МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

Институт №8 «Компьютерные науки и прикладная математика»
Кафедра 806 «Вычислительная математика и программирование»

Лабораторная работа №2
по курсу «Операционные системы»

Выполнил: Д. М. Мишин
Группа: М8О-207БВ-24
Преподаватель: Е. С. Миронов

Москва, 2025

Условие

Цель работы:
Приобретение практических навыков в:

- Управление потоками в ОС
- Обеспечение синхронизации между потоками

Задание:
Составить программу на языке C++, обрабатывающую данные в многопоточном режиме. При обработки использовать стандартные средства создания потоков операционной системы (Windows/Unix). Ограничение максимального количества потоков, работающих в один момент времени, должно быть задано ключом запуска вашей программы. Так же необходимо уметь продемонстрировать количество потоков, используемое вашей программой с помощью стандартных средств операционной системы. В отчете привести исследование зависимости ускорения и эффективности алгоритма от входных данных и количества потоков. Получившиеся результаты необходимо объяснить.

Вариант: 19
Дан массив координат (x, y). Пользователь вводит число кластеров. Проведите кластеризацию методом k-средних

Метод решения

Алгоритм решения задачи:

1. Пользователь запускает программу с целочисленным ключом запуска, означающем максимальное количество потоков которое может использовать программа.
2. Пользователь вводит в консоль количество кластеров - n.
3. Генерируется массив из 10 000 точек на плоскости Oxy и добавляются в общий массив.
4. Первыми n точчками инициализируются центройды кастеров.
5. Каждый поток проверяет к какому кластеру точка в "своей"части массива точек и ближе и перемещает ее туда.
6. После того, как все потоки отработали, главный поток пересчитывает центройды.
7. Если центройды изменили свое положение, то потоки снова начинают перемещать точки.
8. Алгоритм завершается если положение центройдов не изменилось или количество итераций превысило 1000.

Архитектура программы:

```
lab2/
├── build/
├── include/
│   ├── thread.h
│   └── exceptions.h
├── src/
│   └── thread.cpp
└── main.cpp
```

Описание программы

`main.cpp` — реализация алгоритма кластеризации k средних.
Основные функции:

- `void updateCentroids();` — вспомогательная функция изменения центройдов кластеров.
- `void* updateClusters(void* threadData);` — вспомагательная функция "перемещения"точек в кластеры.
- `kMeans();` — основная функция реализации алгоритма кластеризации методом k средних.

`exceptions.h` — объявление необходимых исключений.

- `CreateThreadException` — ошибка создания потока.
- `WaitThreadException` — ошибка ожидания завершения потока.

`include/thread.h` — объявления методов класса thread.
`src/thread.cpp` — реализация.

Основные функции:

- `void Run(void* threadData);` — создание и запуск. Используется системный вызов `pthread_create(...)`.
- `void Join();` — ожидание завершения потока.
  Используется системный вызов `pthread_join(...)`.

Результаты

Рис. 1: Диаграмма зависимости времени от количества потоков.

