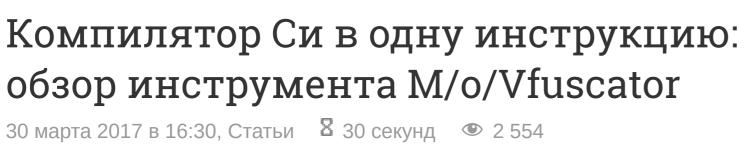
(i) X

Q



Новости Задачи с ІТ-собеседований Эксперты отвечают читателям Вакансии



|::\ \ /::\ \ /\ \ /:/\_/\_\:\ \ /:/ \ /::\ \ \_\_\_ /::\ \ \ ... \_|:|\:\ \ \:\ \\:\ \/:/ / \:\ \ /::\ \ / \_://::\ \/ // \:\ \/:/ / /::::|\_\:\\_\_\/ \:\\_\_\ \:\\_\_\:/ / \ \:\\_\_\:/\:\\_\_\/ /\\_\_/ \:\\_\_\:/\_\_/ \:\~~\ \/\_\_/\ /:/ / |:| |/ /\ \/:/ ///:/ // \/\_/ \\ \/:/ /::::/ / \:\ \ \:\ /:/ /\|:| |\_\_/\:\ /:/ // /:/ //\_/:/\:\ \\ /:/ //~~/~~~~ \:\ \ \:\/:/ /\\_|:|\_| \ \:\/:/ //:/ /:\/:/ /:\ \\\_\:\ \/:/ /:\~~\ \:\\_\ \::/ /\:::/\_/:\\_\ \::/ / /:/ / \::/ / \:\\_\ \:\\_\ \/\_/ \/\_/ ~~~~ \/\_/ \/\_/ \/\_/ \/\_/ \/\_/ M/o/Vfuscator компилирует программы в инструкции mov, и только

процессора х86, но адаптивен и легко настраивается под другие языки и архитектуры. Демонстрация Компиляция функции, вычисляющей простые числа, с помощью

в них. Инструмент ориентирован на язык Си и архитектуру

## M/o/Vfuscator в сравнении с GCC:

Язык Ассемблера GCC M/o/Vfuscator

<is prime>:

DWORD PTR [ebp+0x8],0x1 8048490 <is\_prime+0x13> DWORD PTR [ebp+0x8],0x2 DWORD PTR [ebp-0x4],0x2 idiv DWORD PTR [ebp-θx4] eax,θxθ 80484a6 <is prime+0x29: Граф потока управления

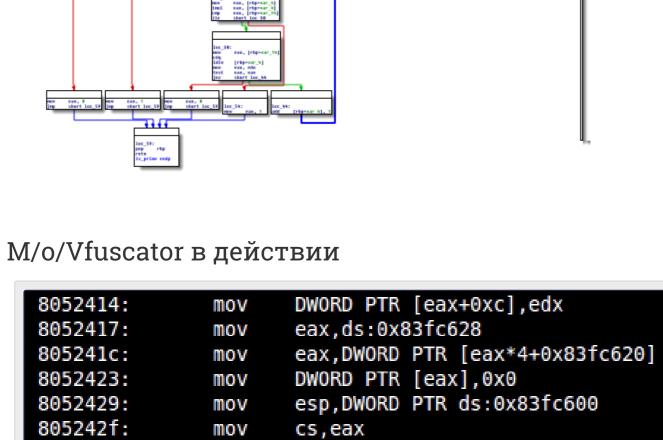


\$ ./prime

primes, 1-1000:

GCC



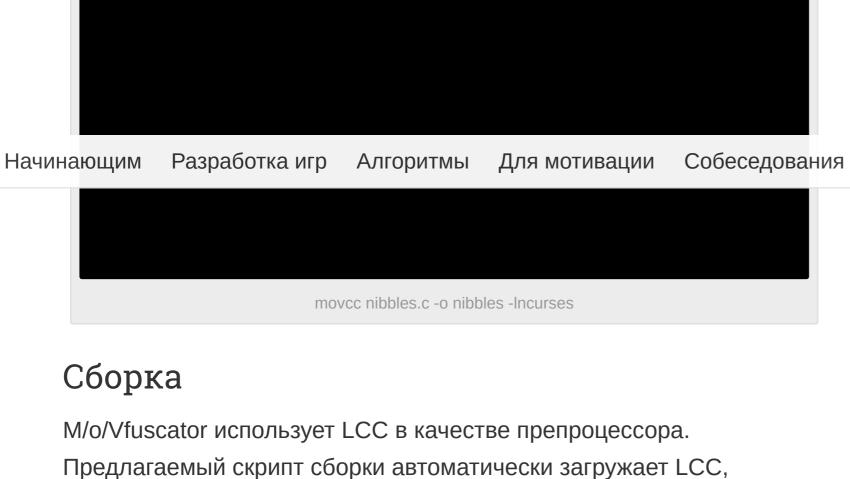


349 353 359 367 373 379 383 389 397 401 409 419 421 431 433 43 9 443 449 457 461 463 467 479 487 491 499 503 509 521 523 541 547 557 563 569 571 577 587 593 599 601 607 613 617 619 631 64 1 643 647 653 659 661 673 677 683 691 701 709 719 727 733 739 743 751 757 761 769 773 787 797 809 811 821 823 827 829 839 85 3 857 859 863 877 881 883 887 907 911 919 929 937 941 947 953 967 971 977 983 991 997 movcc prime.c -o prime Пример поинтереснее \$ movcc nibbles.c -o nibbles -lnc

2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79

83 89 97 101 103 107 109 113 127 131 137 139 149 151 157 163 167 173 179 181 191 193 197 199 211 223 227 229 233 239 241 25

1 257 263 269 271 277 281 283 293 307 311 313 317 331 337 347



настраивает его конфигурацию для MOV и собирает M/o/Vfuscator.

Если вы проводите сборку на 64-битной системе, убедитесь, что

у вас есть доступ к 32-битной стандартной библиотеке (например,

apt-get install libc6-dev-i386 ИЛИ yum install glibc-devel.i686 ).

## git clone https://github.com/xoreaxeax/movfuscator cd movfuscator ./build.sh

следующего скрипта:

./check.sh

sudo ./install.sh Тестирование

Проверьте, что сборка произошла корректно, с помощью

Использование Компилируйте программы как обычно.

Всю остальную необходимую информацию можно найти в репозитории проекта на GitHub.

@tproger\_official

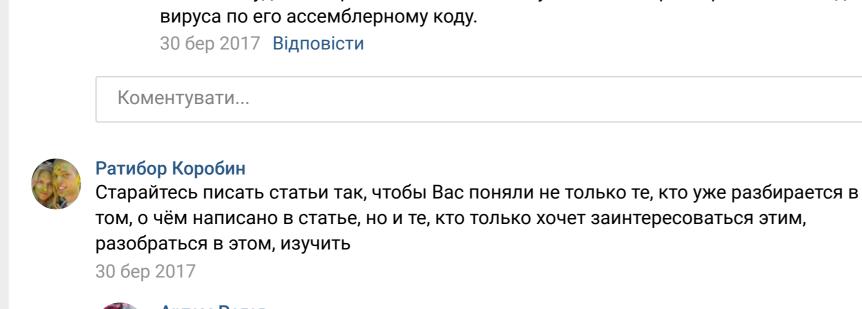
movcc example.c -o example

Язык ассемблера, Язык Си

**f** Подписаться

Типичный программист W

Open Source, Инструменты, Низкоуровневое программирование,



если на сайте будут статьи только для новичков, то это будет очень скучный сайт. Касательно этой статьи - учи ассемблер 30 бер 2017 Відповісти Евгений Халуев відповів Артему Артем, это из разряда а почему бы и нет. Корни растут из процессоров в которых

выполнить могут абсолютно любую программу.

