МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение

высшего образования

«КРЫМСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ им. В. И. ВЕРНАДСКОГО»

ФИЗИКО-ТЕХНИЧЕСКИЙ ИНСТИТУТ

Кафедра компьютерной инженерии и моделирования

Утилиты мониторинга производительности в среде Linux

Отчет по лабораторной работе 12

по дисциплине «**Системное программное обеспечение**»

студента 3 курса группы ИВТ-б-о-202

Шор Константина Александровича

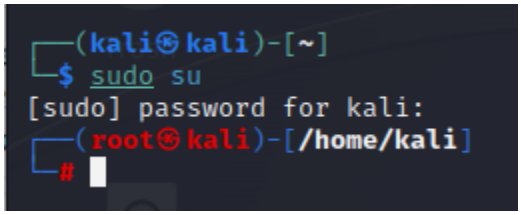
Направления подготовки 09.03.01«Информатика и вычислительная техника»

Симферополь, 2023

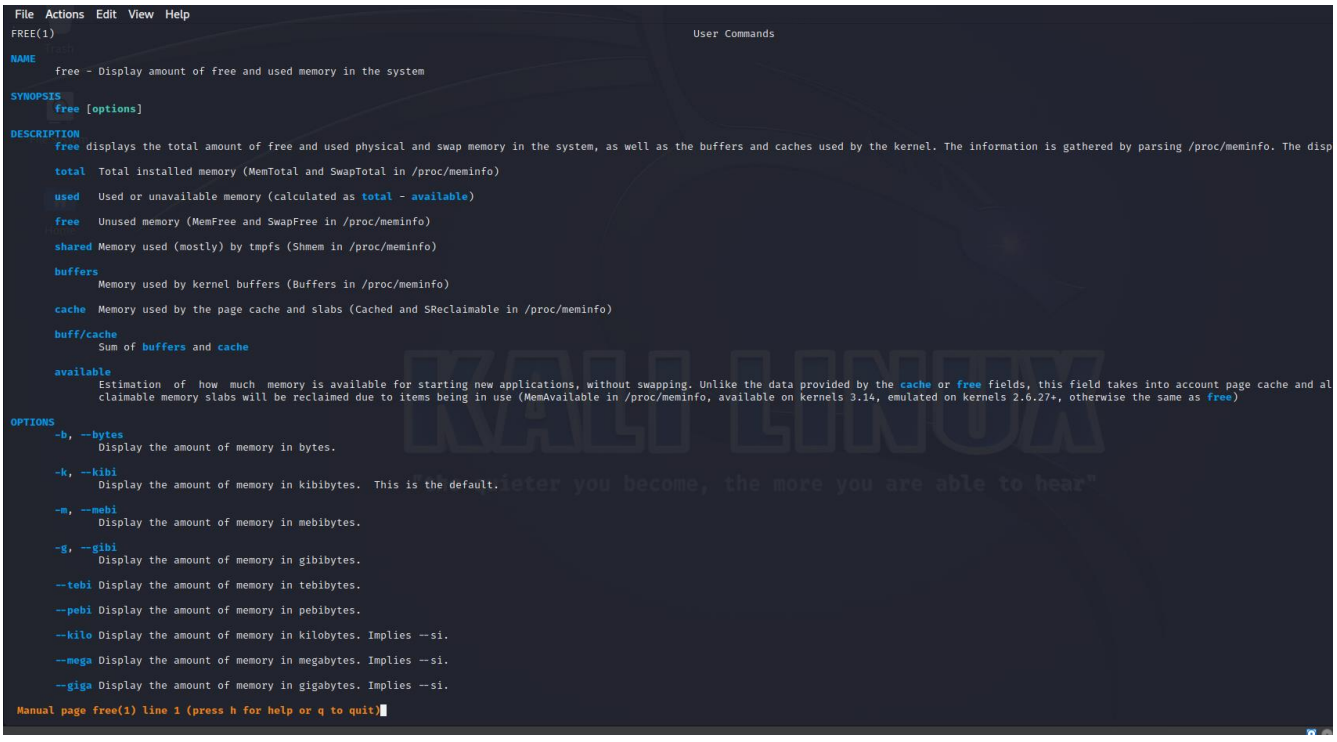
Лабораторная работа №12. Утилиты мониторинга производительности в среде Linux

Цель работы: Получение навыков работы с утилитами диагностирования состояния памяти и выполнения процессов в ос Linux.