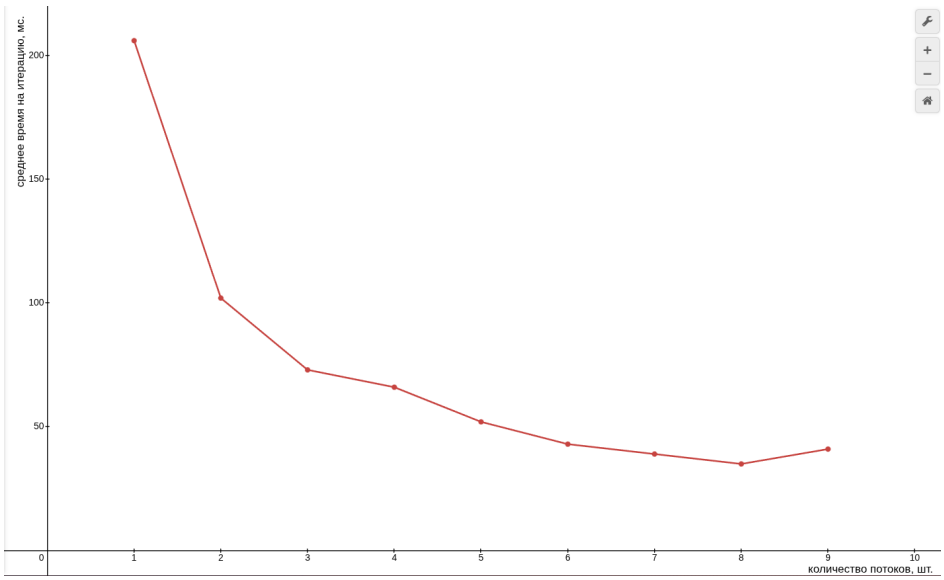

Рис. 2: График зависимости времени от количества потоков.

Результатом работы программы является время (мс) за которое алгоритм выполняет кластеризацию и количество итераций, которое понадобилось. При входных данных 10 000 точек и 500 класстеров получены следующие данные:

- 1 поток — 28 итераций, 5771 мс -> 206 мс на итерацию. (BASE)
- 2 потока — 27 итераций, 2764 мс -> 102 мс на итерацию. (x2.0)
- 3 потока — 28 итераций, 2064 мс -> 73 мс на итерацию. (x2.8)
- 4 потока — 22 итерации, 1358 мс -> 66 мс на итерацию. (x3.1)
- 5 потоков — 53 итерации, 2766 мс -> 52 мс на итерацию. (x4.0)
- 6 потоков — 31 итерация, 1320 мс -> 43 мс на итерацию. (x4.8)
- 7 потоков — 25 итераций, 978 мс -> 39 мс на итерацию. (x5.3)
- 8 потоков — 20 итераций, 698 мс -> 35 мс на итерацию. (x5.9)
- 9 потоков — 28 итераций, 1145 мс -> 41 мс на итерацию. (x5.0)

Выводы

- Увеличение количества потоков не всегда дает прирост к скорости. В случае, когда количестов потоков больше 8, время затрачивается как на синхронную часть программы, так и на создание и завершение потоков, поэтому увеличение потоков не дает желаемых результатов или даже увеличивает время работы программы. Также существуют накладные расходы на работу процессора и планировщика ОС с потоками.
- Ускорение при увелеении количества потоков всегда будет стремиться к линейности.

Исходная программа

```
#pragma once

#include <exception>
#include <string>

namespace exceptions {
    class CreateThreadException: std::exception {
    public:
        explicit CreateThreadException(const std::string& text): error_message_(text) {}
        const char* what() const noexcept override {
            return error_message_.data();
        }
    private:
        std::string error_message_;
    };

    class WaitThreadException: std::exception {
    public:
        explicit WaitThreadException(const std::string& text): error_message_(text) {}
        const char* what() const noexcept override {
            return error_message_.data();
        }
    private:
        std::string error_message_;
    };
}
```

Листинг 1: exceptions.h

```
#pragma once

#include <pthread.h>

#include "exceptions.h"

namespace thread {
    using threadFunc = typeof(void*(void*))*;
    struct threadInfo;
    using threadHandle = threadInfo*;

    class Thread {
      private:
        threadFunc func;
        threadHandle handle;
      public:
        Thread(threadFunc func);
        Thread(const Thread&) = delete;
        Thread& operator=(const Thread&) = delete;
        Thread(Thread&& other);
        Thread& operator=(Thread&& other);
        void Run(void* threadData);
        void Join();
        ~Thread() noexcept;
    };
}
```

