

Как узнать что занимает оперативную память на сервере?

Category: awk,cli commands,GNU/Linux,memory,ps,Информация о системе
2025-01-29

Задача.

Вывести ТОП-10 процессов с наибольшим использованием памяти на хосте?

Выполнение задачи.

Для этого существует следующий скрипт, который можно запустить в **bash** консоли:

```
# ps axo rss,comm,pid | awk '{ proc_list[$2] += $1; } END { for  
(proc in proc_list) { printf("%d\t%s\n", proc_list[proc],proc); } }'  
| sort -n | tail -n 10 | sort -rn | awk '{ $1/=1024;printf  
"%0.0fMB\t", $1 } { print $2 }'
```

Ответ:



Разберем команду по частям, чтобы понять, что она делает:

~~~~~ 1 ~~~~~

```
ps axo rss,comm,pid
```

- ps – это команда для отображения информации о текущих процессах.
- axo и
  - a – показывает все процессы, запущенные пользователями.
  - x – показывает процессы, не привязанные к терминалу.
  - o – позволяет указать, какие поля следует показывать.

В данном случае указываются поля `rss` (размер резидентной памяти в килобайтах), `comm` (название команды) и `pid` (идентификатор процесса).

Если у вас работает, к примеру, `php-fpm`, то у него может быть сотни процессов, так что сама по себе эта команда малоинформативна, так как генерирует огромный список, его и обрабатываем.

Эта часть команды выводит резидентную память, названия команд и идентификаторы процессов.

~~~~~ 2 ~~~~~

```
awk '{ proc_list[$2] += $1; } END { for (proc in proc_list) {  
printf("%d\t%s\n", proc_list[proc],proc); } }'
```

awk – это язык программирования для обработки текстовых файлов и потоковых данных.

- { proc_list[\$2] += \$1; } – здесь мы создаем ассоциативный массив proc_list, где ключом является название процесса (\$2), а значением – суммированный rss размер (\$1). Эта часть добавляет RAM, использованную процессом, к соответствующему процессу.

Таким образом мы создаем словарь из названий процессов и в этом словаре сразу же суммируем rss всех процессов с одним и тем же именем, то есть записываем примерно следующее: proc_list = ([php-fpm]=51224, [mysql]=31441) и так далее.

- END { for (proc in proc_list) { printf("%d\t%s\n", proc_list[proc],proc); } } – в блоке END мы проходим по всему массиву и выводим общее значение rss памяти для каждой программы.

Получается, что в цикле перебирает все названия процессов в словаре и выводит их по одному в каждой строке. В данном случае proc_list[proc] будет выводить rss процесса, proc – его название, конструкция "%d\t%s\n" определяет формат вывода: %d – десятичное число, \t – табуляция, %s – строка, \n – переход на новую строку.

~~~~~ 3 ~~~~~

```
sort -n
```

- Сортирует вывод в числовом порядке по размеру rss.

~~~~~ 4 ~~~~~

```
tail -n 10
```

- Берёт последние 10 строк (то есть 10 процессов с наибольшим использованием памяти).

~~~~~ 5 ~~~~~

```
sort -rn
```

- Сортирует ранее отобранные 10 процессов в обратном порядке (по убыванию), так чтобы наибольшие значения были первыми.

~~~~~ 6 ~~~~~

```
awk '{$1/=1024;printf "%.0fMB\t",$1}{print $2}'
```

awk – это язык программирования для обработки текстовых файлов и потоковых данных.

- `$(1/=1024;` – преобразуем первую колонку, представляющую память (в килобайтах), в мегабайты, деля на 1024. (`$(1/=1024)` – здесь мы просто причесываем вывод, деля `rss` на 1024, чтобы перевести в мегабайты и их же дописываем в конце.
- `printf "%.0fMB\t",$1` – форматирует вывод, чтобы отобразить результат в мегабайтах без десятичных знаков. `%.0f` – округление до целого, `\t` – добавляет табуляцию. Можете это убрать, если вам не нужно.
- `{print $2}` – выводит название процесса в том же порядке.

Оригиналы источников информации.

1. [t.me](https://t.me/ServerAdmin) «ServerAdmin.ru канал Telegram».