

# Программист Си

[otus.ru](https://otus.ru)

**Меня хорошо видно  
&& слышно?**



## Защита проекта № 2

Тема: *Программа по анализу 29 параметров нагрузки ПК с записью результатов в БД PostgreSQL.*



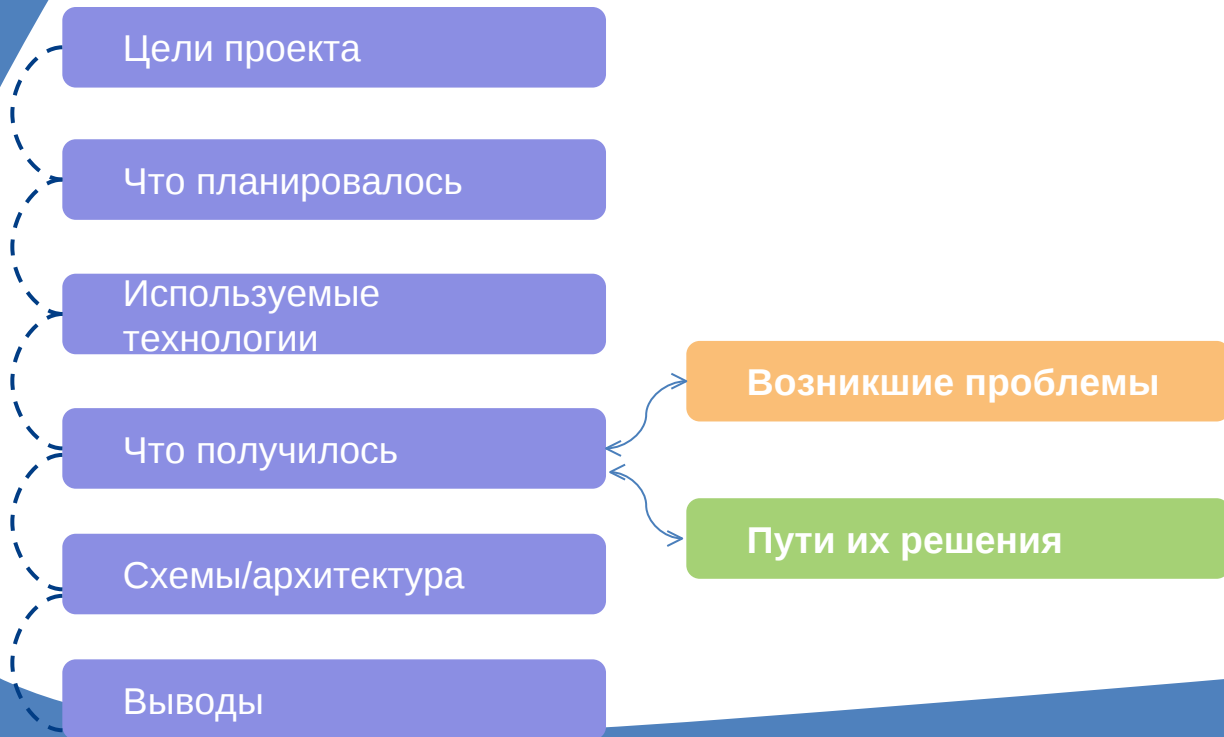
Сартаков Алексей

[pismo@ohido.ru](mailto:pismo@ohido.ru)

[https://t.me/Sartakov\\_Aleksey](https://t.me/Sartakov_Aleksey)

[Подробное описание проекта в папке «описание\\_инструкции»](#)

# План защиты



# Цели проекта

Какие цели вы поставили и какие задачи решили своим проектом

1. Учёт нагрузки сервера по температуре процессоров.
2. Учет нагрузки на жёсткий диск (SSD).
3. Принудительное снижение скорости работы программы из первого проекта, чтобы нагрев (ноутбука) не выходил за допустимые границы.
4. Запись всех данных о нагрузке сервера в БД PostgreSQL.
5. Автоматический запуск программы в среде Debian, после перезагрузки ПК (в течении 30 секунд после перезагрузки).
6. Вывод данных о нагрузках сервера, по всем 29 параметрам в виде отдельных графиков, с разбивкой по дням (выбор по датам).