Листинг 2: thread.h

```cpp
#include "thread.h"

namespace thread {
    struct threadInfo {
        pthread_t thread;
    };

    Thread::Thread(threadFunc func): func(func) {
        handle = new threadInfo();
    }

    Thread::Thread(Thread&& other): func(other.func), handle(other.handle) {
        other.func = nullptr;
        other.handle = nullptr;
    }

    Thread& Thread::operator=(Thread&& other) {
        Thread temp = std::move(other);
        std::swap(func, temp.func);
        std::swap(handle, temp.handle);
        return *this;
    }

    void Thread::Run(void* data) {
        int res = pthread_create(&(handle->thread), NULL, func, data);
        if (res != 0) {
            throw exceptions::CreateThreadException("failed to create thread");
        }
    }

    void Thread::Join() {
        int res = pthread_join(handle->thread, NULL);
        if (res != 0) {
            throw exceptions::WaitThreadException("Failed to wait thred");
        }
    }

    Thread::~Thread() {
        delete handle;
    }
}
```

Листинг 3: thread.cpp

```cpp
#include <cmath>
#include <cstdlib>
#include <vector>
#include <iostream>
#include <random>
#include <chrono>

#include "thread.h"

struct Point {
    float x;
    float y;
    int32_t cluster;

    Point(float x = 0.0, float y = 0.0, int32_t cluster = -1): x(x), y(y), cluster(
        cluster) {}
```

```
16 };
17
18 struct Cluster {
19     Point centroid;
20     Cluster(const Point& point): centroid(point) {}
21 };
22
23 struct ThreadData {
24     int32_t begin;
25     int32_t end;
26     int32_t id;
27 };
28
29 double distance(const Point& a, const Point& b) {
30     return sqrt(pow(a.x - b.x, 2) + pow(a.y - b.y, 2));
31 }
32
33 const int32_t MAXITERATIONS = 1000;
34 const double EPS = 1e-6;
35
36 int32_t threadsCount;
37 int32_t clustersCount;
38
39 std::vector <Point> allPoints;
40 std::vector <Cluster> allClusters;
41
42 void generatePointsAndClusters(int32_t count = 10000) {
43     std::random_device rd;
44     std::mt19937 gen(rd());
45     std::uniform_real_distribution<> dis_x(-10000, 10000);
46     std::uniform_real_distribution<> dis_y(-10000, 10000);
47
48     allPoints.clear();
49     for (size_t i = 0; i < count; ++i) {
50         allPoints.emplace_back(dis_x(gen), dis_y(gen), -1);
51     }
52
53     if (clustersCount > allPoints.size()) {
54         std::cerr << "Error: count points must be greater than clusters count" << std::
              endl;
55         exit(1);
56     }
57
58     allClusters.clear();
59     for (size_t i = 0; i < clustersCount; ++i) {
60         Cluster cl = Cluster(Point(allPoints[i].x, allPoints[i].y, i));
61         allClusters.push_back(cl);
62     }
63 }
64
65 void* updateClusters(void* threadData) {
66     ThreadData* td = static_cast<ThreadData*>(threadData);
67
68     if (td == nullptr) {
69         std::cerr << "Error: thead data mustn't be nullptr." << std::endl;
70         return nullptr;
71     }
72
73     for (size_t i = td->begin; i < td->end; ++i) {
74         double baseDist = std::numeric_limits<double>::max();
```

```
75          int cluster = 0;
76          for (size_t j = 0; j < allClusters.size(); ++j) {
77              double dist = distance(allPoints[i], allClusters[j].centroid);
78              if (dist < baseDist) {
79                  baseDist = dist;
80                  cluster = j;
81              }
82          }
83          allPoints[i].cluster = cluster;
84      }
85      return nullptr;
86  }
87
88  void updateCentroids() {
89      std::vector<int32_t> count(clustersCount, 0);
90      std::vector<double> xSums(clustersCount, 0);
