8. United Nations Economic Commission for Europe (UNECE); Bureau International des Poids et Mesures (BIPM). Role of Metrology, Standards and Conformity Assessment in Global Quality Infrastructure. — URL: https://unece.org/trade/wp6 (дата обращения: 10.09.2025).

УДК 004.056.55:004.272.2

И. С. Чередайло
Учреждение образования «Барановичский государственный университет», Барановичи, Республика Беларусь

Научный руководитель А. И. Калько

МНОГОПОТОЧНАЯ БРУТФОРС-ПОДБОРКА ПАРОЛЕЙ: ВЛИЯНИЕ ДЛИНЫ КЛЮЧА И ЧИСЛА ПОТОКОВ

Введение. В современном цифровом пространстве пароли остаются основным механизмом защиты информации, от учётных записей в интернете до конфиденциальных документов Microsoft Office. С развитием вычислительных платформ и распространением многопоточности брутфорс-подбор однажды простых паролей может быть выполнен в считанные минуты, что существенно повышает риски несанкционированного доступа и кражи данных. Оценка времени, необходимого для полного перебора ключевого пространства, позволяет настроить требования к длине и сложности пароля, а также определить слабые места в используемом оборудовании [1].

В рамках исследования ставились следующие задачи:

1. Получить экспериментальные данные по времени перебора паролей длиной от 3 до 6 символов на ПК уровня Intel Core i5-10400 при 1, 2, 4 и 8 потоках.

2. Экстраполировать различные сценарии многопоточности на более длинные пароли (8, 10, 12, 16 и 20 символов) и оценить теоретическое время полного перебора.

3. Проанализировать зависимость ускорения от числа потоков, нагрузку на процессор и объём потребляемой оперативной памяти для всех сценариев.

Основная часть. Брутфорс — универсальный метод подбора паролей, основанный на полном переборе всех возможных комбинаций символов из заданного алфавита. При этом любая потенциальная строка формируется последовательно и проверяется на совпадение с заданным ключом до тех пор, пока подбор не завершится успехом или пока не будут исчерпаны все варианты.

Многопоточность — способ организации выполнения программы, при котором её алгоритм разделяется на несколько независимых или слабо связанных «потоков» (threads). Каждый поток работает в своём контексте, выполняя часть общего объёма задач параллельно с другими. Это позволяет задействовать несколько ядер процессора и ускорять обработку больших наборов данных за счёт одновременного выполнения нескольких последовательностей инструкций [2].

Экстраполяция — метод прогнозирования значений за пределами области экспериментальных данных на основе построенной модели. В контексте данного исследования экстраполяция используется для оценки времени перебора длинных паролей ($L \geq 8$), не замеренных напрямую.

Теоретические основы:

При алфавите из N символов (латинские прописные буквы и цифры, $N = 36$) объём пространства ключей длины L выражается формулой: $V = 36^L$.

Последовательное время перебора пропорционально V, а при T потоках в идеале сокращается примерно в T раз. Фактическое ускорение ниже из-за:

– накладных расходов на создание и завершение потоков;
– дисбаланса задач между потоками;
– переключений контекста ОС;
– конкуренции за кэш и подсистему памяти.

Аппаратная платформа и методика. Эксперименты проводились на следующем оборудовании:

– Процессор: Intel Core i5-10400 (6 физических ядер / 12 логических потоков, 2,9 ГГц, Turbo Boost до 4,3 ГГц).
– Оперативная память: 16 ГБ DDR4-2666.
– Накопитель: SSD Samsung 970 EVO NVMe 500 ГБ.
– Операционная система: Windows 10 Pro (64-bit).
– Среда разработки: Delphi 10.4 Sydney, Win64.

Алфавит: 26 латинских прописных букв + 10 цифр (0–9) = 36 символов. Сценарии многопоточности: 1, 2, 4, 8 потоков. Длины паролей L: 3, 4, 5, 6 (эксперимент), 8, 10, 12, 16, 20 (теоретическая экстраполяция).

Методика:

1. Каждый эксперимент повторялся трижды, результаты усреднялись.
2. Измерялись:
1. Полное время перебора (в секундах или часах);

2. Средняя загрузка CPU (в % по всем ядрам);
3. Пиковый объём потребляемой оперативной памяти (в МБ).