1. Войти в систему под рутом



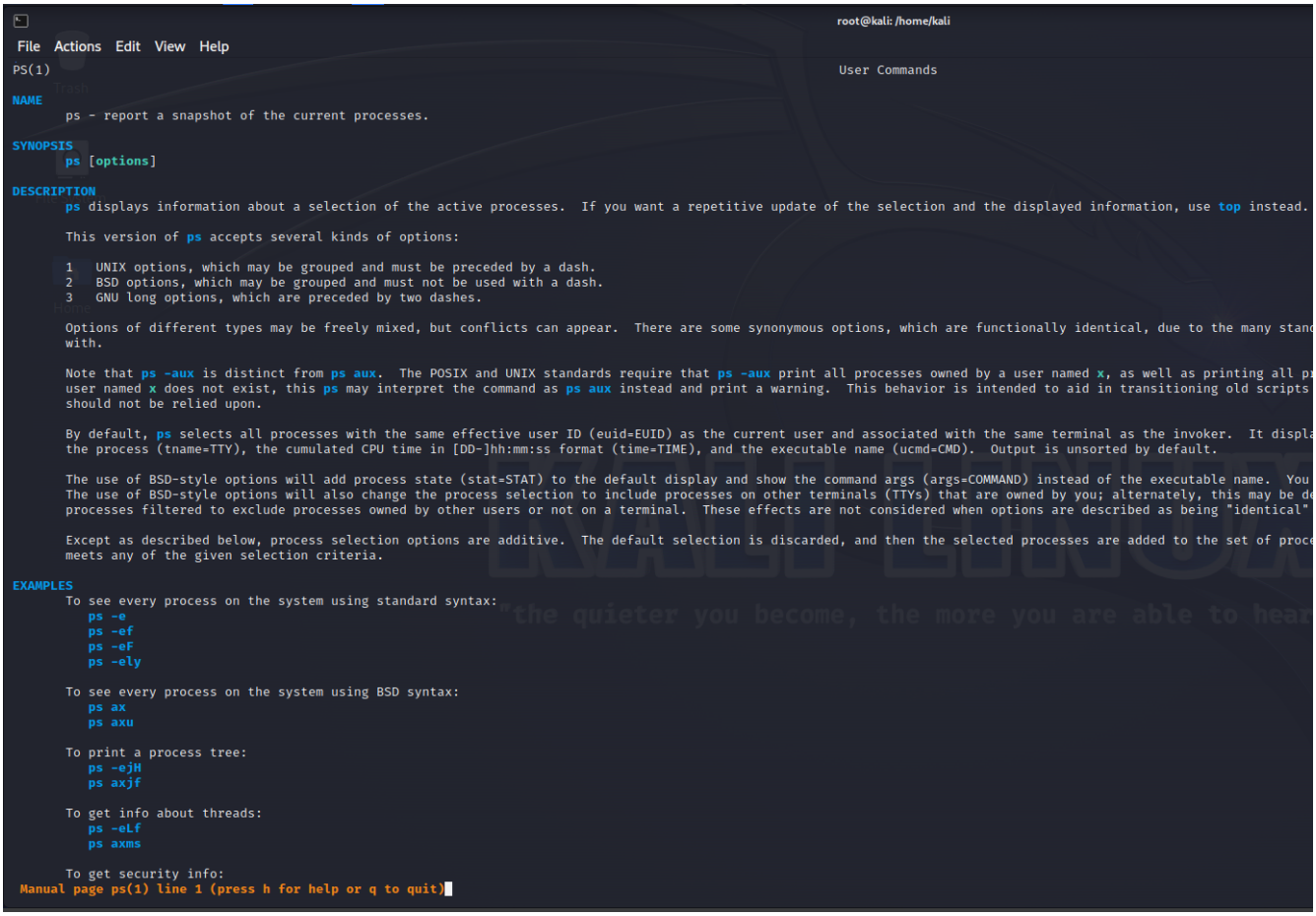
2. Ознакомится с документацией



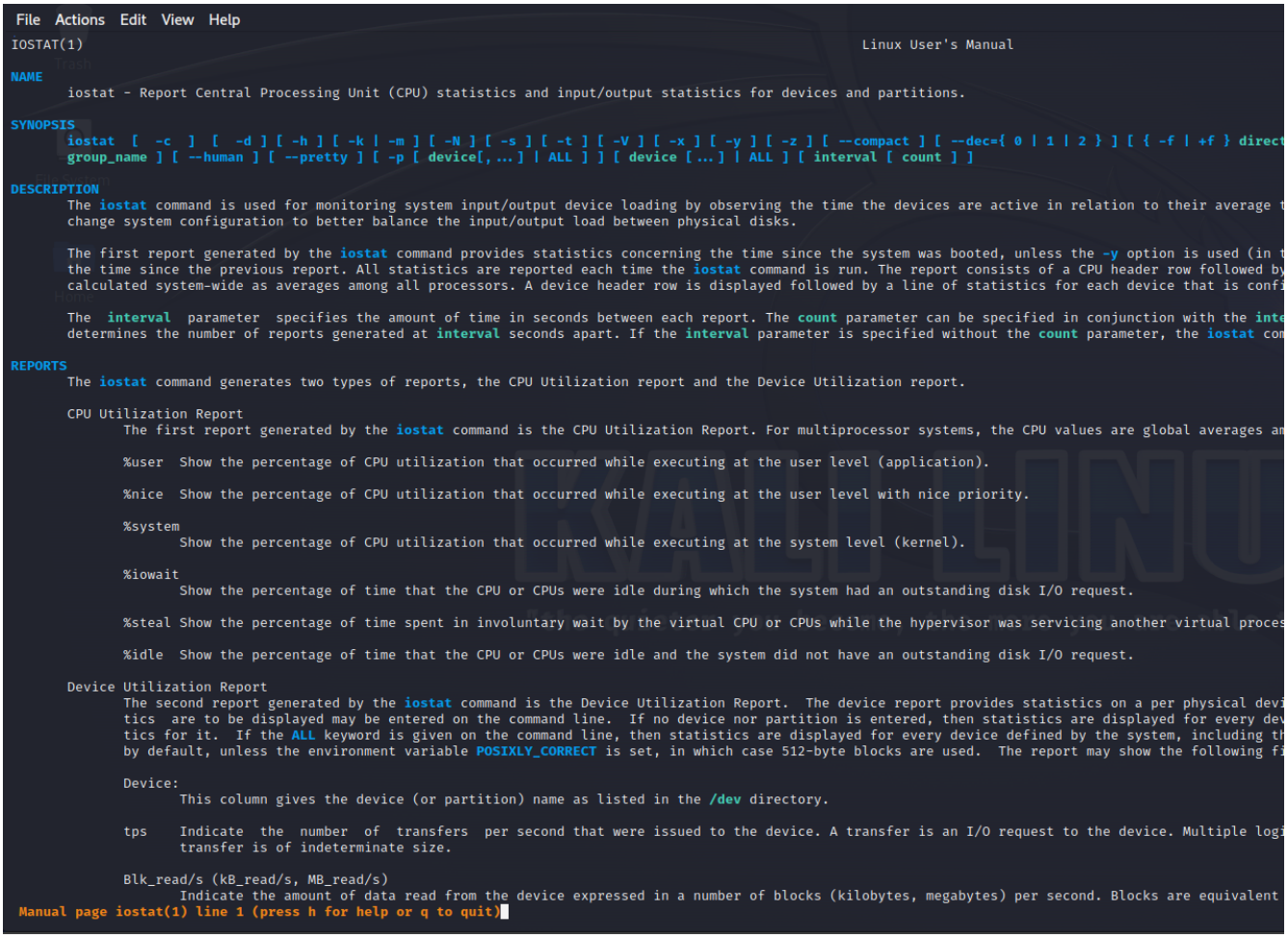
```
File  Actions  Edit  View  Help
VMSTAT(8)                                                    System Administration

NAME
       vmstat - Report virtual memory statistics

SYNOPSIS
       vmstat [options] [delay [count]]

DESCRIPTION
       vmstat reports information about processes, memory, paging, block IO, traps, disks and cpu activity.

       The first report produced gives averages since the last reboot.  Additional reports give information on a sampling period of length delay.  The process and memory reports are instantaneous in eith

OPTIONS
       delay  The delay between updates in seconds.  If no delay is specified, only one report is printed with the average values since boot.

       count  Number of updates.  In absence of count, when delay is defined, default is infinite.

       -a, --active
              Display active and  inactive memory, given a 2.5.41 kernel or better.

       -f, --forks
              The  -f  switch  displays  the  number  of forks since boot.  This includes the fork, vfork, and clone system calls, and is equivalent to the total number of tasks created.  Each process is
              tasks, depending on thread usage.  This display does not repeat.

       -m, --slabs
              Displays slabinfo.

       -n, --one-header
              Display the header only once rather than periodically.

       -s, --stats
              Displays a table of various event counters and memory statistics.  This display does not repeat.

       -d, --disk
              Report disk statistics (2.5.70 or above required).

       -D, --disk-sum
              Report some summary statistics about disk activity.

       -p, --partition device
              Detailed statistics about partition (2.5.70 or above required).

       -S, --unit character
              Switches outputs between 1000 (k), 1024 (K), 1000000 (m), or 1048576 (M) bytes.  Note this does not change the swap (si/so) or block (bi/bo) fields.

       -t, --timestamp
              Append timestamp to each line

       -w, --wide
              Wide output mode (useful for systems with higher amount of memory, where the default output mode suffers from unwanted column breakage).  The output is wider than 80 characters per line.

       -y, --no-first
              Omits first report with statistics since system boot.
 Manual page vmstat(8) line 1 (press h for help or q to quit)
```

```
TOP(1)                                                       User Commands

NAME
       top - display Linux processes

SYNOPSIS
       top [options]

DESCRIPTION
       The  top  program provides a dynamic real-time view of a running system.  It can display system summary information as well as a list of processes or thre
       summary information shown and the types, order and size of information displayed for processes are all user configurable and that configuration can be mad

       The program provides a limited interactive interface for process manipulation as well as a much more extensive interface for personal configuration  --  e
       to throughout this document, you are free to name the program anything you wish.  That new name, possibly an alias, will then be reflected on top's displa

OVERVIEW
   Documentation
       The remaining Table of Contents

           OVERVIEW
              Operation
              Linux Memory Types
           1. COMMAND-LINE Options
           2. SUMMARY Display
              a. UPTIME and LOAD Averages
              b. TASK and CPU States
              c. MEMORY Usage
           3. FIELDS / Columns Display
              a. DESCRIPTIONS of Fields
              b. MANAGING Fields
           4. INTERACTIVE Commands
              a. GLOBAL Commands
              b. SUMMARY AREA Commands
              c. TASK AREA Commands
                 1. Appearance
                 2. Content
                 3. Size
                 4. Sorting
              d. COLOR Mapping
           5. ALTERNATE-DISPLAY Provisions
              a. WINDOWS Overview
              b. COMMANDS for Windows
              c. SCROLLING a Window
              d. SEARCHING in a Window
              e. FILTERING in a Window
           6. FILES
              a. PERSONAL Configuration File
              b. ADDING INSPECT Entries
              c. SYSTEM Configuration File
              d. SYSTEM Restrictions File
           7. ENVIRONMENT VARIABLE(S)
           8. STUPID TRICKS Sampler
              a. Kernel Magic
              b. Bouncing Windows
 Manual page top(1) line 1 (press h for help or q to quit)
```

```
HTOP(1)                                                                 User Commands

NAME
       htop, pcp-htop - interactive process viewer

SYNOPSIS
       htop [-dCFhpustvH]
       pcp htop [-dCFhpustvH] [--host/-h host]

DESCRIPTION
       htop is a cross-platform ncurses-based process viewer.

       It  is  similar  to  top, but allows you to scroll vertically and horizontally, and interact using a pointing device (mouse).  You can observe
       well as view them in a tree format, select multiple processes and act on them all at once.

       Tasks related to processes (killing, renicing) can be done without entering their PIDs.

       pcp-htop is a version of htop built using the Performance Co-Pilot (PCP) Metrics API (see PCPIntro(1), PMAPI(3)), allowing to extend htop to c
       for further details.

COMMAND-LINE OPTIONS
       Mandatory arguments to long options are mandatory for short options too.

       -d --delay=DELAY
              Delay between updates, in tenths of a second. If the delay value is less than 1, it is increased to 1, i.e. 1/10 second. If the delay v

       -C --no-color --no-colour
              Start htop in monochrome mode

       -F --filter=FILTER
              Filter processes by terms matching the commands. The terms are matched case-insensitive and as fixed strings (not regexs). You can sepa

       -h --help
              Display a help message and exit

       -p --pid=PID,PID ...
              Show only the given PIDs

       -s --sort-key COLUMN
              Sort by this column (use --sort-key help for a column list).  This will force a list view unless you specify -t at the same time.

       -u --user=USERNAME|UID
              Show only the processes of a given user

       -U --no-unicode
              Do not use unicode but ASCII characters for graph meters

       -M --no-mouse
              Disable support of mouse control

       --readonly
              Disable all system and process changing features

       -V --version
Manual page htop(1) line 1 (press h for help or q to quit)
```


