Министерство образования и науки Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

«Пермский национальный исследовательский политехнический университет»

Электротехнический факультет

Кафедра «Информационные технологии и автоматизированные системы»

Направление 09.03.04 – «Программная инженерия»

Дисциплина: «Администрирование информационных систем»

Профиль: «Разработка программно-информационных систем»

Семестр 5

ОТЧЕТ

по лабораторной работе №6

Тема: «Контроль ресурсов. Планирование задач»

Выполнила: студент группы РИС-22-1б

Поважный В.Е. \_\_\_\_\_\_\_\_\_\_\_

Проверил: ст. преподаватель кафедры ИТАС

Шереметьев В. Г. \_\_\_\_\_\_\_\_\_\_\_

Дата \_\_\_\_\_\_

Пермь, 2024

**Цель работы**

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

# Теоретические сведения

* 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. |
| /рrос/diskstats | Дисковые устройства и статистика их использования |
| /рrос/version | Версия ядра |
| /рrос/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 загруженности 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 помощь 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 применятся 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.

# Порядок выполнения лабораторной работы

1. Получите информацию о версии ядра вашей операционной системы, модели и частоте центрального процессора, количестве процессорных ядер.
2. 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.
3. Получите информацию о средней загруженности процессора в течении последних 15с.
4. Опишите текущее состояние страниц памяти, доступных в вашей системе.
5. Опишите текущее состояние разделов жестких дисков, доступных в вашей системе.
6. Получите информацию о размере вашего домашнего каталога, с помощью команд, изученных ранее получите список 3 самых больших каталогов в вашей домашней директории.
7. 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.

1. Создайте задание для **cron**, согласно которому каждые 3 минуты файл

~/memory/stat будет упаковываться в архив.

1. После выполнения работы удалите все записи из **crontab** файла.

**Ход работы**

1. Получение информации о версии ядра, модели и частоте процессора, количестве ядер:

* uname -r - версия ядра
* lscpu | grep 'Model name' - модель процессора
* grep 'MHz' /proc/cpuinfo | head -n 1 - частота процессора
* lscpu | grep '^CPU(s):' - количество ядер

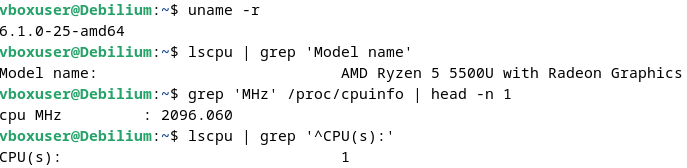


Рис. 1 – Получение информации о версии ядра

1. Сбор статистики об использовании ресурсов с помощью vmstat:

* vmstat 3 10

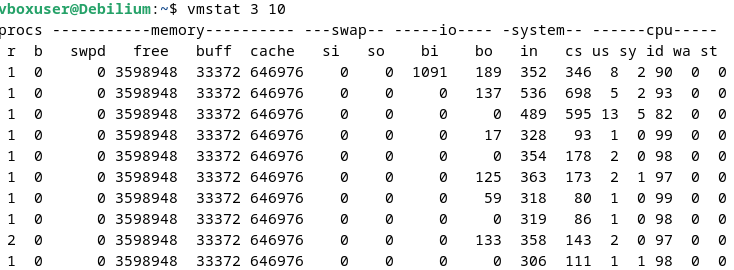


Рис. 2 – Сбор статистики

Среднее значение равно: **250,3**

1. Получение средней загруженности процессора за последние 15 секунд:

* uptime



Рис. 3 – Получение средней загруженности процессора

1. Информация о состоянии страниц памяти:

* free -h - текущее состояние памяти
* vmstat - статистика о памяти и страницах

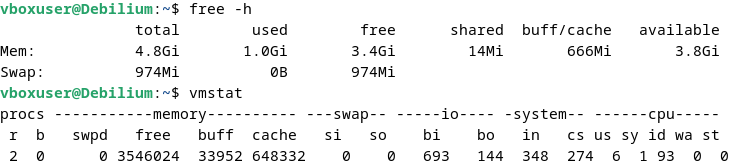


Рис. 4 – Информация о состоянии страниц памяти

1. Информация о состоянии разделов жесткого диска:

* df -h - список разделов и свободного места

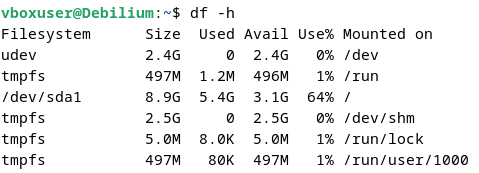


Рис. 5 – Информация о состоянии разделов жесткого диска

1. Получение размера домашнего каталога и списка самых больших папок:

* du -sh ~ - размер домашней директории
* du -ah ~ | sort -rh | head -n 3 - 3 самые большие каталоги

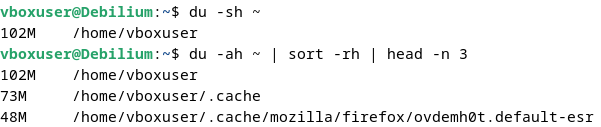


Рис. 6 – Получение размера домашнего каталога

1. Создание задания для cron, добавляющего информацию о состоянии памяти каждую минуту:

cron — это служба, которая позволяет выполнять команды и скрипты по расписанию.

