переход в фон с приостановкой ctrl+Z
 возобновление в фоне bg номер_программы
 завершение работы скрипта находящегося в фоне kill номер_процесса

```
bruhe@wibob-X61:~/devops/scripts_4_10$ ps -aux | grep proc_qlogger.sh
bruhe 14995 0.0 0.1 9080 3184 pts/1 S 04:09 0:00 /bin/bash ./proc_qlogger.sh
bruhe 15069 0.0 0.0 8492 788 pts/1 S+ 04:18 0:00 grep --color=auto proc_qlogger.sh
bruhe@wibob-X61:~/devops/scripts_4_10$ jobs
[1]+  Running                  ./proc_qlogger.sh &
bruhe@wibob-X61:~/devops/scripts_4_10$ fg 1
./proc_qlogger.sh
^Z
[1]+  Stopped                  ./proc_qlogger.sh
bruhe@wibob-X61:~/devops/scripts_4_10$ jobs
[1]+  Stopped                  ./proc_qlogger.sh
bruhe@wibob-X61:~/devops/scripts_4_10$ bg 1
[1]+ ./proc_qlogger.sh &
bruhe@wibob-X61:~/devops/scripts_4_10$ jobs
[1]+  Running                  ./proc_qlogger.sh &
bruhe@wibob-X61:~/devops/scripts_4_10$ ps -aux | grep proc_qlogger.sh
bruhe 14995 0.0 0.1 9080 3184 pts/1 S 04:09 0:00 /bin/bash ./proc_qlogger.sh
bruhe 15133 0.0 0.0 8492 876 pts/1 S+ 04:30 0:00 grep --color=auto proc_qlogger.sh
bruhe@wibob-X61:~/devops/scripts_4_10$ kill 14995
bruhe@wibob-X61:~/devops/scripts_4_10$ ps -aux | grep proc_qlogger.sh
bruhe 15136 0.0 0.0 8492 900 pts/1 S+ 04:31 0:00 grep --color=auto proc_qlogger.sh
[1]+  Terminated              ./proc_qlogger.sh
bruhe@wibob-X61:~/devops/scripts_4_10$
```

Лог за время работы скрипта с приостановками

```
05/05/20 04:20:53 225
05/05/20 04:21:52 224
05/05/20 04:22:51 222
05/05/20 04:23:50 223
05/05/20 04:30:09 223
bruhe@wibob-X61:~/tmp$
```

4.10.4 Просмотреть процессы в реальном режиме времени и вывести те, что используют больше всего памяти. Понизить приоритет самого ресурсоёмкого процесса на 2.

использую вывод информации только о процессах текущего пользователя для компактности. Процессы отсортированные по потреблению памяти

```
1 [ | 2.0%] Tasks: 122, 194 thr; 1 running
2 [ | 20.3%] Load average: 0.16 0.08 0.02
Mem[ | 586M/2.39G] Uptime: 10 days, 08:33:26
Swp[ | 0K/0K]

PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command
11167 bruhe 20 0 12376 7256 6412 S 0.0 0.3 0:00.08 /lib/systemd/systemd --user
11276 bruhe 20 0 10912 5828 4008 S 0.0 0.2 0:00.92 -bash
13882 bruhe 20 0 10396 5228 3984 S 0.0 0.2 0:00.45 -bash
14733 bruhe 20 0 10396 5208 3964 S 0.0 0.2 0:00.26 -bash
14631 bruhe 20 0 7740 4192 3184 R 2.0 0.2 1:00.03 htop -u bruhe
15158 bruhe 20 0 11348 3972 3412 S 20.3 0.2 0:27.72 top -u bruhe
11275 bruhe 20 0 11616 3244 2476 S 0.0 0.1 0:01.29 sshd: bruhe@pts/1
13881 bruhe 20 0 11616 3096 2328 S 0.0 0.1 0:00.74 sshd: bruhe@pts/2
14732 bruhe 20 0 11616 3044 2276 S 0.0 0.1 0:00.41 sshd: bruhe@pts/3
11168 bruhe 20 0 32312 2240 16 S 0.0 0.1 0:00.00 (sd-pam)
```

за пущенный в параллельном терминале топ потребляют больше всего процессорного времени, его приоритет и понижаем (для понижения приоритета нужны права суперпользователя)