root@kali: /home/kali
File Actions Edit View Help
PS(1) User Commands
NAME
ps - report a snapshot of the current processes.
SYNOPSIS
ps [options]
DESCRIPTION
ps displays information about a selection of the active processes. If you want a repetitive update of the selection and the displayed information, use top instead.
This version of ps accepts several kinds of options:
1 UNIX options, which may be grouped and must be preceded by a dash.
2 BSD options, which may be grouped and must not be used with a dash.
3 GNU long options, which are preceded by two dashes.
Options of different types may be freely mixed, but conflicts can appear. There are some synonymous options, which are functionally identical, due to the many stand
with.
Note that ps -aux is distinct from ps aux. The POSIX and UNIX standards require that ps -aux print all processes owned by a user named x, as well as printing all p
user named x does not exist, this ps may interpret the command as ps aux instead and print a warning. This behavior is intended to aid in transitioning old scripts
should not be relied upon.
By default, ps selects all processes with the same effective user ID (euid=EUID) as the current user and associated with the same terminal as the invoker. It displa
the process (tname=TTY), the cumulated CPU time in [DD-]hh:mm:ss format (time=TIME), and the executable name (ucmd=CMD). Output is unsorted by default.
The use of BSD-style options will add process state (stat=STAT) to the default display and show the command args (args=COMMAND) instead of the executable name. You
The use of BSD-style options will also change the process selection to include processes on other terminals (TTYs) that are owned by you; alternately, this may be de
processes filtered to exclude processes owned by other users or not on a terminal. These effects are not considered when options are described as being "identical"
Except as described below, process selection options are additive. The default selection is discarded, and then the selected processes are added to the set of proce
meets any of the given selection criteria.
EXAMPLES
To see every process on the system using standard syntax:
ps -e
ps -ef
ps -eF
ps -ely
To see every process on the system using BSD syntax:
ps ax
ps axu
To print a process tree:
ps -ejH
ps axjf
To get info about threads:
ps -eLf
ps axms
To get security info:
Manual page ps(1) line 1 (press h for help or q to quit)