* crontab –e
* \* \* \* \* \* free | grep -v Swap >> ~/memory/stat

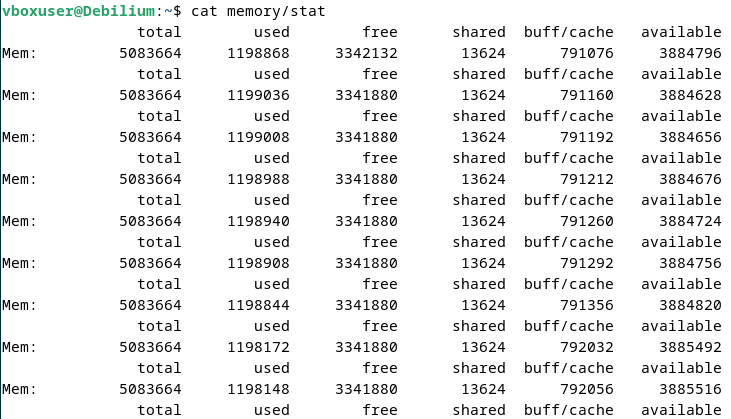


Рис. 7 – Получение размера домашнего каталога

1. Создание задания для cron, упаковывающего файл каждые 3 минуты:

* \*/3 \* \* \* \* tar -czf ~/memory/stat\_$(date +\%Y\%m\%d\%H\%M).tar.gz ~/memory/stat



Рис. 8 – Получение размера домашнего каталога

1. Удаление всех записей из crontab

* crontab -r - удаление всех задач

**Ответы на контрольные вопросы**

**1. Как получить информацию о состоянии памяти?**

Для проверки состояния оперативной памяти в системе используйте следующие команды:

* **Команда free**: Показывает общий объём памяти, использованную и доступную память, а также данные по swap:

free -h

* **Команда vmstat**: Выводит статистику по использованию ресурсов, включая информацию о памяти, CPU и swap:

vmstat

* **Файл /proc/meminfo**: Содержит подробную информацию о состоянии памяти, включая количество свободной и занятой памяти:

cat /proc/meminfo

**2. Как получить информацию о доступном дисковом пространстве?**

Для проверки информации о дисковом пространстве можно использовать:

* **Команда df**: Показывает размер и доступное место на каждом диске и разделе:

df -h

* **Команда du**: Выводит размер файлов и каталогов. Например, для получения размера домашней директории:

du -sh ~

**3. Как происходит работа с файлами заданий планировщика cron?**

Файлы cron управляют расписанием автоматического выполнения задач в системе. Чтобы управлять заданиями cron, используйте следующие команды:

* **Команда crontab -e**: Открывает файл crontab текущего пользователя в текстовом редакторе для редактирования. В этом файле можно добавить или удалить задания cron.
* **Синтаксис задания** в crontab состоит из пяти полей для указания времени выполнения задачи (минуты, часы, дни и т.д.) и команды, которую нужно запустить:

\* \* \* \* \* /path/to/command

Каждое \* означает определённый интервал (минуты, часы, дни и т.д.).

* **Команда crontab -l**: Показывает список всех активных заданий cron для текущего пользователя.

**4. Какая информация содержится в директории /proc?**

Директория /proc — это виртуальная файловая система, которая предоставляет информацию о состоянии системы, процессов и оборудования в режиме реального времени. Она содержит файлы и каталоги, отображающие текущие параметры ядра, памяти, процессора и устройств:

* **/proc/cpuinfo**: Содержит информацию о процессоре, включая модель, ядра и частоту.
* **/proc/meminfo**: Отображает состояние памяти (общий объём, использованная и свободная память, swap).
* **/proc/uptime**: Хранит информацию о времени работы системы с момента запуска.
* **/proc/[pid]**: Каталог для каждого активного процесса, где [pid] — это идентификатор процесса. Внутри содержатся файлы с параметрами и ресурсами, которые использует данный процесс.

Эти файлы в /proc можно просматривать с помощью команды cat, чтобы получить актуальную информацию о состоянии системы.