

Лабораторная работа 6

КОНТРОЛЬ РЕСУРСОВ. ПЛАНИРОВАНИЕ ЗАДАЧ.

Цель работы

Познакомиться с утилитами предоставляющими информацию о состоянии системы. Научиться создавать запланированные задания.

1. Теоретические сведения

1.1. Получение информации о системе

Многие утилиты, позволяющие осуществлять контроль над ресурсами системы, получают свою информацию из файлов, содержащихся в каталоге виртуальной файловой системы **/proc**. Ядро помещает в данный каталог данные о процессах, имеющихся в системе, статистические сведения, генерируемые ядром, настраиваемые параметры ядра и т.д. Некоторые ключевые файлы перечислены в Таблице 1.

Таблица 1. Некоторые источники информации о системе в каталоге /proc

Файл	Описание
/proc/cpuinfo	Информация о центральном процессоре.
/proc/meminfo	Информация об использовании памяти.
/proc/diskstats	Дисковые устройства и статистика их использования
/proc/version	Версия ядра
/proc/mounts	Информация о смонтированных файловых системах.

Одной из утилит для отслеживания информации о производительности является **vmstat**. Вывод команды представляет собой отчет о состоянии системы получаемый с заданным интервалом времени. Например, получить отчет, состоящий из 8 строк, содержащих статистику, собранную с интервалом в 2 секунды можно командой:

\$ vmstat 2 8

Вывод содержит информацию о готовых к выполнению и спящих процессах, использовании оперативной памяти, подкачки, дисковых операциях, количестве переключений контекста, статистики использования

центрального процессора. Подробнее о формате вывода можно узнать на страницах справочного руководства.

Информацию о средней загрузке системы за период 5с, 10с и 15с, а также время непрерывной работы системы, можно получить с помощью утилиты **uptime**.

\$ uptime

В системе Linux память имеет страничную организацию. Процессам предоставляются виртуальные страницы в пределах имеющейся оперативной памяти и области подкачки (*swap*). Ядро старается хранить часто используемые станицы в оперативной памяти, т.к. доступ к информации, сохранённой на жестком диске занимает больше времени.

Чтобы определить размер области подкачки можно использовать команду **swapon** с опцией **-s**.

\$ swapon -s

Получить общую информацию об имеющейся физической и виртуальной памяти в системе можно с помощью команды **free**. Помимо этого, вывод содержит информацию о размере буферов системы.

\$ free -h

Получить информацию об использовании дискового пространства в системе можно с помощью утилиты **df**. Вывод команды содержит данные по всем смонтированным на данный момент файловым системам с указанием процента использованного пространства и точки монтирования каждой из них. Опция **-h** выполняет приводит форматирование вывода к удобному для пользователя виду.

\$ df -h

Чтобы узнать размер не всего размера в целом, а какой-либо директории можно применить утилиту **du**. Первоначальный вывод команды содержит размеры всех вложенных директорий, имеется возможность управлять глубиной вложенности. Также, как и в случае с командой **df** может применяться опция **-h**.

\$ du -sh

2 Планирование повторяющихся задач

Для большинства задач, стоящих перед системными администраторами характерно периодическое выполнение. Для удобства составления расписания пользовательских задач в операционной системе имеется служба **cron**.

Задания планировщика **cron** построчно перечислены в специальном **crontab** файле. Записи в файле имеют следующий формат:

```
10 15 * * * /home/user/my_script.sh
```

Где пять полей, разделенных пробелами, означают числовые представления минут, часов, дней месяца, месяца в году и дня недели соответственно. Символ «*» соответствует любому значению. Символ «/» служит для указания дополнительной периодичности задания. Например, «*/3» в первом поле означает «каждые 3 минуты». В приведенном выше примере пользовательский скрипт **my_script.sh** будет выполняться каждый день в 15 часов 10 минут.

Каждый пользователь может иметь свой файл **crontab**, чтобы сообщить системе имя файла необходимо выполнить команду:

```
$ crontab filename
```

Вывод имеющихся заданий выполняется данной утилитой с опцией -l. Очистка списка заданий выполняется командой **crontab** с опцией -r. Редактирование имеющегося файла заданий возможно текстовым редактором с использованием опции -e, например:

```
$ crontab -e
```

После добавления файла задания или изменения имеющегося файла происходит проверка синтаксиса.

3 Порядок выполнения лабораторной работы

1. Прочитайте теоретический материал по лабораторной работе.
2. Ознакомьтесь с содержимым файлов, приведенных в Таблице 1. Изучите для упомянутых в тексте работы команд страницы справочного руководства.
3. Получите информацию о версии ядра вашей операционной системы, модели и частоте центрального процессора, количестве процессорных ядер.
4. С помощью команды **vmstat**, в течении 30с с интервалом в 3с, собирайте статистику об использовании ресурсов системы. Посчитайте среднее количество переключений контекста ядра в секунду на заданном интервале времени.

5. Получите информацию о средней загрузке процессора в течение последних 15с.
6. Опишите текущее состояние страниц памяти, доступных в вашей системе.
7. Опишите текущее состояние разделов жестких дисков, доступных в вашей системе.
8. Получите информацию о размере вашего домашнего каталога, с помощью команд, изученных ранее получите список 3 самых больших каталогов в вашей домашней директории.
9. Создайте задание для **cron**, согласно которому каждую минуту в файл `~/memory/stat` будет добавляться информация о текущем состоянии памяти, без учета размера подкачки и заголовка.
10. Создайте задание для **cron**, согласно которому каждые 3 минуты файл `~/memory/stat` будет упаковываться в архив.
11. После выполнения работы удалите все записи из **crontab** файла.

3. Контрольные вопросы

1. Как получить информацию о состоянии памяти?
2. Как получить информацию о доступном дисковом пространстве?
3. Как происходит работа с файлами заданий планировщика cron?
4. Какая информация содержится в директории `/proc`?