только одна инструкция. Такие процессоры очень простые в реализации, а

Курсы для тех, кто уже разбирается в программировании. Чтобы записаться, нужно пройти тестирование и доказать свой уровень: JavaScript Java Тимлид DevOps Python Android

Пройти

66 Сметана — это драйвер для борща. "

Анонимный поиск

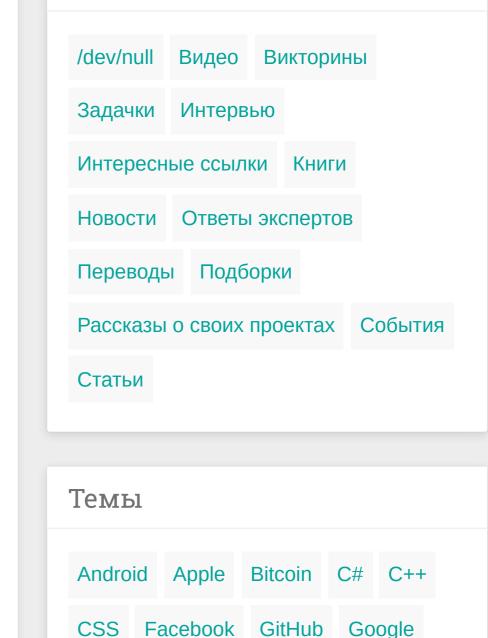
работы для

программистов

Найти интересные предложения без

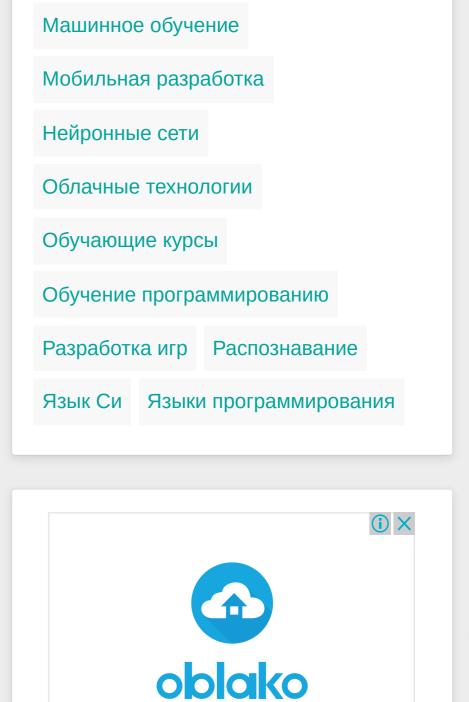
проблем на текущем месте работы

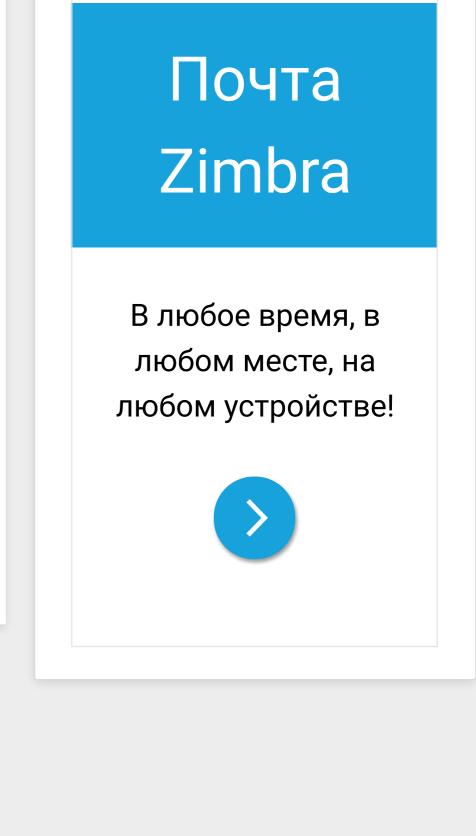
Рубрики

