

## Управление процессами

### Цели работы:

- изучить основные команды и инструменты для управления процессами
- ознакомиться со средствами мониторинга производительности в Linux
- приобрести практический навык автоматизации запуска процессов

### Теоретические сведения

Процесс – это программа в стадии выполнения или, с точки зрения операционной системы, сущность, представляющая понятие активности/работы. Каждый процесс обладает уникальным идентификатором (PID) и контекстом выполнения, включающим в себя стек вызовов, снимок содержимого регистров процессора и прочие данные, необходимые для корректного выполнения в многозадачной системе. Операционная система осуществляет управление процессами, планирует их выполнение, выделяет ресурсы процессам, осуществляет контроль за их работой.

Процесс является потребителем ресурсов вычислительной системы, таких как процессорное время, память, дисковое пространство, устройства ввода-вывода, сетевые ресурсы и прочее. Мониторинг ресурсов позволяет выявлять узкие места и проблемы в работе системы, на основе чего принимаются решения по оптимизации настроек, улучшению производительности и предотвращению возможных сбоев и проблем.

### Утилиты для мониторинга процессов

Для получения более подробной информации по рассматриваемым утилитам здесь и далее рекомендуется использовать команду **man**.

Утилита **ps** позволяет просмотреть запущенные процессы.

Утилита **top** динамически отображает статистику системы, интерактивная. При запуске работает с окном терминала как динамический процесс. Позволяет получить как общую информацию по загрузке системы, так и детальнее изучить работающие процессы.

Утилита **atop** работает в интерактивном режиме, так же, как и утилита **top**. **Atop** позволяет получить загрузку по процессору, памяти, дискам и сети.

Команда **atopsar** — это встроенный анализатор логов **atop**, который позволит локализовать проблему путем получения лога за требуемый период времени. Утилита по умолчанию снимает показания каждые 10 минут, и сохраняет их в лог `/var/log/atop-YYYYMMDD`. Файл конфигураций находится `/etc/default/atop`.

Примеры команд для логирования:

нагрузки на RAM с 09:35:01 до 10:55:01 16 января

```
atopsar -r /var/log/atop/atop_20230116 -b 09:35:01 -e 10:55:01 -m
```

нагрузки по сетевым интерфейсам:

```
atopsar -r /var/log/atop/atop_20230116 -i -b 00:40:00 -e 01:30:00
```

нагрузки на диски:

```
atopsar -r /var/log/atop/atop_20230116 -d -b 01:10:00 -e 01:30:00
```

### Пакет sysstat

**Sysstat** – это набор инструментов мониторинга производительности для Linux, установка: ***sudo apt install sysstat***

В состав пакета входят утилиты: **mpstat**, **pidstat**, **vmstat**, **iostat**.

Утилита **mpstat** выводит отчет о использовании процессоров.

Утилита **pidstat** выполняет мониторинг выбранного процесса в реальном времени.

### Загрузка служб в Linux

**Systemd** позволяет планировать выполнение служб, процессов и скриптов и автоматизировать их запуск в фоновом режиме.

Утилита **systemctl** запускает сервисы, описанные в его конфигурации, состоящей из файлов (юнитов).

Системные сервисы расположены в директориях:

**/usr/lib/systemd/system/** – юниты из установленных пакетов RPM;

**/run/systemd/system/** – динамически создаваемые юниты;

**/etc/systemd/system/** – юниты, созданные системным администратором.

Юнит – текстовый файл, содержащий секции вида:

[Название секции в квадратных скобках]

имя\_переменной = значение

Для написания простейшего юнита нужно использовать секции:

[Unit] – для описания службы;

[Service] – необходимые для запуска команды;

[Install] – секция используется при установке службы (при enable/disable).

Пример простого юнита, запускающего (после запуска network.target) службу, выполняющую скрипт my\_service.sh:

[Unit]

Description=Sysstat service to collect data by my\_service.sh

After=network.target

[Service]

Type=simple

ExecStart=/etc/systemd/system/my\_service.sh

[Install]

WantedBy=multi-user.target

Для обновления конфигурации **systemctl** выполните команду **daemon-reload**: *sudo systemctl daemon-reload*.

Для добавления юнита в список служб, запускаемых при загрузке системы, используйте команду **enable**: *sudo systemctl enable my\_service.service*.

Для запуска службы используйте команду **start**: *sudo systemctl start my\_service.service*.

Убедиться в успешности запуска можно с помощью команды **status**, выполненной с указанием имени службы.

### Порядок выполнения работы

#### Задание 1. Мониторинг производительности системы

1. Выполните проверку системы при помощи команды **top**. Отсортируйте процессы по: объему используемой памяти; времени работы; идентификатору; проценту использования времени процессора.

2. Запустите новый процесс в системе, найдите его в выводе команды **top**. Измените приоритет запущенного процесса командой **nice** и убедитесь, что изменения отражены в выводе **top**.
3. С помощью команд **atop** и **atopsar** выполните сортировку процессов минимум по трем параметрам общей нагрузки.
4. При помощи команды **mpstat** и опции **P** выведите информацию по: определенному процессору, всем процессорам.
5. Выполните проверку системы при помощи команды **pidstat**. Выведите по четыре отчета статистики использования процессора для каждой активной задачи в системе с интервалом в три секунды.

В отчете приведите ответ в виде снимков экрана по каждому пункту задания.

### **Задание 2. Создание скрипта для получения статистики**

Напишите скрипт, осуществляющий сбор статистики использования процессоров и статистики работы процессов средствами **mpstat** и **pidstat**, выводящий данные в текстовый файл с именем `уууу-мм-дд_h:m:s_syslog`.

В отчет включите текст скрипта, содержимое файла лога и вывод команды **ls -l** для директории сохранения логов скрипта.

### **Задание 3. Создание автозапускаемой службы**

С помощью утилиты **systemd** зарегистрируйте свою службу по сбору статистики, запускающую скрипт из задания 2 при старте системы. Запустите службу и проверьте ее работу.

В отчете приведите снимок экрана с выводом команды **ls -l** для директории сохранения логов скрипта, статистику выполнения службы и текст файла `.service`.

### **Задание 4. Создание таймера**

Создайте таймер для запуска созданной службы каждую минуту, запустите таймер и проверьте его статус. После нескольких (5-7) минут работы, остановите таймер. Отрадите в отчете результат работы таймера (снимок экрана со статистикой работы таймера и вывод команды **ls -l** для директории сохранения логов скрипта).