91      std::vector<double> ySums(clustersCount, 0);
92
93      for (const Point& p: allPoints) {
94          xSums[p.cluster] += p.x;
95          ySums[p.cluster] += p.y;
96          ++count[p.cluster];
97      }
98
99      for (size_t i = 0; i < allClusters.size(); ++i) {
100         if (count[i] > 0) {
101             allClusters[i].centroid.x = xSums[i] / count[i];
102             allClusters[i].centroid.y = ySums[i] / count[i];
103         }
104     }
105 }
106
107 void kMeans() {
108     generatePointsAndClusters();
109
110     for (size_t j = 0; j < MAXITERATIONS; ++j) {
111         std::vector<thread::Thread> threads;
112         threads.reserve(threadsCount);
113         std::vector<ThreadData> threadsData(threadsCount);
114
115         int32_t chunkSize = allPoints.size() / threadsCount;
116
117         for (size_t i = 0; i < threadsCount; ++i) {
118
119             threadsData[i].begin = i * chunkSize;
120             threadsData[i].end = (i == threadsCount - 1) ? allPoints.size() : (i + 1) *
                    chunkSize;
121             threadsData[i].id = i;
122
123             try {
124                 threads.emplace_back(updateClusters);
125                 threads[i].Run(&threadsData[i]);
126             } catch (const exceptions::CreateThreadException& e) {
127                 std::cerr << e.what() << std::endl;
128                 for (size_t i = 0; i < threads.size(); ++i) {
129                     threads[i].~Thread();
130                 }
131                 exit(1);
132             }
133         }
```

```
134
135         try {
136             for (size_t i = 0; i < threadsCount; ++i) {
137                 threads[i].Join();
138             }
139         } catch (const exceptions::WaitThreadException& e) {
140             std::cerr << e.what() << std::endl;
141             for (size_t i = 0; i < threads.size(); ++i) {
142                 threads[i].~Thread();
143             }
144             exit(1);
145         }
146
147         std::vector<Cluster> oldClasters = allClusters;
148         updateCentroids();
149
150         bool changed = false;
151         for (size_t i = 0; i < clustersCount; ++i) {
152             if (distance(oldClasters[i].centroid, allClusters[i].centroid) > EPS) {
153                 changed = true;
154                 break;
155             }
156         }
157         if (!changed) {
158             std::cout << "Algorithm ended on " << j + 1 << " iterations" << std::endl;
159             break;
160         }
161     }
162 }
163
164 int main(int argc, char* argv[]) {
165     std::cout << "Enter the clusters count: " << std::endl;
166     std::cin >> clustersCount;
167
168     if (argc < 2) {
169         std::cerr << "Must be min 2 args" << std::endl;
170         return 1;
171     }
172
173     try {
174         threadsCount = std::stoul(argv[1]);
175     } catch (const std::invalid_argument&) {
176         std::cerr << "Error: incorrect number format: not a number" << std::endl;
177     } catch (const std::out_of_range&) {
178         std::cerr << "Error: incorrect number format: too long." << std::endl;
179     }
180
181     if (threadsCount > 10000) {
182         std::cerr << "Incorrect threads count." << std::endl;
183         return 1;
184     }
185
186     auto start = std::chrono::steady_clock::now();
187     kMeans();
188     auto end = std::chrono::steady_clock::now();
189
190     std::cout << "Algorithm was done in: " << std::chrono::duration_cast<std::chrono::
         milliseconds>(end - start).count() << " ms" << std::endl;
191     return 0;
```