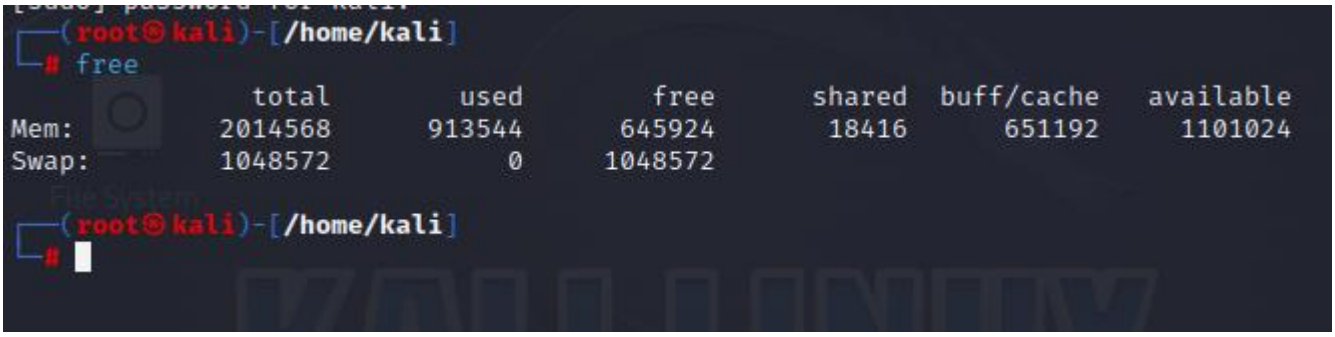

root@kali: /home/kali
File Actions Edit View Help
DSTAT(1)
NAME
dstat - versatile tool for generating system resource statistics
SYNOPSIS
dstat [-afv] [options..] [delay [count]]
DESCRIPTION
Dstat is a versatile replacement for vmstat, iostat and ifstat. Dstat overcomes some of the limitations and adds some extra features.
Dstat allows you to view all of your system resources instantly, you can eg. compare disk usage in combination with interrupts from your IDE control
throughput (in the same interval).
Dstat also cleverly gives you the most detailed information in columns and clearly indicates in what magnitude and unit the output is displayed. Les
Dstat is unique in letting you aggregate block device throughput for a certain diskset or network bandwidth for a group of interfaces, ie. you can s
filesystem or storage system.
Dstat allows its data to be directly written to a CSV file to be imported and used by OpenOffice, Gnumeric or Excel to create graphs.
Note
Users of Sleuthkit might find Sleuthkit's dstat being renamed to datastat to avoid a name conflict. See Debian bug #283709 for more information.
OPTIONS
-c, --cpu
enable cpu stats (system, user, idle, wait), for more CPU related stats also see --cpu-adv and --cpu-use
-C 0,3,total
include cpu0, cpu3 and total (when using -c/--cpu); use all to show all CPUs
-d, --disk
enable disk stats (read, write), for more disk related stats look into the other --disk plugins
-D total,hda
include total and hda (when using -d/--disk)
-g, --page
enable page stats (page in, page out)
-i, --int
enable interrupt stats
-I 5,10
include interrupt 5 and 10 (when using -i/--int)
-l, --load
enable load average stats (1 min, 5 mins, 15mins)
-m, --mem
enable memory stats (used, buffers, cache, free); for more memory related stats also try --mem-adv and --swap
-n, --net
enable network stats (receive, send)
Manual page dstat(1) line 1 (press h for help or q to quit)


File Actions Edit View Help
IOSTAT(1) Linux User's Manual
NAME
iostat - Report Central Processing Unit (CPU) statistics and input/output statistics for devices and partitions.
SYNOPSIS
iostat [-c] [-d] [-h] [-k | -m] [-N] [-s] [-t] [-V] [-x] [-y] [-z] [--compact] [--dec={ 0 | 1 | 2 }] [{ -f | +f } direct
group_name] [--human] [--pretty] [-p [device[,...] | ALL]] [device [...] | ALL] [interval [count]]
DESCRIPTION
The iostat command is used for monitoring system input/output device loading by observing the time the devices are active in relation to their average t
change system configuration to better balance the input/output load between physical disks.
The first report generated by the iostat command provides statistics concerning the time since the system was booted, unless the -y option is used (in t
the time since the previous report. All statistics are reported each time the iostat command is run. The report consists of a CPU header row followed by
calculated system-wide as averages among all processors. A device header row is displayed followed by a line of statistics for each device that is confi
The interval parameter specifies the amount of time in seconds between each report. The count parameter can be specified in conjunction with the inte
determines the number of reports generated at interval seconds apart. If the interval parameter is specified without the count parameter, the iostat com
REPORTS
The iostat command generates two types of reports, the CPU Utilization report and the Device Utilization report.
CPU Utilization Report
The first report generated by the iostat command is the CPU Utilization Report. For multiprocessor systems, the CPU values are global averages am
%user Show the percentage of CPU utilization that occurred while executing at the user level (application).
%nice Show the percentage of CPU utilization that occurred while executing at the user level with nice priority.
%system
Show the percentage of CPU utilization that occurred while executing at the system level (kernel).
%iowait
Show the percentage of time that the CPU or CPUs were idle during which the system had an outstanding disk I/O request.
%steal Show the percentage of time spent in involuntary wait by the virtual CPU or CPUs while the hypervisor was servicing another virtual proces
%idle Show the percentage of time that the CPU or CPUs were idle and the system did not have an outstanding disk I/O request.
Device Utilization Report
The second report generated by the iostat command is the Device Utilization Report. The device report provides statistics on a per physical devi
tics are to be displayed may be entered on the command line. If no device nor partition is entered, then statistics are displayed for every dev
tics for it. If the ALL keyword is given on the command line, then statistics are displayed for every device defined by the system, including th
by default, unless the environment variable POSIXLY_CORRECT is set, in which case 512-byte blocks are used. The report may show the following fi
Device:
This column gives the device (or partition) name as listed in the /dev directory.
tps Indicate the number of transfers per second that were issued to the device. A transfer is an I/O request to the device. Multiple logi
transfer is of indeterminate size.
Blk_read/s (kB_read/s, MB_read/s)
Indicate the amount of data read from the device expressed in a number of blocks (kilobytes, megabytes) per second. Blocks are equivalent
Manual page iostat(1) line 1 (press h for help or q to quit)

3. Запустить поочерёдно каждую утилиту

Free

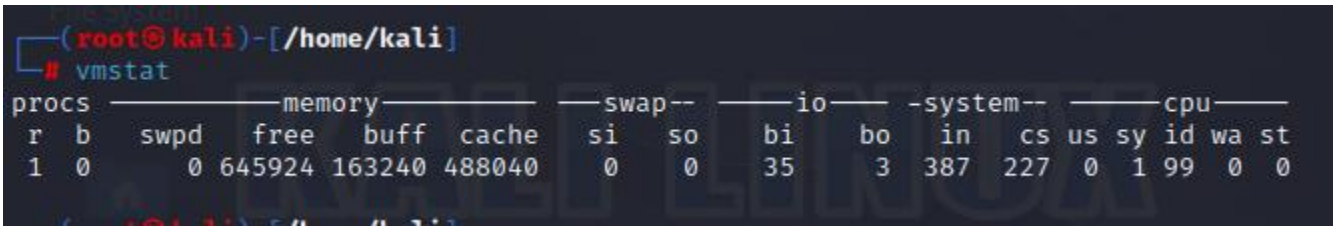
Команда free в Linux используется для отображения информации о использовании оперативной памяти (RAM) на системе. Она выводит статистику по памяти в байтах и представляет ее в виде таблицы с несколькими столбцами.