Результаты для L = 3…6 представлены в таблице 1.

Таблица 1 — Результаты для L = 3…6

Длина L	Пространство $V = 36^L$	Потоков	Время, с	CPU, %	Пик ОЗУ, МБ
3	46 656	1	0,11	12	48
3	46 656	2	0,06	22	50
3	46 656	4	0,04	38	54
3	46 656	8	0,03	55	60
4	1 679 616	1	3,80	10	52
4	1 679 616	2	2,00	21	54
4	1 679 616	4	1,10	36	60
4	1 679 616	8	0,75	60	65
5	60 466 176	1	134,50	8	60
5	60 466 176	2	67,80	17	64
5	60 466 176	4	34,20	32	68
5	60 466 176	8	21,50	55	75
6	2 176 782 336	1	4 930,00	5	68
6	2 176 782 336	2	2 470,00	14	74
6	2 176 782 336	4	1 250,00	23	82
6	2 176 782 336	8	820,00	42	100

Теоретическая экстраполяция для L = 8…20 представлена в таблице 2.

Таблица 2 — Теоретическая экстраполяция для L = 8…20

Длина L	36^L (млрд)	1 поток	2 потока	4 потока	8 потоков
8	2,82	472,0	238,0	132,5	84,0
10	60,47	10 130	5 075	2 825	1 790
12	1 296,0	217 000	108 500	60 000	38 000
16	28 200 000	4 720 000	2 360 000	1 320 000	840 000
20	3 656 000 000	613 000 000	306 500 000	162 000 000	93 250 000

Анализ зависимостей:

1. Экспоненциальный рост времени. Последовательное время перебора увеличивается крайне быстро с ростом длины пароля. Уже при L = 8 один поток требует сотни часов, а при L = 12 — десятки тысяч часов.
2. Выигрыш от параллелизма. Многопоточность в 2 и 4 потока даёт ускорение близкое к идеальному для больших пространств ($L \geq 8$), поскольку накладные расходы на синхронизацию становятся незначительными по сравнению с общим временем перебора [3].
3. Лимиты многопоточности. При 8 потоках ускорение не достигает 8× из-за конкуренции за кэш и переключений контекста. Чем больше L, тем ближе ускорение к числу потоков, так как фиксированные затраты «распыляются» на больший объём работы.
4. Загрузка CPU и память. Загрузка процессора растёт почти линейно до 60–70 % при 8 потоках, что соответствует возможностям 6 ядер / 12 потоков. Объём памяти увеличивается от 48 МБ до ~100 МБ за счёт стеков и внутренних буферов — несущественно для современных систем.

Практические выводы:

1. Для паролей длиной больше 8 символов даже 8-поточный брутфорс остаётся чрезмерно долгим; требуется GPU-акселерация (использование графического процессора для массового параллелизма) или гибридные методы (словарный перебор + маски) [4].
2. На платформе Core i5-10400 оптимальным оказывается 4 потока: дальнейшее увеличение даёт всё меньший выигрыш.
3. Для средних длин (L = 5…8) четырёхпоточный режим ускоряет перебор в 3,5–3,8 раза без значительных затрат памяти.
4. Динамическое распределение диапазонов потоков снижает дисбаланс и повышает общую эффективность.

Заключение. Исследование подтвердило экспоненциальный рост времени перебора с длиной пароля и практически линейный выигрыш от распараллеливания до числа физических ядер. Алгоритм потребляет мало памяти, но чувствителен к аппаратным характеристикам — числу ядер, объёму кэша и пропускной способности памяти. Для объективной оценки стойкости пароля важно учитывать и параметры целевой платформы.

Список цитируемых источников

1. *Иванова, Е. В.* Многопоточность в современных приложениях / Е. В. Иванова. — М. : Информатика, 2020. — 240 с.
2. *Кузнецов, П. В.* Эффективное использование CPU и GPU в криптоанализе / П. В. Кузнецов. — М. : Наука, 2021. — 280 с.
3. *Калько, А. И.* Проектирование приложений с использованием DLL-библиотек / А. И. Калько, А. А. Масло, О. И. Наранович // Содружество наук. Барановичи-2015 : материалы XI Междунар. науч.-практ. конф. молодых исследователей, Барановичи, 21-22 мая 2015 г. : в 3 кн. / редкол.: А. В. Никишова (гл. ред.) [и др.]. — Барановичи : РИО БарГУ, 2015. — Кн. 2. — С. 37–40.
4. Intel 64 and IA-32 Architectures Software Developer's Manual / Intel Corporation. — 2020. — 1024 с.

