Мониторинг ресурсов

Мониторинг ресурсов (оперативная память, процессоры, диски). Использование утилиты htop. Архивирование файлов. Поиск данных.

**Оглавление**

[Мониторинг ресурсов](#_fp8ep8n9ts5s)

[htop](#_ci4r85tere1w)

[Мониторинг дисков](#_w8y965xk2y0x)

[Команда df](#_5ym4myuea9w4)

[Команда du](#_oxkcqoiqkq3d)

[Архивирование файлов](#_9ywijemii18)

[Команда tar](#_na1ejkr6j2rn)

[Поиск файлов](#_729f9rspti2z)

[Команда find](#_otjom2rh07oc)

[Практическое задание](#_ugb1l81cl7d1)

[Дополнительные материалы](#_3l18frh)

[Используемая литература](#_206ipza)

# 

# Мониторинг ресурсов

Специалисту, работающему с данными, приходится регулярно следить за размером свободной оперативной памяти. Это связано с тем, что в некоторых библиотеках (например, в Pandas) наборы данных часто целиком располагаются в оперативной памяти, что позволяет ускорить работу с ними. Мониторинг ресурсов оперативной памяти необходим как при единовременных задачах, так и при создании программ, работающих на регулярной основе или в режиме реального времени — чтобы избегать их аварийного завершения.

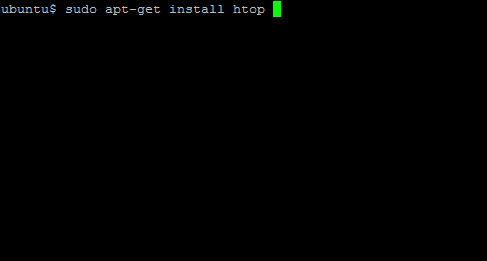
Также необходимо следить за тем, как расходуются ресурсы процессоров. Дело в том, что многие алгоритмы машинного обучения работают в многопоточном режиме, и при одновременном использовании нескольких таких алгоритмов процессорные мощности могут расходоваться не оптимально. От этого модели машинного обучения работают медленнее. В данном уроке мы будем использовать термин «процессор» как синоним понятия «ядро».

При работе с большими объемами данных, даже если оперативная память и процессоры используются оптимизированно, могут возникать ситуации, когда на сервере образовалось большое количество данных и заканчивается место на диске. Предотвращать такие случаи также позволяет мониторинг свободного дискового пространства.

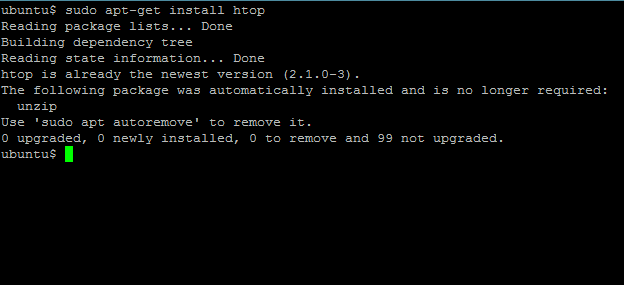
На этом уроке расскажем о том, как можно контролировать использование оперативной памяти, процессоров и дисков, работая в операционной системе Linux.

## htop

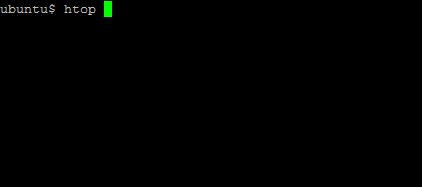
Чтобы следить за использованием оперативной памяти и процессоров, удобно пользоваться утилитой **htop** (по функциям она похожа нестандартную утилиту **top**). Она не входит в стандартный дистрибутив Ubuntu, поэтому нужно будет установить ее с помощью команды **sudo apt-get install htop**:



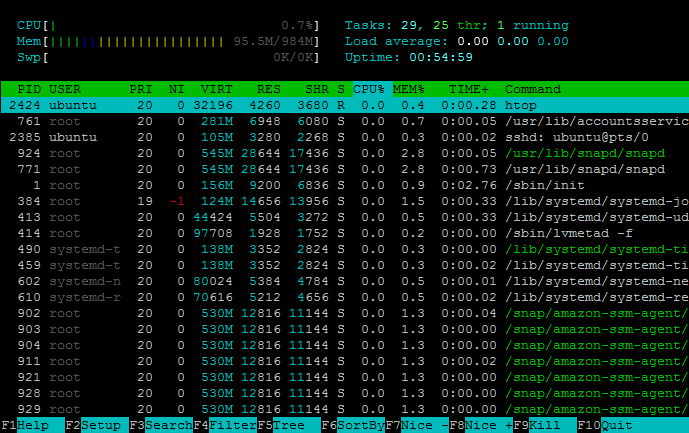
После запуска команды утилита **htop** будет установлена:



Запустим утилиту с помощью команды **htop** либо **sudo htop**:



Появится панель **htop** с информацией об используемых ресурсах и работающих процессах:



Здесь мы видим, что CPU используется на 0,7 %. В данном инстансе только один процессор, а если бы их было несколько, то они были бы пронумерованы по порядку. Зеленым цветом в диаграмме, показывающей загруженность CPU, обозначается часть времени процессора, занятая процессами с нормальным приоритетом. Остальных цветов на данной диаграмме нет, но о них тоже расскажем. Синим обозначается часть, занятая процессами с низким приоритетом. Красным — процессы с приоритетом ядра. Желтым — время процессора, потраченное на виртуализацию либо невольно «украденное» другими пользователями, которые работают в других виртуальных серверах, но на тех же физических ресурсах, и активно нагружают процессоры.

Далее располагается информация об использовании оперативной памяти (**Mem**): 95,5 Мб из 984 Мб. (В данном примере используется инстанс **t2.micro**, имеющий довольно скромные ресурсы по сравнению с инстансами, которые обычно используют в рабочих условиях. Тем не менее для изучения основных принципов работы в **bash** такой инстанс вполне подходит.)

Мы видим, что диаграмма, отображающая использование оперативной памяти, состоит из трех цветов: зеленого, синего и желтого.

Зеленый цвет показывает используемую оперативную память, синий — буферы и желтый — системный кеш.

Раздел **Swp** показывает использование **swap** (своп) — в данном случае оно нулевое.

Далее располагается информация о процессах. Наиболее информативными здесь будут поля:

* **PID** — id процесса;
* **USER** — пользователь, от чьего имени запущен процесс;
* **PRI** — текущий приоритет процесса;
* **RES** — количество резидентной, то есть не перемещаемой в **swap**, памяти в килобайтах;
* **SHR** — количество разделяемой, то есть **shared**, памяти программы в килобайтах (памяти, которая может быть использована другими приложениями);
* **CPU %** — использование процессора в процентах;
* **MEM %** — использование процессом памяти в процентах;
* **TIME+** — время работы процесса;
* **Command** — команда, которой был запущен процесс.

Более подробное описание программы **htop** можно прочитать по ссылке: <http://linux-bash.ru/menusistem/79-htop.html>.

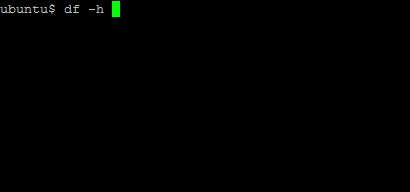
Чтобы выйти из программы, нужно нажать F10.

## Мониторинг дисков

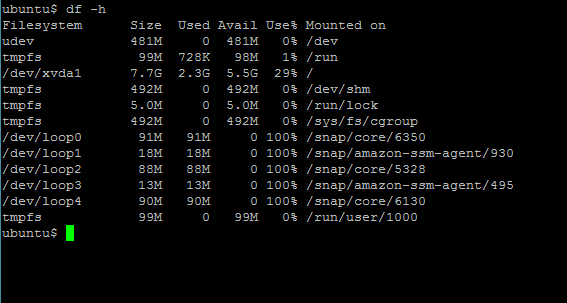
### Команда df

Для мониторинга дисковых ресурсов можно пользоваться утилитами **df** и **du**.

Например, вот так можно в удобном виде получить информацию об использовании дискового пространства с помощью программы **df –h** (в данном случае **–h**, сокращение от слова ***human***, используется для более привычного человеку отображения):



После запуска программы можно увидеть название устройства (диска), общий размер, используемое пространство в абсолютном значении, доступное пространство, используемое пространство в процентах и путь, к которому прикреплено устройство.



Например, на данном скриншоте устройство **/dev/xvda1** имеет размер 7,7 Гб, из них использовано 2,3 Гб, доступно 5,5 Гб, использовано 29 %, диск закреплен за корневой директорией (**“/”**).

### Команда du

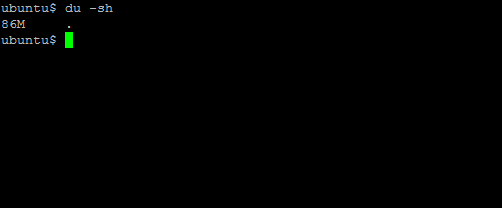
С помощью команды **du** по умолчанию можно узнать размеры всех файлов и папок, находящихся в текущей директории. Команда **du –h** покажет их размеры в удобном для человека формате.

Если нужно посмотреть суммарный объем файлов и папок, находящихся в текущей директории, следует запустить команду **du –s**.