- total (всего): общий объем оперативной памяти в системе
- used (используется): количество памяти, занятой процессами, кэшем и буферами.
- free (свободно): количество неиспользуемой памяти, доступной для новых процессов
- shared (разделяемая): размер памяти, используемой разделяемыми библиотеками.
- buff/cache (буфер/кэш): объем памяти, используемый ядром операционной системы для буферов и кэша данных
- available (доступно): оценка объема памяти, доступного для новых процессов без подкачки на диск

Vmstat

Команда vmstat в Linux используется для отображения статистики использования виртуальной памяти, процессора, ввода-вывода (I/O) и других системных ресурсов. Она предоставляет информацию о производительности системы в реальном времени.



1. Procs - информация о процессах:
    - r - количество процессов, ожидающих выполнения (в состоянии "runnable").
    - b - количество процессов, блокированных, ожидающих ввода-вывода (I/O).

2. Memory - информация о использовании памяти:
    - swpd - количество использованного подкачки (swap space) в килобайтах.
    - free - количество свободной физической памяти в килобайтах.
    - buff - количество памяти, используемой в буферах ядра в килобайтах.
    - cache - количество памяти, используемой в кэше файловой системы в килобайтах.

3. Swap - информация о использовании подкачки:
    - si - количество данных, считываемых из подкачки в секунду (swap in) в килобайтах
    - so - количество данных, записываемых в подкачку в секунду (swap out) в килобайтах.

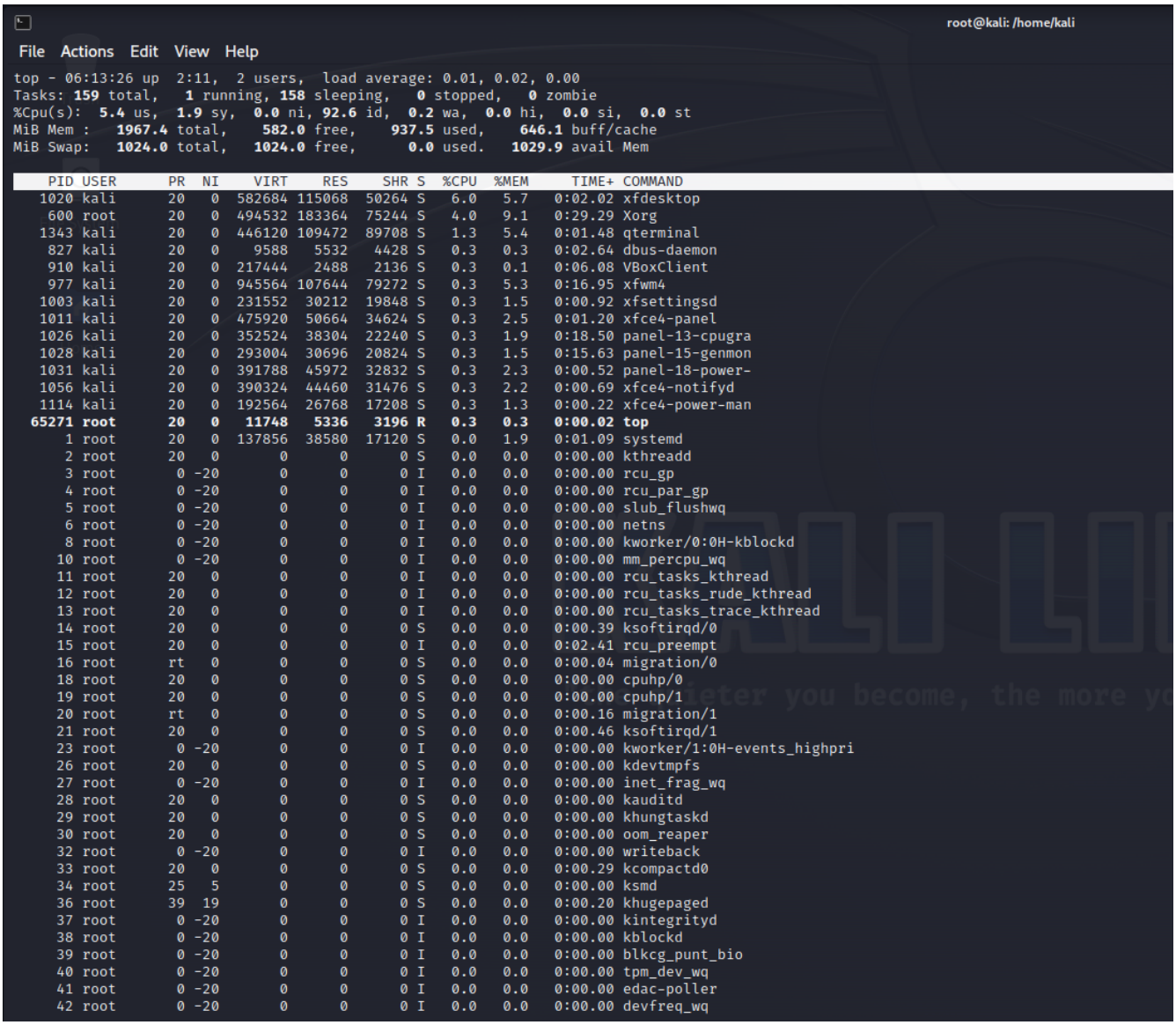
4. IO - информация о вводе-выводе:
    - bi - количество блоков, считываемых с блочных устройств в секунду (блок = 512 байт).
    - bo - количество блоков, записываемых на блочные устройства в секунду.

5. System - информация о системных операциях:
    - in - количество прерываний от устройств в секунду, обрабатываемых ядром
    - cs - количество контекстных переключений (включая переключения между процессами) в секунду.

6. CPU - информация о загрузке процессора:
    - us - процент времени процессора, затраченного на выполнение пользовательских процессов (user).
    - sy - процент времени процессора, затраченного на выполнение системных задач ядра (system).
    - id - процент времени процессора, простаивающего (idle).
    - wa - процент времени процессора, затраченного на ожидание ввода-вывода (I/O wait).
    - st - процент времени процессора, затраченного на выполнение виртуализованных задач (steal, если вы используете виртуализацию).

Top

Команда top в Linux используется для отображения системной информации о процессах, запущенных на компьютере, и их использования ресурсов, таких как ЦПУ, память и т.д. Она обновляет данные в реальном времени, что позволяет отслеживать текущую нагрузку на систему.