 }

Листинг 4: main.cpp

Strace

Strace при входных данных — 10000 кластеров, 4 потока, 500 кластеров.

```
execve("./main", ["./main", "4"], 0x7ffd413b2088 /* 77 vars */) = 0
brk(NULL)                               = 0x5b4a7b96c000
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
↪  0x79ba83e07000
access("/etc/ld.so.preload", R_OK)      = -1 ENOENT (Нет такого файла или
↪  каталога)
openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
fstat(3, {st_mode=S_IFREG|0644, st_size=74871, ...}) = 0
mmap(NULL, 74871, PROT_READ, MAP_PRIVATE, 3, 0) = 0x79ba83df4000
close(3)                                = 0
openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libstdc++.so.6", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\0\0\0\0\0\0\0"..., 832)
↪  = 832
fstat(3, {st_mode=S_IFREG|0644, st_size=2592224, ...}) = 0
mmap(NULL, 2609472, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x79ba83a00000
mmap(0x79ba83a9d000, 1343488, PROT_READ|PROT_EXEC,
↪  MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x9d000) = 0x79ba83a9d000
mmap(0x79ba83be5000, 552960, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
↪  0x1e5000) = 0x79ba83be5000
mmap(0x79ba83c6c000, 57344, PROT_READ|PROT_WRITE,
↪  MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x26b000) = 0x79ba83c6c000
mmap(0x79ba83c7a000, 12608, PROT_READ|PROT_WRITE,
↪  MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x79ba83c7a000
close(3)                                = 0
openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libm.so.6", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\0\0\0\0\0\0\0"..., 832)
↪  = 832
fstat(3, {st_mode=S_IFREG|0644, st_size=952616, ...}) = 0
mmap(NULL, 950296, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x79ba83d0b000
mmap(0x79ba83d1b000, 520192, PROT_READ|PROT_EXEC,
↪  MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x10000) = 0x79ba83d1b000
mmap(0x79ba83d9a000, 360448, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
↪  0x8f000) = 0x79ba83d9a000
mmap(0x79ba83df2000, 8192, PROT_READ|PROT_WRITE,
↪  MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0xe7000) = 0x79ba83df2000
close(3)                                = 0
openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libgcc_s.so.1", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\0\0\0\0\0\0\0"..., 832)
↪  = 832
fstat(3, {st_mode=S_IFREG|0644, st_size=183024, ...}) = 0
mmap(NULL, 185256, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x79ba83cdd000
mmap(0x79ba83ce1000, 147456, PROT_READ|PROT_EXEC,
↪  MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x4000) = 0x79ba83ce1000
mmap(0x79ba83d05000, 16384, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
↪  0x28000) = 0x79ba83d05000
```

```
mmap(0x79ba83d09000, 8192, PROT_READ|PROT_WRITE,
↪  MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x2b000) = 0x79ba83d09000
close(3)                                = 0
openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\220\243\2\0\0\0\0\0"...,
↪  832) = 832
pread64(3, "\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0"...,
↪  784, 64) = 784
fstat(3, {st_mode=S_IFREG|0755, st_size=2125328, ...}) = 0
pread64(3, "\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0"...,
↪  784, 64) = 784
mmap(NULL, 2170256, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x79ba83600000
mmap(0x79ba83628000, 1605632, PROT_READ|PROT_EXEC,
↪  MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x28000) = 0x79ba83628000
mmap(0x79ba837b0000, 323584, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
↪  0x1b0000) = 0x79ba837b0000
mmap(0x79ba837ff000, 24576, PROT_READ|PROT_WRITE,
↪  MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1fe000) = 0x79ba837ff000
mmap(0x79ba83805000, 52624, PROT_READ|PROT_WRITE,
↪  MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x79ba83805000
close(3)                                = 0
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
↪  0x79ba83cdb000
mmap(NULL, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
↪  0x79ba83cd8000
arch_prctl(ARCH_SET_FS, 0x79ba83cd8740) = 0
set_tid_address(0x79ba83cd8a10)         = 214366
set_robust_list(0x79ba83cd8a20, 24)     = 0
rseq(0x79ba83cd9060, 0x20, 0, 0x53053053) = 0
