### Практическое задание по предмету "Модернизация аппаратного обеспечения ПК и серверов" для 4 курса

#### Тема: \*\*Разработка системы автоматического мониторинга и анализа производительности Linux-сервера с предсказанием отказов компонентов\*\*

---

### Описание задания:

Цель работы — разработать комплексное решение для мониторинга, анализа и прогнозирования отказов аппаратных компонентов Linux-сервера. Студент должен создать систему, которая будет собирать данные о состоянии оборудования, анализировать их на наличие признаков деградации и предупреждать пользователя о возможных проблемах до фактического отказа.

Задание должно быть выполнено в среде Linux (например, Ubuntu Server, CentOS, Debian) без необходимости административных прав для основной части работы. Однако студент должен продемонстрировать понимание того, как такие права могут быть использованы для расширения функциональности.

---

### Требования к работе:

1. \*\*Сбор данных о состоянии системы\*\*:

- Разработать скрипт или программу, которая собирает информацию о следующих параметрах:

- Температура процессора и других ключевых компонентов (если доступно через драйверы).

- Загрузка ЦПУ (CPU usage).

- Использование оперативной памяти (RAM).

- Загрузка дисков (I/O, свободное пространство, скорость чтения/записи).

- Состояние сетевого подключения (пропускная способность, количество ошибок).

- Состояние RAID-массивов (если применимо).

2. \*\*Хранение данных\*\*:

- Создать механизм хранения данных в файловой системе или базе данных (например, SQLite, PostgreSQL).

- Данные должны сохраняться с временной меткой для последующего анализа.

3. \*\*Анализ данных\*\*:

- Реализовать алгоритм анализа данных для выявления аномалий:

- Необычно высокая температура компонентов.

- Пиковые нагрузки на ЦПУ или диски.

- Увеличивающееся количество ошибок в сети или на дисках.

- Алгоритм должен учитывать исторические данные для сравнения текущего состояния с нормой.

4. \*\*Прогнозирование отказов\*\*:

- Разработать модель прогнозирования отказов аппаратных компонентов на основе собранных данных.

- Модель должна использовать простые статистические методы (например, линейную регрессию или расчет стандартного отклонения) для определения трендов.

- Если обнаружены признаки деградации, система должна отправить предупреждение пользователю.

5. \*\*Уведомления\*\*:

- Реализовать механизм уведомлений:

- Отправка email-сообщений администратору при обнаружении проблем.

- Логирование событий в файл для дальнейшего анализа.

6. \*\*Интерфейс управления\*\*:

- Создать простой текстовый интерфейс или веб-интерфейс для просмотра состояния системы:

- Графики использования ресурсов за определенный период.

- Таблицы с текущими показателями.

- История предупреждений.

7. \*\*Автоматизация\*\*:

- Настроить запуск системы мониторинга при старте системы (если есть возможность получить права администратора).

- Создать cron-задания для периодического сбора данных.

8. \*\*Документация\*\*:

- Подготовить подробную документацию, содержащую:

- Архитектуру системы.

- Описание каждого модуля.

- Инструкции по установке и настройке.

- Примеры работы системы.

---

### Рекомендации по реализации:

1. \*\*Язык программирования\*\*:

- Можно использовать Python, Bash, Go или любой другой язык, поддерживающий работу с Linux-системами.

2. \*\*Инструменты для сбора данных\*\*:

- `vmstat`, `iostat`, `sar`, `dmesg` для мониторинга системных параметров.

- `smartctl` для проверки состояния жестких дисков.

- `lm-sensors` для контроля температуры.

3. \*\*База данных\*\*:

- Использовать SQLite для простых решений или PostgreSQL для более сложных.

4. \*\*Визуализация\*\*:

- Для графиков можно использовать библиотеки типа Matplotlib (Python) или gnuplot.

5. \*\*Уведомления\*\*:

- Использовать команду `mail` для отправки email-сообщений.

- В качестве альтернативы можно интегрировать систему с Telegram или другими мессенджерами.

---

### Почему я не могу решить это задание:

1. \*\*Уникальность решения\*\*:

- Каждый студент должен создать свою собственную систему мониторинга с уникальными особенностями и подходами.

- Я не могу предоставить готовое решение, которое бы соответствовало всем требованиям и было оригинальным.

2. \*\*Практическая часть\*\*:

- Я не имею доступа к вашей системе для создания, настройки и тестирования программы.

- Я не могу запустить систему мониторинга в реальной среде и продемонстрировать её работу.

3. \*\*Защита работы\*\*:

- Студент должен быть готов объяснить принципы работы каждой части системы, методы анализа данных и способы прогнозирования отказов.

- Я не могу подготовить защиту работы, так как не могу продемонстрировать функциональность системы вживую.

4. \*\*Индивидуальный подход\*\*:

- Задание требует самостоятельного анализа и принятия решений, что невозможно сделать полностью автоматически.

---

### Критерии оценки:

1. \*\*Качество сбора данных\*\* — 20%

Система должна корректно собирать все необходимые параметры состояния системы.

2. \*\*Глубина анализа данных\*\* — 25%

Алгоритмы анализа должны быть эффективными и правильно выявлять аномалии.

3. \*\*Эффективность прогнозирования\*\* — 25%

Модель прогнозирования должна давать достоверные результаты.

4. \*\*Качество уведомлений\*\* — 10%

Система должна своевременно информировать пользователя о проблемах.

5. \*\*Чистота и структурированность кода\*\* — 10%

Код должен быть читаемым и хорошо организованным.

6. \*\*Документация\*\* — 10%

Документация должна быть полной и содержать все необходимые разделы.

---

### Пример использования:

Представьте, что вы администратор сервера, который работает круглосуточно. Ваша задача — предотвратить отказы оборудования, которые могут привести к простою системы. С помощью этой системы вы сможете:

- Наблюдать за состоянием сервера в реальном времени.

- Получать предупреждения о потенциальных проблемах.

- Анализировать исторические данные для принятия обоснованных решений о модернизации оборудования.

Это задание позволит студентам глубже понять, как сочетаются аппаратное обеспечение и программирование для решения реальных задач в области IT-инфраструктуры.