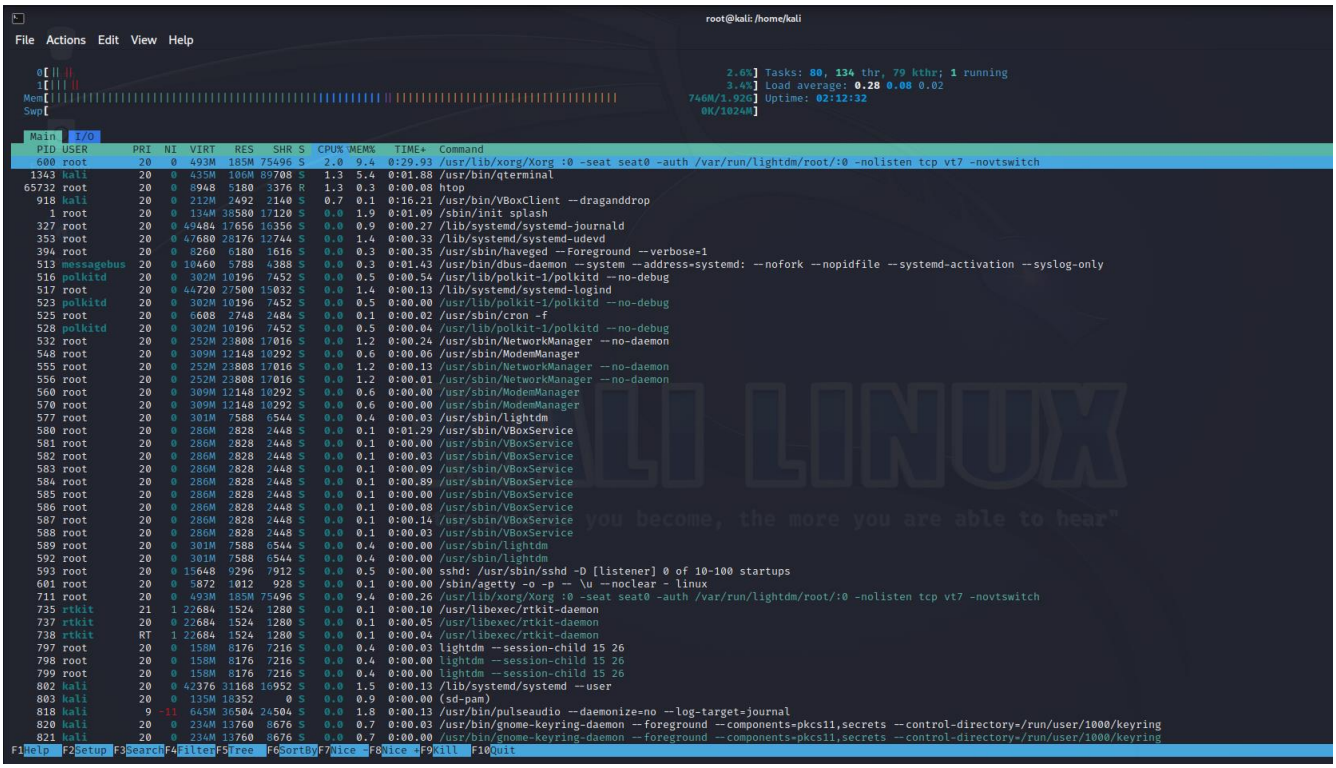


- PID (Process ID): идентификатор процесса
- USER (User): имя пользователя, от имени которого выполняется процесс
- PR (Priority): приоритет процесса.

- NI (Nice value): значение приоритета в виде "доброты" процесса
- VIRT (Virtual memory): общий объем виртуальной памяти, используемый процессом.
- RES (Resident memory): объем физической памяти, используемый процессом в настоящий момент.
- SHR (Shared memory): объем общей памяти, используемой процессом
- S (Status): текущий статус процесса (запущен, спит и т.д.).
- %CPU (CPU usage): процент использования ЦПУ процессом.
- %MEM (Memory usage): процент использования памяти процессом
- TIME+ (CPU time): общее количество процессорного времени, использованного процессом
- COMMAND (Command name): имя команды или программы, запущенной процессом.

Htop

Команда htop в Linux представляет собой интерактивный процесс-менеджер, который позволяет отслеживать системные ресурсы и управлять процессами в реальном времени. Она обеспечивает более детальное и удобное отображение информации о процессах, чем стандартная команда top .