или в программе htop клавишей F7

```

1  [
2  [|||||
Mem[|||||
Swp[|||||

0.0%] Tasks: 123, 194 thr; 1 running
21.2%] Load average: 0.15 0.15 0.10
585M/2.39G] Uptime: 10 days, 08:45:48
0K/0K]

PID USER      PRI  NI  VIRT   RES   SHR  S  CPU% MEM%   TIME+  Command
11167 bruh        20    0 12376  7256  6412  S   0.0  0.3   0:00.08 /lib/systemd/systemd --user
11276 bruh        20    0 10912  5828  4008  S   0.0  0.2   0:01.01 -bash
13882 bruh        20    0 10396  5228  3984  S   0.0  0.2   0:00.46 -bash
14733 bruh        20    0 10396  5208  3964  S   0.0  0.2   0:00.26 -bash
15158 bruh        18   -2 11348  3972  3412  S  19.0  0.2   2:54.44 top -u bruh
11275 bruh        20    0 11616  3244  2476  S   0.0  0.1   0:01.33 sshd: bruh@pts/1
13881 bruh        20    0 11616  3096  2328  S   0.7  0.1   0:00.91 sshd: bruh@pts/2
14732 bruh        20    0 11616  3044  2276  S   0.0  0.1   0:01.41 sshd: bruh@pts/3
11168 bruh        20    0 32312  2240    16  S   0.0  0.1   0:00.00 (sd-pam)

F1Help F2Setup F3Search F4Filter F5Tree F6SortBy F7Nice F8Nice + F9Kill F10Quit

```

или командой `# renice -n -2 -p 15158` (PID нужного нам процесса)

```

bru@wibob-X61: ~/tmp (ssh)
bru@wibob-X61:~/devops/scripts_4_10$ sudo renice -n -2 -p 15158
15158 (process ID) old priority 0, new priority -2
bru@wibob-X61:~/devops/scripts_4_10$

```

4.10.5 Создать скрипт, который выведет квадрат чисел, введенных в качестве аргументов (позиционные параметры), независимо от их количества.

`##sqrt_unlim.sh`

```

for var in "$@"
do
    echo "$(($var**2))"
done

bru@wibob-X61:~/devops/scripts_4_10$ ./sqrt_unlim.sh 6 4 5 77 88 2 1 45
36
16
25
5929
7744
4
1
2025
bru@wibob-X61:~/devops/scripts_4_10$ ./sqrt_unlim.sh 6 4
36
16
bru@wibob-X61:~/devops/scripts_4_10$

```

4.10.6 Создать скрипт для решения линейного уравнения с помощью функции.

linear equation.sh. Первым делом проверяем не дали ли нам значение для a=0 (чтоб избежать ошибки деления на "0") Далее вычисление с представлением в виде числа с плавающей точкой и каскад операторов, каждый из которых обрабатывает свой случай, а именно:

's/^0\$//'* - удаляем все незначащие нули в десятичной дроби

's/.\$//' - удаляем после первого седа и точку, если она оказалась в конце строки

(результат целое)

'/[^0-9]\/s\/0\.' - для строки у которой перед точкой идут любые не цифры

заменяем "." на "0." (когда в ответе не показывается ведущий ноль в десятичной дроби)

's/^\$/0/' - когда пусто (x=0) заменяем ничего на 0

```
function linear_eq {
    if [ $1 -eq 0 ]; then
        echo "ERROR - It's nonsense!"
    else
        echo "$ ($3-$2) / $1" | bc -l | sed -e 's/^0*$//' | sed -e 's/.$//' | sed -e '/[^0-9]\/s\/0\.'/ | sed -e 's/^$/0/'
    fi
}
echo " x = $(linear_eq $1 $2 $3)"
```

И проверка на разных классах эквивалентности

```
bruha@wibob-X61:~/devops/scripts_4_10$ ./linear_equation.sh 4 5 2
x = -0.75
bruha@wibob-X61:~/devops/scripts_4_10$ ./linear_equation.sh 4 5 1
x = -1
bruha@wibob-X61:~/devops/scripts_4_10$ ./linear_equation.sh 3 5 3
x = -0.6666666666666666
bruha@wibob-X61:~/devops/scripts_4_10$ ./linear_equation.sh 4 5 3
x = -0.5
bruha@wibob-X61:~/devops/scripts_4_10$ ./linear_equation.sh 4 5 5
x = 0
bruha@wibob-X61:~/devops/scripts_4_10$ ./linear_equation.sh 5 2 32
x = 6
bruha@wibob-X61:~/devops/scripts_4_10$ ./linear_equation.sh 0 2 32
x = ERROR - It's nonsense!
bruha@wibob-X61:~/devops/scripts_4_10$
```

4.10.7 Создать скрипт, который регулярно мониторит появление новых пользователей в /etc/passwd и записывает их логины и UID в файл.