mprotect(0x79ba837ff000, 16384, PROT_READ) = 0
mprotect(0x79ba83d09000, 4096, PROT_READ) = 0
mprotect(0x79ba83df2000, 4096, PROT_READ) = 0
mprotect(0x79ba83c6c000, 45056, PROT_READ) = 0
mprotect(0x5b4a6c0e2000, 4096, PROT_READ) = 0
mprotect(0x79ba83e45000, 8192, PROT_READ) = 0
prlimit64(0, RLIMIT_STACK, NULL, {rlim_cur=8192*1024, rlim_max=RLIM64_INFINITY}) =
↪  0
munmap(0x79ba83df4000, 74871)           = 0
futex(0x79ba83c7a7bc, FUTEX_WAKE_PRIVATE, 2147483647) = 0
getrandom("\x16\x50\x3a\x1b\x86\x7e\x4e\xed", 8, GRND_NONBLOCK) = 8
brk(NULL)                               = 0x5b4a7b96c000
brk(0x5b4a7b98d000)                     = 0x5b4a7b98d000
fstat(1, {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0), ...}) = 0
write(1, "Enter the clusters count: \n", 27) = 27
fstat(0, {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0), ...}) = 0
read(0, "500\n", 1024)                  = 4
brk(0x5b4a7b9b7000)                     = 0x5b4a7b9b7000
mmap(NULL, 200704, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
↪  0x79ba83ca7000
brk(0x5b4a7b9a0000)                     = 0x5b4a7b9a0000
rt_sigaction(SIGRT_1, {sa_handler=0x79ba83699530, sa_mask=[],
↪  sa_flags=SA_RESTORER|SA_ONSTACK|SA_RESTART|SA_SIGINFO,
↪  sa_restorer=0x79ba83645330}, NULL, 8) = 0
```

```
rt_sigprocmask(SIG_UNBLOCK, [RTMIN RT_1], NULL, 8) = 0
mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1, 0) =
↪  0x79ba82dff000
mprotect(0x79ba82e00000, 8388608, PROT_READ|PROT_WRITE) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV⌋
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba835ff990, parent_tid=0x79ba835ff990, exit_signal=0,
↪  stack=0x79ba82dff000, stack_size=0x7fff80, tls=0x79ba835ff6c0} =>
↪  {parent_tid=[214396]}, 88) = 214396
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1, 0) =
↪  0x79ba825fe000
mprotect(0x79ba825ff000, 8388608, PROT_READ|PROT_WRITE) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV⌋
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba82dfe990, parent_tid=0x79ba82dfe990, exit_signal=0,
↪  stack=0x79ba825fe000, stack_size=0x7fff80, tls=0x79ba82dfe6c0} =>
↪  {parent_tid=[214397]}, 88) = 214397
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1, 0) =
↪  0x79ba81dfd000
mprotect(0x79ba81dfe000, 8388608, PROT_READ|PROT_WRITE) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV⌋
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba825fd990, parent_tid=0x79ba825fd990, exit_signal=0,
↪  stack=0x79ba81dfd000, stack_size=0x7fff80, tls=0x79ba825fd6c0} =>
↪  {parent_tid=[214398]}, 88) = 214398
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1, 0) =
↪  0x79ba815fc000
mprotect(0x79ba815fd000, 8388608, PROT_READ|PROT_WRITE) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV⌋
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba81dfc990, parent_tid=0x79ba81dfc990, exit_signal=0,
↪  stack=0x79ba815fc000, stack_size=0x7fff80, tls=0x79ba81dfc6c0} =>
↪  {parent_tid=[214399]}, 88) = 214399
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
futex(0x79ba835ff990, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 214396, NULL,
↪  FUTEX_BITSET_MATCH_ANY) = 0
futex(0x79ba82dfe990, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 214397, NULL,
↪  FUTEX_BITSET_MATCH_ANY) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV⌋
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba81dfc990, parent_tid=0x79ba81dfc990, exit_signal=0,
↪  stack=0x79ba815fc000, stack_size=0x7fff80, tls=0x79ba81dfc6c0} =>
↪  {parent_tid=[214413]}, 88) = 214413
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
```

```
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV⌋
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba825fd990, parent_tid=0x79ba825fd990, exit_signal=0,
↪  stack=0x79ba81dfd000, stack_size=0x7fff80, tls=0x79ba825fd6c0} =>
↪  {parent_tid=[214414]}, 88) = 214414
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV⌋
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba82dfe990, parent_tid=0x79ba82dfe990, exit_signal=0,
↪  stack=0x79ba825fe000, stack_size=0x7fff80, tls=0x79ba82dfe6c0} =>
↪  {parent_tid=[214415]}, 88) = 214415
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV⌋
