1. Изучение архитектуры процессора можно выполнить наиболее эффективно, если написать эмулятор

процессора x86 архитектуры и программировать для него в машинных кодах, затем пребывать на реальном

процессоре тоже самое. Линус Торвальдс пишет, что программирование в машинных кодах выводит на новый уровень понимания работы программ, исчезают все барьеры

2. Конструирование простейшей вычислительной машины (любой природы: механической, электронной) + ассемблера для нее, языка высокого уровня, системы ввода (клавиши механические или электронные) и вывода (монитор или устройство затенения света для отображения символов)

3. Разработать какой-либо жесткий, универсальный ассемблер, с возможностью писать, как с макросами ЯВУ так и на машинном коде. Необходимо от и до знать архитектуру процессора.

4. Можно писать программы под чистое железо, выполняемые сразу после загрузки компьютера (без ОС) таким образом можно исключить выполнение постороннего кода. Что может дать такой подход, какие программы и методы, функционально?

А) Диагностика железа

Б) Интересно если в цикле выполнять одну и ту же инструкцию многократно, а затем вытащить процессор и посмотреть на него в телевизор, то можно увидеть локальные точки перегрева

В) Если выполнять различные инструкции процессора многократно, то можно вычислить точное время выполнения каждой инструкции данной архитектуры команд процессора

5. Можно написать на С# или C++ под .NET систему имитационного моделирования. За одно подучить эти языки и платформы, а также возможно её кто-то купит

~~6. Сделать свой сайт-блог~~

7. Интернет-магазин

~~8. Сайт для больницы~~

9. Операционная система для квантового компьютера. Так как его у меня нет, то нужно запустить эмулятор квантового компьютера (такое вообще возможно?)

10. ИНТЕРЕСНАЯ И ГЛУБОКАЯ МЫСЛЬ: программы разрабатываются путем логического разбиения задачи на мелкие подзадачи и алгоритмической реализации этих подзадач на конкретном машинном языке (при помощи ассемблера или ЯВУ). ОДНАКО, существуют ли другие способы? Ответ на этот вопрос зависит от двух факторов:

А) Архитектура вычислительного устройства (в том числе ПРИРОДА? (квантовые компьютеры))

Б) Парадигмы программирования (теперь то я глубже понял мощность этого понятия, после изучения ассемблера)

Любой язык программирования, любая парадигма, должна обеспечивать выполнение двух критериев: максимального быстродействия выполнения программы и минимального времени написания (разработки) программы. Минимальное время написания зависит от МОЩНОСТИ ЯЗЫКА. Язык - это средство коммуникации, коммуникация - это ПЕРЕДАЧА ИНФОРМАЦИИ, в данном случае - ПЕРЕДАЧА ЗАМЫСЛА ПРОГРАММИСТА МАШИНЕ. Почему я не могу писать программы на человеческом языке? Например, "Разработать операционную систему для процессоров Intel Core с вытесняющей многозадачностью и .т.д."? Дело в том, что все понятия в этом предложении требуют уточнения. Уточнение - это определение понятия через другие понятия, которые в свою очередь также уточняются пока в этом процессе уточнений не встретятся АТОМАРНЫЕ ПОНЯТИЯ МАШИННОЙ АРХИТЕКТУРЫ. Например, "Вытесняющая многозадачность -> Перебор всех задач в памяти и исполнение на выбор какой-либо из них -> Для конкретной переменной-

идентификатора задачи узнать сколько секунд прошло с момента выполнения, если кол-во секунд превышает 10 ms, то загрузить в CS:IP адрес следующей задачи -> и т.д. до поместить в память значение FA5590 F9CC23 и т.д.". Это конечно приблизительно. В этой связи возникает вопрос: какие вообще могут быть парадигмы? Короче подумать над вопросом: "КАКИЕ ЕСТЬ СПОСОБЫ ПЕРЕДАЧИ ЗАМЫСЛА ПРОГРАММИСТА МАШИНЕ?".

11. Написать книгу "10 нетривиальных проектов в Delphi". Какие проекты?