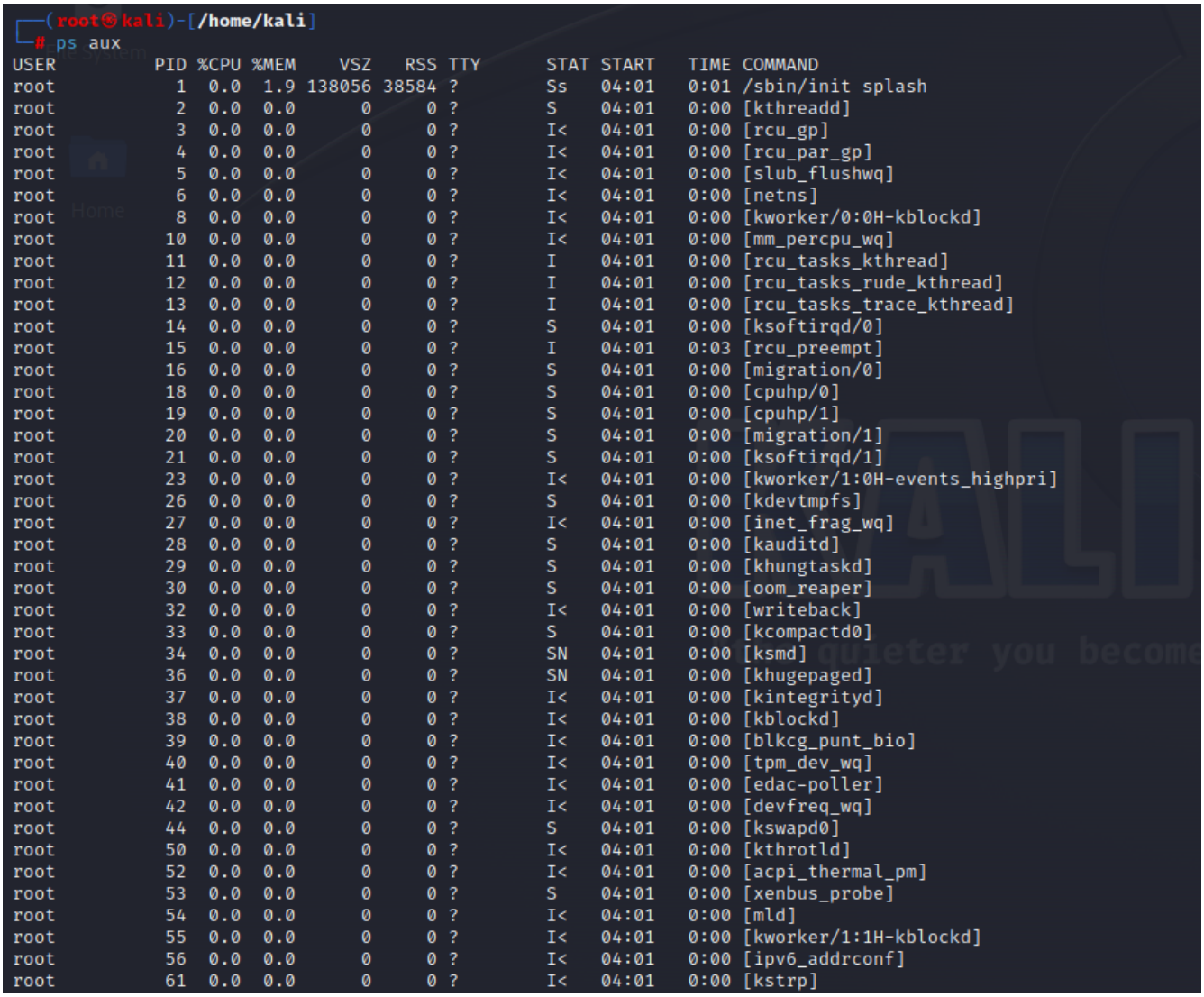


- PID (Process ID) - уникальный идентификатор процесса.
- USER - имя пользователя, от имени которого выполняется процесс.
- PR (Priority) - приоритет процесса.
- NI (Nice value) - значение приоритета, заданное пользователем
- VIRT (Virtual memory) - объем виртуальной памяти, используемой процессом.
- RES (Resident memory) - объем оперативной памяти, используемой процессом.
- SHR (Shared memory) - объем общей памяти, используемой процессом.
- S (Status) - текущий статус процесса (running, sleeping, stopped, etc.).

- %CPU (CPU usage) - процент использования процессорного времени.
- %MEM (Memory usage) - процент использования оперативной памяти.
- . TIME+ (CPU time) - общее процессорное время, затраченное процессом.
- COMMAND (Command name) - имя выполняемой команды или программы.

Ps

Команда ps в Linux используется для вывода информации о текущих процессах в системе



```
┌──(root㉿kali)-[/home/kali]
└─# ps aux
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root           1  0.0  1.9 138056 38584 ?        Ss   04:01   0:01 /sbin/init splash
root           2  0.0  0.0      0     0 ?        S    04:01   0:00 [kthreadd]
root           3  0.0  0.0      0     0 ?        I<   04:01   0:00 [rcu_gp]
root           4  0.0  0.0      0     0 ?        I<   04:01   0:00 [rcu_par_gp]
root           5  0.0  0.0      0     0 ?        I<   04:01   0:00 [slub_flushwq]
root           6  0.0  0.0      0     0 ?        I<   04:01   0:00 [netns]
root           8  0.0  0.0      0     0 ?        I<   04:01   0:00 [kworker/0:0H-kblockd]
root          10  0.0  0.0      0     0 ?        I<   04:01   0:00 [mm_percpu_wq]
root          11  0.0  0.0      0     0 ?        I    04:01   0:00 [rcu_tasks_kthread]
root          12  0.0  0.0      0     0 ?        I    04:01   0:00 [rcu_tasks_rude_kthread]
root          13  0.0  0.0      0     0 ?        I    04:01   0:00 [rcu_tasks_trace_kthread]
root          14  0.0  0.0      0     0 ?        S    04:01   0:00 [ksoftirqd/0]
root          15  0.0  0.0      0     0 ?        I    04:01   0:03 [rcu_preempt]
root          16  0.0  0.0      0     0 ?        S    04:01   0:00 [migration/0]
root          18  0.0  0.0      0     0 ?        S    04:01   0:00 [cpuhp/0]
root          19  0.0  0.0      0     0 ?        S    04:01   0:00 [cpuhp/1]
root          20  0.0  0.0      0     0 ?        S    04:01   0:00 [migration/1]
root          21  0.0  0.0      0     0 ?        S    04:01   0:00 [ksoftirqd/1]
root          23  0.0  0.0      0     0 ?        I<   04:01   0:00 [kworker/1:0H-events_highpri]
root          26  0.0  0.0      0     0 ?        S    04:01   0:00 [kdevtmpfs]
root          27  0.0  0.0      0     0 ?        I<   04:01   0:00 [inet_frag_wq]
root          28  0.0  0.0      0     0 ?        S    04:01   0:00 [kauditd]
root          29  0.0  0.0      0     0 ?        S    04:01   0:00 [khungtaskd]
root          30  0.0  0.0      0     0 ?        S    04:01   0:00 [oom_reaper]
root          32  0.0  0.0      0     0 ?        I<   04:01   0:00 [writeback]
root          33  0.0  0.0      0     0 ?        S    04:01   0:00 [kcompactd0]
root          34  0.0  0.0      0     0 ?        SN   04:01   0:00 [ksmd]
root          36  0.0  0.0      0     0 ?        SN   04:01   0:00 [khugepaged]
root          37  0.0  0.0      0     0 ?        I<   04:01   0:00 [kintegrityd]
root          38  0.0  0.0      0     0 ?        I<   04:01   0:00 [kblockd]
root          39  0.0  0.0      0     0 ?        I<   04:01   0:00 [blkcg_punt_bio]
root          40  0.0  0.0      0     0 ?        I<   04:01   0:00 [tpm_dev_wq]
root          41  0.0  0.0      0     0 ?        I<   04:01   0:00 [edac-poller]
root          42  0.0  0.0      0     0 ?        I<   04:01   0:00 [devfreq_wq]
root          44  0.0  0.0      0     0 ?        S    04:01   0:00 [kswapd0]
root          50  0.0  0.0      0     0 ?        I<   04:01   0:00 [kthrotld]
root          52  0.0  0.0      0     0 ?        I<   04:01   0:00 [acpi_thermal_pm]
root          53  0.0  0.0      0     0 ?        S    04:01   0:00 [xenbus_probe]
root          54  0.0  0.0      0     0 ?        I<   04:01   0:00 [mld]
root          55  0.0  0.0      0     0 ?        I<   04:01   0:00 [kworker/1:1H-kblockd]
root          56  0.0  0.0      0     0 ?        I<   04:01   0:00 [ipv6_addrconf]
root          61  0.0  0.0      0     0 ?        I<   04:01   0:00 [kstrp]
```

USER: Имя пользователя, владеющего процессом.

PID: Идентификатор процесса (Process ID).

. %CPU: Процент процессорного времени, используемого процессом.

%MEM: Процент используемой процессом оперативной памяти

VSZ: Виртуальный размер процесса в килобайтах.

RSS: Размер резидентной набора страниц процесса в килобайтах

TTY: Терминал, связанный с процессом (если есть).