# Что планировалось

Что было в начале, что знали до курса, сколько времени заняло выполнение проекта

1. Первоначально хотел знать температуру центрального процессора и его нагрузку. Потом возникла идея записывать все это в БД PostgreSQL, т.к. основная работа у меня связана с БД.
2. Затем, в процессе реализации проекта, при анализе каталога «/proc/» в Linux узнал о других учитываемых параметрах системы (нагрузках). Те, которые показались мне интересными, так же добавил в этот проект.
3. Проект выполнялся около 20-25 рабочих дней (с середины августа 2023) по 6-7 часов.
4. До проекта очень хорошо знал БД PostgreSQL (SQL, настройки, оптимизация, роли, функции pl/pgsql, окна, выборки). Так же знаю Python, HTML и SVG разметку, сервер-сайта Apache2.
5. На Си это была первая серьезная программа. До этого только тестовые задачки из книг.

# Используемые технологии

1.	PostgreSQL, SQL, pl/pgsql
2.	Python
3.	HTML, SVG и сервер-сайта Apache2
4.	Cи

Какие технологии использовались и какое  
у вас мнение о новых технологиях

# Что получилось

1. Ссылка на репозиторий с исходным кодом, копией БД, кодом на Python для сайта. [Перейти.](#)
2. Подробное описание проекта. [Перейти.](#)





# Возникшие проблемы

1. Именно в этом проекте особых проблем и не было. Просто новый материал шел с обычными заковычками
2. Сама логика структур и запросов очень простая. В Си не было никаких сложных структур, никакой многопоточности.
3. Разве что имя БД должно быть на латинице, а сами схемы, таблицы, столбцы могут уже на русском языке. И имя «демон файла» должно быть только латинскими буквами.
4. С БД были проблема – если указать что соединение идет как localhost, то все данные дублировались в запросе два раза.

Какие технологии использовались и какое у вас мнение о новых технологиях

# Пути их решения

1. Переименовать имя БД и имя демон файла латинскими буквами. Но эти нюансы были известны еще за пол года до текущего проекта.
2. Запрос более 1 раза в секунду приводит к блокировке этого файла. То есть если считывать данные с частотой более 3-х раз в секунду, через некоторое время данные о нагрузке процессора перестанут поступать. Это классификация в Linux и там прямо указано про интервал запроса в 1 секунду.
3. При подсоединении к БД как localhost просто вырезался ключ с параметром «localhost» и все проблемы с дублированием строк исчезли.

Как были устранены возникшие проблемы

# Схемы (архитектура, БД)

DBeeper 23.3.3 - нагрузка\_системы

Файл Редактирование Навигация Поиск Редактор SQL База данных Окна Справка

SQL Auto test\_si\_2023 public@test\_si\_2023

Базы данных x Проекты

Введите часть имени объекта дл...

test\_si\_2023 - localhost:5434

- Базы данных
  - test\_si\_2023
    - Схемы
      - \_репликация
      - db\_log
        - Таблицы
          - нагрузка\_системы 1,6M
      - Внешние таблицы
      - Представления
      - Мат. представления
      - Индексы
      - Функции
      - Последовательности
      - Типы данных
      - Агрегатные функции
    - public
    - site\_html
    - temp
    - temp\_big\_moex\_mmvb
    - акции\_рф
    - справочники
    - торги\_реальн\_время
    - Событийные триггеры
    - Расширения
    - Хранилище
    - Системные объекты
    - Роли
    - Администрирование
    - Системные объекты
  - еда - localhost:5433
  - mmvb\_робот - localhost:5432

Название: нагрузка\_системы ID объекта: 16644

Табличное пространство: pg\_default Владелец: postgres

Дополнительные опции:

Секционирование по:

Комментарий: учитывает нагрузку операционной системы. Одно точка равна 5 минутному интервалу. Каждую 1 минуту происходит обновление данных в БД.