Запустим команду **du –sh**, показывающую суммарный объем файлов и папок в текущей директории (в данном примере это **/home/ubuntu**):



После запуска команды можно увидеть, что суммарный объем составляет 86 Мб.

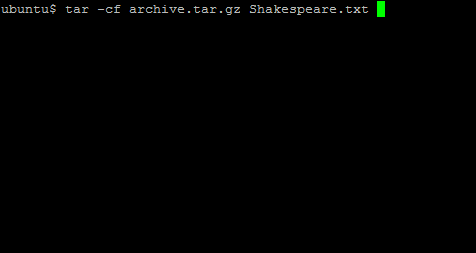


# Архивирование файлов

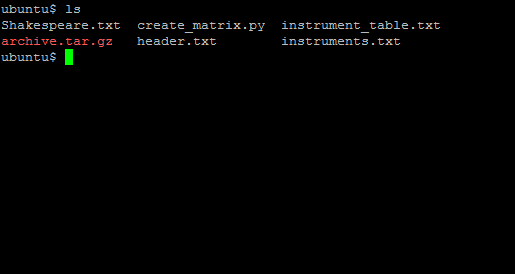
## Команда tar

С помощью программы **tar** можно сжимать данные (при этом выбирать разные алгоритмы сжатия информации), а также объединять несколько файлов в архив.

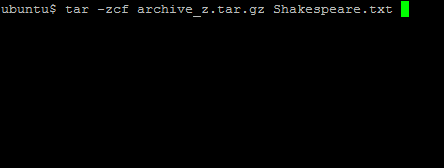
Чтобы создать архив под названием **archive.tar.gz**, в котором будет содержаться файл **Shakespeare.txt** (с ним мы уже работали в домашней директории пользователя **ubuntu**), запустим команду **tar –cf archive.tar.gz Shakespeare.txt**. Опция **c** (от слова ***create***) в этой команде означает, что нужно создать архив, а опция **f** обозначает файл для архива:



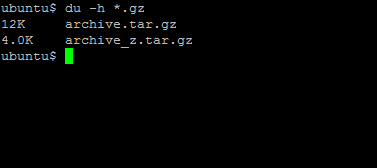
Посмотрим содержимое домашней директории — там появился файл **archive.tar.gz**:



Чтобы создать сжатый архив, нужно помимо опций **c** и **f** указать опцию **z**. И тогда команда запустится как **tar –zcf archive\_z.tar.gz Shakespeare.txt** (здесь мы создаем файл с названием **archive\_z.tar.gz**):



Посмотрим на размер файлов, используя команду **du \*.gz**:



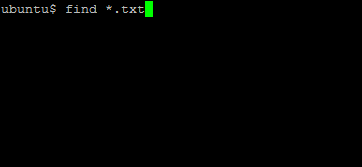
Видно, что файл **archive\_z.tar.gz** в 3 раза меньше **archive.tar.gz**.

Извлечь файлы из архива можно с помощью команды **tar –xf archive.tar.gz**. В этой команде вместо опции **c**, использованной при создании архива, применяется опция **x** (от слова **extract**).

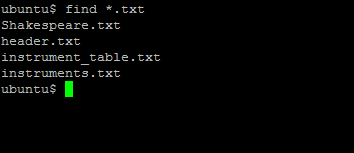
# Поиск файлов

## Команда find

Для поиска файлов удобно использовать команду **find**. Например, можно найти все файлы с расширением **.txt** в текущей директории с помощью команды **find \*.txt**:



Результат будет выглядеть так:



# Практическое задание

1. Запустить **htop** и посмотреть, сколько процессоров и оперативной памяти есть на сервере.
2. Найти все программы с расширением **.py**.
3. \* Создать и запустить программу на Python, выводящую числа от 0 до 100 включительно.

Запустить **htop** во время выполнения программы и найти выполняемую программу в списке процессов, используя поиск по ключевому слову ***python*** (использовать средства поиска **htop**).

# Дополнительные материалы

1. [htop и многое другое на пальцах](https://habr.com/ru/post/316806/).
2. [Поиск файлов с помощью find](http://www.opennet.ru/docs/RUS/linux_base/node149.html).
3. [18 примеров команды tar в Linux](http://blog.sedicomm.com/2018/10/10/18-primerov-komand-tar-v-linux/).

# Используемая литература

Для подготовки данного методического пособия были использованы следующие ресурсы:

1. [HTOP — монитор процессов](http://linux-bash.ru/menusistem/79-htop.html).
2. [htop — продвинутый консольный монитор процессов](https://studylinux.ru/htop.html).
3. [Команда tar в Linux](https://losst.ru/komanda-tar-v-linux).