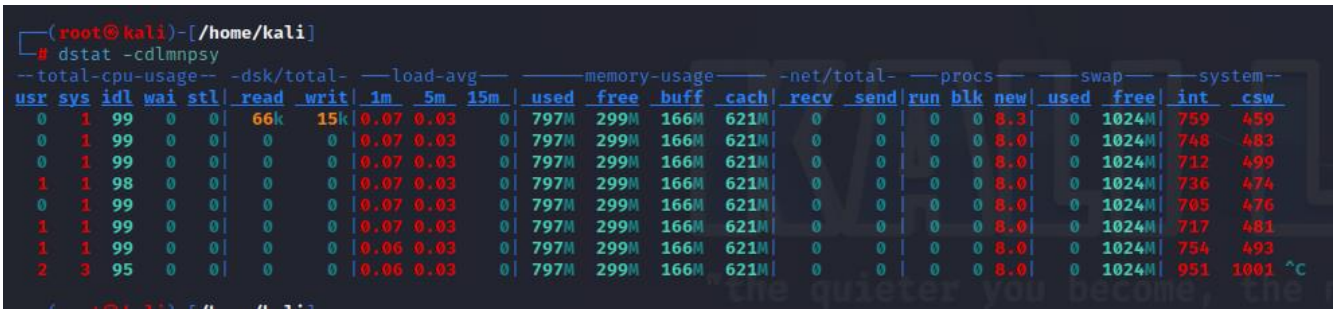
STAT: Состояние процесса (например, S - спящий, R - работает, Z - зомби и т.д.).

START: Время запуска процесса или дата для длительно работающих процессов.
TIME: Общее процессорное время, затраченное процессом.
COMMAND: Имя выполняемой команды или программы.

Флаги:
-A: показывает процессы всех пользователей, а не только текущего пользователя
-U: выводит дополнительную информацию о процессах, включая информацию о владельце процесса (пользователь), использование CPU и памяти, время запуска процесса и команду, которой был запущен процесс.
-X: выводит также процессы, не связанные с терминалом. Это позволяет отображать процессы, запущенные в фоновом режиме или системные процессы.

Dstat

Команда dstat в Linux представляет собой утилиту мониторинга системы, которая отображает различную информацию о производительности системы в реальном времени. Она собирает и отображает данные о процессоре, памяти, дисковом пространстве, сети и других системных ресурсах.



- cpu : Процент использования CPU. Это отображает общую загрузку процессора в процентах.
- dsk : Загрузка диска. Этот столбец показывает количество операций чтения/записи на диске в секунду.
- net : Загрузка сети. Он отображает количество байтов, переданных и полученных через сетевые интерфейсы в секунду.
- load : Загрузка системы. Это отображает среднюю нагрузку системы за последние 1, 5 и 15 минут
- mem : Использование памяти. Он показывает количество использованной и свободной памяти, а также память, используемую для кэширования данных
- swap : Использование области подкачки. Этот столбец показывает использование области подкачки (swap) в системе.
- procs : Информация о процессах. Он показывает количество процессов, запущенных в системе, а также статистику по процессам в режиме реального времени.
- sys : Системные вызовы. Этот столбец отображает количество системных вызовов в секунду
- int : Прерывания. Он показывает количество прерываний в системе.

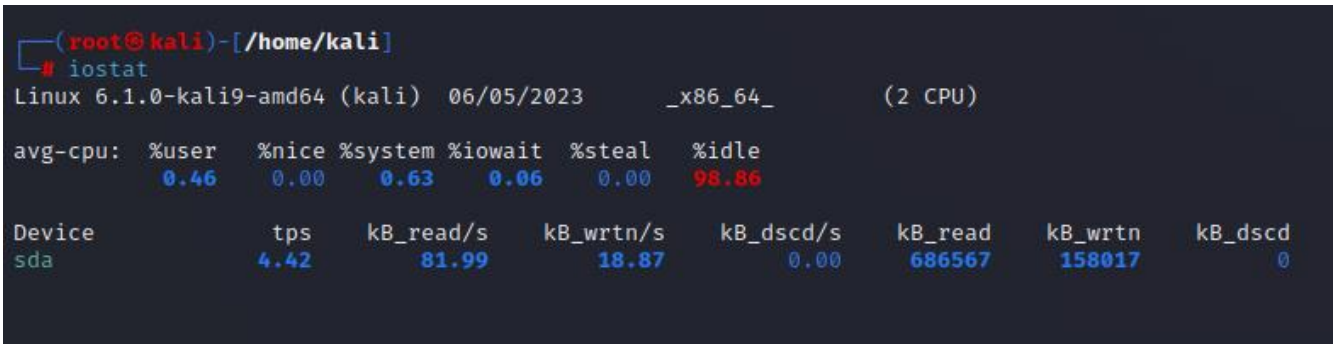
- csw : Переключения контекста. Этот столбец показывает количество переключений контекста в секунду.
- disk : Загрузка диска (расширенная информация). Он отображает загрузку диска по отдельным разделам и устройствам.
- lock : Загрузка блокировок. Этот столбец показывает использование блокировок в системе

Флаги:

- c : Загрузка процессора (CPU)
- d : Загрузка диска (disk).
- l : Загрузка системы (load).
- m : Использование памяти (memory).
- n : Загрузка сети (network).
- p : Информация о процессах (processes).
- s : Использование области подкачки (swap).
- y : Системные вызовы (system).

Iostat

Команда iostat в Linux используется для отображения статистики производительности вводавывода (I/O) системы. Она предоставляет информацию о использовании дисков, дисковых контроллерах, и средств коммуникации ввода-вывода, таких как сетевые интерфейсы.



- avg-cpu : Заголовок столбца, который указывает, что следующие столбцы содержат средние значения использования ЦП в процентах для определенных интервалов времени.
- %user : Процент времени ЦП, затраченного на обработку пользовательских процессов
- %nice : Процент времени ЦП, затраченного на обработку приоритетных процессов.
- %system : Процент времени ЦП, затраченного на обработку системных процессов ядра.
- %iowait : Процент времени, когда ЦП ожидает завершения операций ввода-вывода.
- %steal : Процент времени, когда виртуальный процессор ожидает физического процессора из-за вмешательства других виртуальных процессоров.
- %idle : Процент времени, когда ЦП не занят выполнением задач и ожидает новых задач

- Device : Заголовок столбца, который указывает, что следующие столбцы содержат информацию о конкретных устройствах хранения данных.
- tps : Количество транзакций ввода-вывода в секунду (transactions per second), т.е. количество операций чтения и записи, выполняемых на устройстве в секунду.
- kB_read/s : Количество килобайт, считываемых с устройства в секунду.
- kB_wrtn/s : Количество килобайт, записываемых на устройство в секунду.
- kB_dscd/s : Количество килобайт, отменяемых на устройстве в секунду.
- kB_read : Общее количество килобайт, считанных с устройства с момента его запуска.
- kB_wrtn : Общее количество килобайт, записанных на устройство с момента его запуска
- kB_dscd : Общее количество килобайт, отменяемых на устройстве с момента его запуска.

Вывод: В ходе данной лабораторной работе я вошел в систему под рутом, ознакомился с мануалами команд, предоставленных в теоретической части методички. Запустил каждую команду и описал, под каждой работой команды, весь её функционал, а также самые частые флаги, которые используется вместе с этими командами. Проделав немалую работу, можно сказать, что я освоил все эти команды.