УДК 004.457+159.972

Е. Г. Шапович
Учреждение образования «Барановичский государственный университет», Барановичи, Республика Беларусь

РАЗРАБОТКА ПРИЛОЖЕНИЯ ДЛЯ БОРЬБЫ С ПАНИЧЕСКИМИ АТАКАМИ

Введение. Разработка мобильных приложений для поддержания психического здоровья становится все более актуальной, особенно в контексте растущего числа людей, сталкивающихся с паническими атаками и повышенной тревожностью. Эти состояния, характеризующиеся внезапными и интенсивными приступами страха, могут значительно ухудшить качество жизни, приводя к избеганию социальных ситуаций и ограничению повседневной деятельности. В последние годы технологии предлагают новые подходы к управлению этими состояниями, предоставляя пользователям доступ к инструментам и техникам самопомощи прямо на своих смартфонах.

Данная статья посвящена процессу разработки приложения на платформе Flutter, цель которого — предоставить эффективные и доступные средства для борьбы с паническими атаками. Мы рассмотрим, как мультисенсорные методы — от дыхательных упражнений и звуковой терапии до тактильной обратной связи и визуализации, которые будут интегрированы в единое приложение для создания комплексного инструмента поддержки. Это приложение предлагает ряд интерактивных упражнений, таких как регулирование дыхания с визуальными подсказками, фокусировка на звуках природы, вибрационная стимуляция, а также упражнения на воображение запахов и вкусов, дополненные игрой «Лопание пузырей» для когнитивной концентрации.

Цель этой работы не только описать технические аспекты создания такого приложения, но и подчеркнуть его потенциал в качестве средства для обучения саморегуляции и восстановления контроля во время приступов паники.

Основная часть. Для реализации приложения была выбрана кроссплатформенная среда Flutter, разработанная компанией Google. Flutter — это UI-фреймворк с открытым исходным кодом, позволяющий создавать нативные приложения для Android, iOS, веб- и настольных платформ с использованием единой кодовой базы. Основным языком программирования во Flutter является Dart — современный, объектно-ориентированный язык с поддержкой асинхронного программирования, JIT- и AOT-компиляцией, что обеспечивает как высокую производительность, так и удобство разработки [1].

Одной из ключевых задач приложения является визуальное сопровождение дыхательных и сенсорных упражнений: плавные анимации, визуализация звукового спектра, интерактивные элементы (например, лопание пузырей). Flutter использует собственный рендеринг-движок Skia, который отрисовывает каждый пиксель напрямую, минуя нативные UI-компоненты операционной системы. Это позволяет добиваться стабильной частоты 60 (а на поддерживаемых устройствах — 120) кадров в секунду [1], что критически важно при создании успокаивающего и отзывчивого пользовательского приложения.

Рисунок 1 — Главная страница

Приложение для борьбы с паническими атаками состоит из шести терапевтических модулей, объединённых единой эстетикой и принципами осознанности. Каждый экран спроектирован как «точка опоры» — простой, безопасный, без давления. Ниже представлены основные экраны приложения с пояснением их функциональности.

Главная страница (рисунок 1) представляет собой интуитивно понятную экран с изображением воинственного облака, задающий вопрос: «У вас паническая атака?». Центральное место занимает иллюстрация: милый мозг с мечом, противостоящий призраку тревоги. Это метафора внутренней силы — не борьбы, а осознанного действия. Фон — градиент от синего к бирюзовому, что способствует снижению визуального напряжения.

При нажатии на кнопку «Давайте начнем» откроется экран с дыхательным упражнением, представленный на рисунке 2.

Экран дыхательного упражнения построен вокруг анимированного облака, которое плавно увеличивается во время вдоха (3 секунды), замирает на задержке (3 секунды) и уменьшается при выдохе (3 секунды).