Колонки	Название	#	Тип данных	Not Null	По	Комментарий
Ограничения	123cpu_min	1	numeric(4, 1)	[ ]		нагрузка процессора в процентах от 0.0 до 100.0 .
Внешние ключи	123cpu_max	2	numeric(4, 1)	[ ]		нагрузка процессора в процентах от 0.0 до 100.0
Индексы	123cpu_midle	3	numeric(4, 1)	[ ]		Нагрузка процессора в процентах .Здесь учитывается средне арифметическое значение с инте
Зависимости	123ram_used_min	4	numeric(4, 1)	[ ]		Использование оперативной памяти в процентах от 0.0 до 100.0
Ссылки	123ram_used_max	5	numeric(4, 1)	[ ]		использование оперативной памяти в процентах от 0.0 до 100.0
Секции таблиц	123ram_used_midle	6	numeric(4, 1)	[ ]		Использование ЗАНЯТОЙ/ИСПОЛЬЗУЕМОЙ оперативной памяти в процентах. \$Эту память никак
Триггеры	123swap_read_block	7	int4	[ ]		ЧТЕНИЕ SWOP файлов с диска. Считается как сумма всех изменений, относительно предыдуще
Правила	123swap_write_block	8	int4	[ ]		ЗАПИСЬ SWOP файлов с диска. Считается как сумма всех изменений, относительно предыдуще
Policies	123cpu_t_min	9	numeric(4, 1)	[ ]		Минимальная температура процессора. \$в процентах от 0.0 до 100.0
Statistics	123cpu_t_max	10	numeric(4, 1)	[ ]		Максимальная температура процессора. \$в процентах от 0.0 до 100.0
Права доступа	123cpu_t_midle	11	numeric(4, 1)	[ ]		Средняя температура процессора в градусах цельсия.\$Здесь учитывается средне арифметиче
DDL	123ssd_read_block	12	int4	[ ]		Всего считали блоков с диска за период времени между временными точками
Virtual	123ssd_write_block	13	int4	[ ]	0	Всего записали блоков на ssd диск за период времени между временными точками
	время	14	timestamp	[v]		время начала периода. Строго кратно 5 минутам.
	123ram_full_free	15	numeric(4, 1)	[ ]		оперативная память, которая ПОЛНОСТЬЮ свободна. Возможно, что утром придётся делать очи
	123ram_доступно_max	16	numeric(4, 1)	[ ]		Это размер оперативной памяти в процентах, которую можно использовать, но часть её занят
	123ram_доступно_min	17	numeric(4, 1)	[ ]		Это размер оперативной памяти в процентах, которую можно использовать, но часть её занят
	123ram_доступно_midle	18	numeric(4, 1)	[ ]		Это размер оперативной памяти в процентах, которую можно использовать, но часть её занят
	123ram_buf_min	19	numeric(4, 1)	[ ]		Это оперативная память, занятая буферами. Размер указан в % от размера основной памяти
	123ram_buf_max	20	numeric(4, 1)	[ ]		Это оперативная память, занятая буферами. Размер указан в % от размера основной памяти
	123ram_buf_midle	21	numeric(4, 1)	[ ]		Это оперативная память, занятая буферами. Размер указан в % от размера основной памяти
	123ram_cach_min	22	numeric(4, 1)	[ ]		Это оперативная память, занятая кэш памятью. \$В основном этот кэш выделен под работу Post
	123ram_cach_max	23	numeric(4, 1)	[ ]		Это оперативная память, занятая кэш памятью. \$В основном этот кэш выделен под работу Post
	123ram_cach_midle	24	numeric(4, 1)	[ ]		Это оперативная память, занятая кэш памятью. \$В основном этот кэш выделен под работу Post
	123ram_ssd_очередь	25	int4	[ ]		Это очередь в кб, которая пока ОЖИДАЕТ записи на жесткий диск. Указывается МАКСИМАЛЬНО
	123ram_ssd_запись	26	int4	[ ]		Это блоки в кб, котоые прямо сейчас ЗАПИСЫВАЮТСЯ на жесткий диск. Указывается МАКСИМА
	123ram_user_min	27	numeric(4, 1)	[ ]		В процентах. Это память, которая занята всеми пользовательскими приложениями, программа
	123ram_user_max	28	numeric(4, 1)	[ ]		В процентах. Это память, которая занята всеми пользовательскими приложениями, программа
	123ram_user_midle	29	numeric(4, 1)	[ ]		В процентах. Это память, которая занята всеми пользовательскими приложениями, программа

29 элементов

Сохранить ... Вернуть Обновить

MSK ru RU

# Выводы и планы по развитию

1. Годовой объем данных в БД около 26 МБ данных, что для современных БД крохи.
2. В этом проекте все сделано что запланировано. Планов дальнейших нет.
3. Проект, программа успешно функционирует с ноября 2023 и за 3 месяца никаких проблем, сбоев не было. После каждой перезагрузки успешно стартует и пробелов/разрывов в данных нет. Все данные пишутся в БД успешно.
4. Благодаря данной программе сократил объем выделенной памяти под PostgreSQL с 9,7 Гб до 4,8 Гб. Полет нормальный, SWAP файл не используется. Но как показали дальнейшие работы, это не совсем есть показатель и анализ БД PostgreSQL делать нужно штатными средствами статистики (отдельное приложение stat) и разбор «планов» по функциям. Но это отдельная тема с кучей подводных камней.

Запланируйте пару минут на рефлекссию в конце защиты проекта и расскажите о планах по развитию

**Спасибо за внимание!**