1. Система имитационного моделирования

2. HEX-редактор

3. Простой ассемблер (с упрощенным набором команд)

4. Архиватор

5. Эмулятор сетей Петри

6. Анализатор локальной сети

7. Анализатор HDD

8. Сбор информации о системе

9. Шахматы

10. Отладчик (Дизассемблер или еще что-то в этом духе)

12. Написание книги "Прикладная математика и программирование (на С++ или С#?)" - по книге А. Кафмана

!!! 13. Создать сайт-сервис "Ментор" задания вопросов в живую экспертам за деньги по видеосвязи. Все разбито по темам, имеются рейтинги

14. Какая-либо мысль которая крутится в голове, существует как информация во Вселенной даже до того, как она будет запечатлена вне этого человека. Любая информация - это свойство объекта, позволяющее

познавать этот объект на основе его различий с другими объектами. Следовательно, мысль в голове имеет информацию потому, что она отличается от других мыслей (или системы категорий?).

15. Завести тетради: 1. Идеи 2. Что нового узнал (+ объяснение ребенку по Фейману) 3. Какие книги купить и что прочесть и где найти 4. По тетради по проектам (разработка ОС, написание книги, изучение Ардуино и прочее)

16. Подготовить лекции для курса по программированию для школьников (можно заработать)

17. По-видимому все языки программирования линейны в том смысле, что не позволяют убыстрять процесс программирования при написании сложной программы. Мы конечно можем в очередной раз воспользоваться уже написанным кодом и конструировать на его основе более мощные программы, однако возникает вопрос. Есть ли более эффективная парадигма, механизм, которая позволяет экспоненциальной увеличивать быстроту написания программ от кол-ва уже введенного кода. Такое возможно только в том случае, если уже написанный код (который в естественной линейной парадигме не обязан быть таким же как требуется в дальнейшем, то есть он делает не то что нужно) может быть гарантированно использоваться в дальнейшем. То есть чем больше ты пишешь, тем гарантированное этот код будет использоваться. Значит дело в глубокой внутренней концепции парадигмы. Есть ли вообще в природе такие примеры

когда из мелких кусков строится нечто большее из чего, в свою очередь, строится еще большее? В некотором смысле такое реализовано и в обычных библиотеках языков (использование труда других программистов). Но здесь я подразумеваю нечто большее, а именно не линейное использование, а экспоненциальное. Тогда такое возможно только в том случае, если этот написанный блок обязательно в дальнейшем будет использоваться уже как новый кирпич!!! Вот в чем дело. Экспоненциальный характер функции убыстрения написания программы возможен только в том случае, когда в процессе разработки программист ОБЯЗАТЕЛЬНО шаг за шагом переходит к использованию в качестве инструкции все более крупных блоков (более абстрактных, общих).

18. Все книги из энциклопедического словаря юного математика

19. Продуктивность зависит от сведения к минимуму простоя, вызванного незнанием того, что делать. Нужно при учебе составлять руководства и НАПРАВЛЯЮЩИИЕ конспекты.

20. Можно разработать систему агрегатор информации, которая 1) Хранит всю нужную, отмеченную информацию 2) Упрощает доступ к ней 3) Визуально более восприимчива, чем, например, поиск в интернете. Сила такой системы и основная суть как раз в высоком восприятии информации и в продвижении в процессе учебы, исследовании и разработке. Как она может выглядеть? Что-то типа предметного указателя с ссылками на ресурс + номер страницы и абзаца? Помимо ссылок должна присутствовать визуальная информация. Также должно быть схема. Итак, такая система стоит на трех китах: ссылки. визуальная информация, схема. Продуманная схема - основной аспект. Должно быть два уровня схем: первый уровень - цели, второй - средства, согласованные с целями. Таким образом не будет отклонения в сторону. Это есть оптимальный вариант для соединения точек Джобса, без простоя и пустой траты времени

21. Сайт или приложение для компьютерных мастеров, которое определяет ошибку (не только по номеру, в этом то и дело) и показывает, как её исправить. Более продвинутая система, чем форум. Например, на основе скрина определят параметры и прочее, ищет в базе ошибку и показывает пути её устранения.

!!!22. Возможно ли спроектировать и разработать систему и язык имитационного (или какого-либо другого) моделирования, которая автоматически определяет оптимальную структуру системы. Сначала в систему вводится модель "как-есть", а затем система осуществляет прогон модели и определяет, на основе анализа, оптимальную структуру модели. С теоретической точки зрения здесь есть новшество (?) состоящее в взаимно-однозначном соответствии (отражении) модели "как-сеть" и модели "как должно быть". Может быть здесь кроется какая-либо высшая неизвестная алгебра?