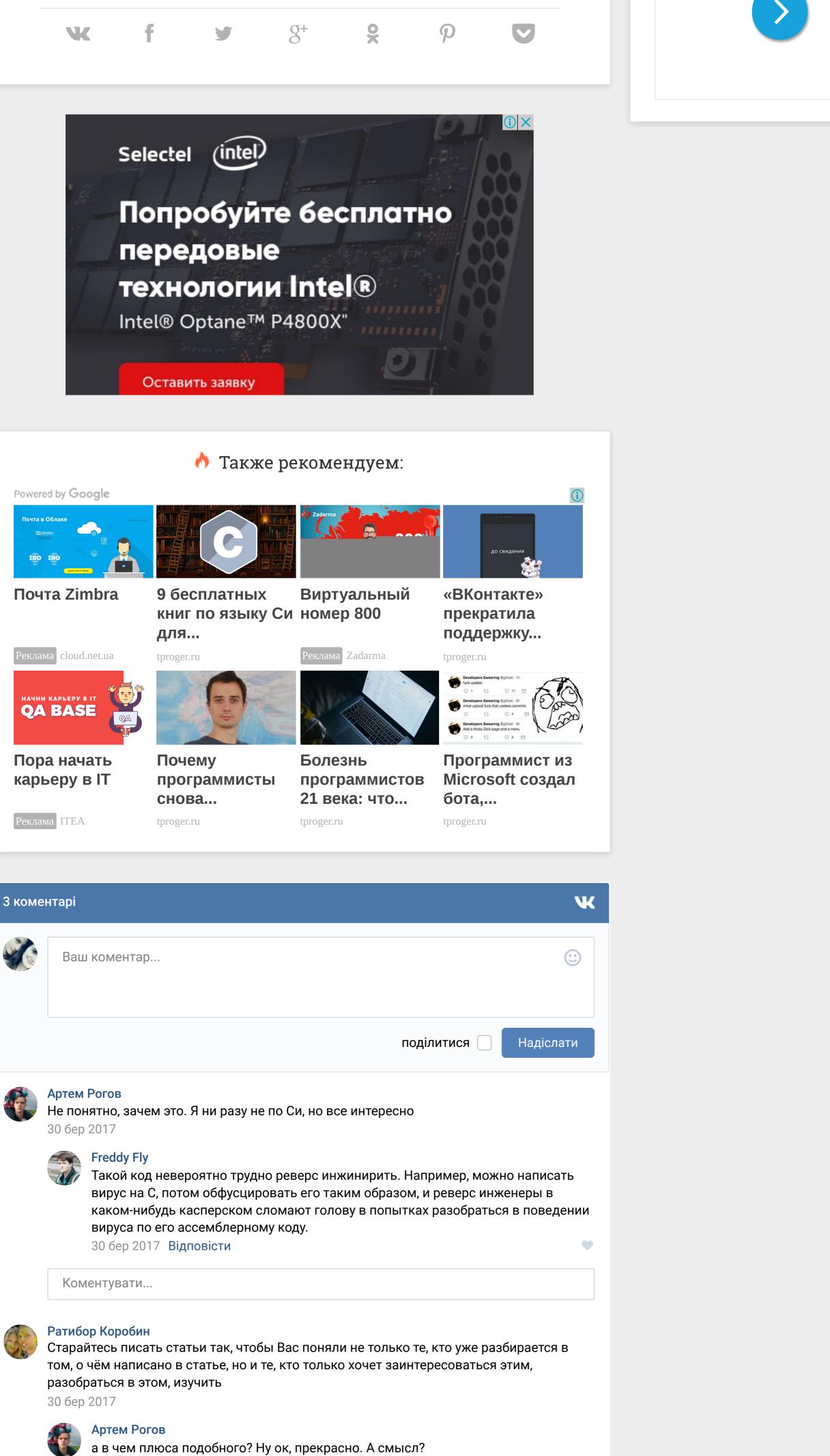


Google Chrome Hardware IDE

iOS Java JavaScript Linux Microsoft Open Source PHP Python Windows Windows 10 Алгоритмы Безопасность Спасибо за внимательность. Опечатка уже отправлена нашим редакторам. ИИ Python C++ C# Си Web Интернет Искусственный интеллект Лучшая практика Материалы от друзей Tproger







Источник: GitHub

30 бер 2017 Відповісти

30 бер 2017 Відповісти

Коментувати...

Никита Тюменцев

Best. Obfuscator. In. The. World.

30 бер 2017 Коментувати

Филипп Усков

**9** 1

¥

¥