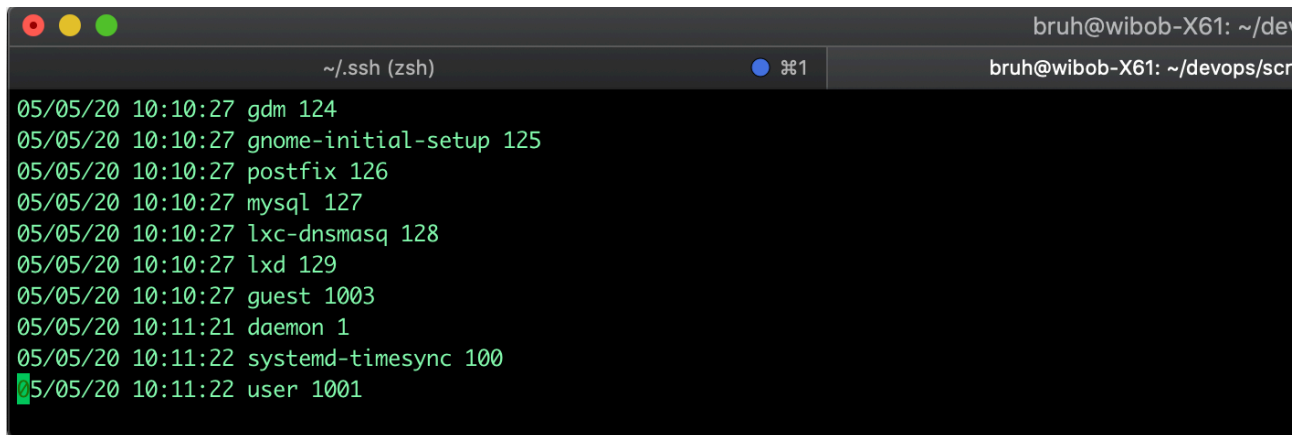
new_user_logger.sh (в файл, кроме логина и UID записывается дата и время, добавления записи в лог, по сути момент обнаружения новых пользователей скриптом;

```
log_file=new_user.log

if [ ! -f $log_file ]
then
    touch $log_file
fi

while read l_passwd; do
    cur_user=`echo "$l_passwd" | cut -d ":" -f1`
    cur_uid=`echo "$l_passwd" | cut -d ":" -f3`
    if [ `awk -v uid=$cur_uid '{if ($4==uid) print "1 "; else;}' $log_file | wc -w` -gt 0 ]
    then
        continue
    #
    # echo `awk -v uid=$cur_uid '{if ($4==uid) print "1 " uid; else;}' $log_file | wc -w`
    # echo "$cur_user old ones"
    #
    # else
    #
    # echo `awk -v uid=$cur_uid '{if ($4==uid) print "1 " uid; else;}' $log_file | wc -w`
    #
    # echo "$cur_user NEW user since previous run"
    # echo "`date +%D %T`" $cur_user $cur_uid >>$log_file
    fi
done </etc/passwd
```

запустил скрипт и чтоб не создавать пользователей, для проверки удалил 3х пользователей и запустил скрип вновь (при повторном запуске эти пользователи были обнаружены и дописаны в конце файла)

A terminal window with a dark background and light green text. The window title bar shows 'bruha@wibob-X61: ~/devops/scr'. The terminal content shows a series of log entries for user creation, including gdm, gnome-initial-setup, postfix, mysql, lxc-dnsmasq, lxd, guest, daemon, systemd-timesync, and a new user. The window has standard macOS window controls (red, yellow, green buttons) in the top left corner.

```
bruha@wibob-X61: ~/devops/scr  
~/.ssh (zsh) 1  
05/05/20 10:10:27 gdm 124  
05/05/20 10:10:27 gnome-initial-setup 125  
05/05/20 10:10:27 postfix 126  
05/05/20 10:10:27 mysql 127  
05/05/20 10:10:27 lxc-dnsmasq 128  
05/05/20 10:10:27 lxd 129  
05/05/20 10:10:27 guest 1003  
05/05/20 10:11:21 daemon 1  
05/05/20 10:11:22 systemd-timesync 100  
05/05/20 10:11:22 user 1001
```