23. Разработать сайт "агрегатор товаров Сухой Лог Богданович Камышлов Асбест", к которому подключатся магазины этих городов для создания единой актуальной товарной базы. Если создать крупную агломерацию магазинов не получится, то можно сделать из него своеобразный интернет-магазин, не меня названия. По анализу поисковых запросов этого магазина можно определить, на основе проблемы запасов, - оптимальную товарную базу

24. Роман о гипотетической стране, в которой все заточено на развитие личности

25. Научно-популярная википедия. Акцент на то, чтобы было всем понятно. "НаучПоп" (или на блоге техноквант создать такой раздел - иерархия понятий)

26. Язык имитационного моделирования "Колмогоров"

27. Для продуктивной работы необходимо наличие цели перед глазами. Что ты в конечном итоге получишь, чего достигнешь?

28. Сайт объективной статистики по запросу от пользователя его мнения о тех или иных явлениях, вещах, событиях. На сайте должна быть реализована мощнейшая система защиты от накруток. Если запустить опрос в одно и тоже время, при разных условиях (например, один прорекламировать, а второй нет или один опрос в одной среде, а другой в другой), то можно делать объективные выводы сравнивая результаты одного и того же запроса, сделанного в разных условиях. Например, уровень манипуляции, заинтересованность тех или иных групп людей и прочее.

29. Разработка языка имитационного моделирования "Колмогоров", который помимо имитационной логики позволяет задавать логику отображения в виде 3d графики

30. Любой компьютер с выполняющейся в данный момент программой является дискретным, информационным аналогом реальности (ДИАР - мой термин) с многократно убыстрённым течением времени. Программа в этом случае - модель (шаблон) реальности. Как в реальности имеются некоторые данные (и их отношения) и действия над ними, так и в ДИАР. Например, в реальности имеется человек, который работает по установленному графику, ставке заработной платы - это все данные, действия это - нормировщик берет ФИО и таб номер этого рабочего и проставляет то, как он работал в текущем месяце и рассчитывает за это месяц заработную плату по установленной ставке. Этот же случай в ДИАР - соотнести определённому id (рабочего) последовательность переменных в соответствии с полями из справочной таблицы и вычислить сумму произведения (ставка \* количество часов за смену).

!!!! 31. в пункте 17 и 10 я писал о парадигме программирования, которая позволяет писать программы экспоненциально. Однако, можно еще убыстрить процесс написания программ с помощью НОВОГО СПОСОБА передачи информации, не вводом символов, а графикой, например, фотками

32. Нужно разработать сайт грузоперевозок где будет личный кабинет заказчиков и владельцев, можно получается делать онлайн заказы, просматривать на карте маршруты, историю заказов и тому подобное.

33. Создать web ресурс генерируемый программно, который содержит перечень (классификацию, сильно связанную информацию) типа википедии, с указанием фактов с ссылками на первоисточники. Информация может быть двух типов: ложь и истина.

34. Эффект Джанибекова для тел, скатывающихся с наклонной плоскости? Будет иметь место или нет? Если в шаре сделать неравномерное распределение масс, такое что что будет три оси инерции, то эффект Джанибекова будет иметь место или нет?

35. Система программирования в которой визуализирован прогресс процесса разработки. Например, по мере написания и интеграции модулей они окрашиваются в один цвет

36. Система и среда имитационного моделирования и императивного программирования (или другая парадигма) с возможностью 3D-визуализации процесса моделирования (продвижения транзакта по модели).

Для этого нужно реализовать виртуальную машину, которая будет выполнять вставки императивного кода (на высокоуровневом языке). Возможно также, чтобы императивный код обладал полнотой по API операционной системы. Тогда, можно делать мощные императивные вставки, хоть реальную пересылку данных по сети с визуализацией передачи прикладной информацией, или выключение и отключение оборудования и прочее. Также возможно наполнить функционалом как в Powersim (системно динамическое моделирование).

37. Интернет-сервис (веб-сайт) поиска, анализа и систематизации научно-технической информации.

38. Написать статью "к теории инерционности общественных процессов" (политических, экономических). Подвести научное обоснование под экономику и политику по типу законов Ньютона. Выделить элементарное взаимодействие, закон.

На основе этой теории показать сильную инерционность долларовой денежной системы, обосновть почему она так жизнеспособна и какие интересы обслуживает ЦБ России.