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba835ff990, parent_tid=0x79ba835ff990, exit_signal=0,
↪  stack=0x79ba82dff000, stack_size=0x7fff80, tls=0x79ba835ff6c0} =>
↪  {parent_tid=[214416]}, 88) = 214416
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
futex(0x79ba81dfc990, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 214413, NULL,
↪  FUTEX_BITSET_MATCH_ANY) = 0
futex(0x79ba825fd990, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 214414, NULL,
↪  FUTEX_BITSET_MATCH_ANY) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV⌋
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba835ff990, parent_tid=0x79ba835ff990, exit_signal=0,
↪  stack=0x79ba82dff000, stack_size=0x7fff80, tls=0x79ba835ff6c0} =>
↪  {parent_tid=[214417]}, 88) = 214417
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV⌋
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba82dfe990, parent_tid=0x79ba82dfe990, exit_signal=0,
↪  stack=0x79ba825fe000, stack_size=0x7fff80, tls=0x79ba82dfe6c0} =>
↪  {parent_tid=[214418]}, 88) = 214418
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV⌋
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba825fd990, parent_tid=0x79ba825fd990, exit_signal=0,
↪  stack=0x79ba81dfd000, stack_size=0x7fff80, tls=0x79ba825fd6c0} =>
↪  {parent_tid=[214419]}, 88) = 214419
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV⌋
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba81dfc990, parent_tid=0x79ba81dfc990, exit_signal=0,
↪  stack=0x79ba815fc000, stack_size=0x7fff80, tls=0x79ba81dfc6c0} =>
↪  {parent_tid=[214420]}, 88) = 214420
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
futex(0x79ba835ff990, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 214417, NULL,
↪  FUTEX_BITSET_MATCH_ANY) = 0
```

```
futex(0x79ba825fd990, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 214419, NULL,
↪  FUTEX_BITSET_MATCH_ANY) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV⌋
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba81dfc990, parent_tid=0x79ba81dfc990, exit_signal=0,
↪  stack=0x79ba815fc000, stack_size=0x7fff80, tls=0x79ba81dfc6c0} =>
↪  {parent_tid=[214421]}, 88) = 214421
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV⌋
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba825fd990, parent_tid=0x79ba825fd990, exit_signal=0,
↪  stack=0x79ba81dfd000, stack_size=0x7fff80, tls=0x79ba825fd6c0} =>
↪  {parent_tid=[214422]}, 88) = 214422
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV⌋
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba82dfe990, parent_tid=0x79ba82dfe990, exit_signal=0,
↪  stack=0x79ba825fe000, stack_size=0x7fff80, tls=0x79ba82dfe6c0} =>
↪  {parent_tid=[214423]}, 88) = 214423
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV⌋
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba835ff990, parent_tid=0x79ba835ff990, exit_signal=0,
↪  stack=0x79ba82dff000, stack_size=0x7fff80, tls=0x79ba835ff6c0} =>
↪  {parent_tid=[214424]}, 88) = 214424
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
futex(0x79ba81dfc990, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 214421, NULL,
↪  FUTEX_BITSET_MATCH_ANY) = 0
futex(0x79ba82dfe990, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 214423, NULL,
↪  FUTEX_BITSET_MATCH_ANY) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV⌋
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba835ff990, parent_tid=0x79ba835ff990, exit_signal=0,
↪  stack=0x79ba82dff000, stack_size=0x7fff80, tls=0x79ba835ff6c0} =>
↪  {parent_tid=[214425]}, 88) = 214425
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV⌋
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba82dfe990, parent_tid=0x79ba82dfe990, exit_signal=0,
↪  stack=0x79ba825fe000, stack_size=0x7fff80, tls=0x79ba82dfe6c0} =>
↪  {parent_tid=[214426]}, 88) = 214426
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV⌋
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba825fd990, parent_tid=0x79ba825fd990, exit_signal=0,
↪  stack=0x79ba81dfd000, stack_size=0x7fff80, tls=0x79ba825fd6c0} =>
↪  {parent_tid=[214427]}, 88) = 214427
```

```
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba81dfc990, parent_tid=0x79ba81dfc990, exit_signal=0,
↪  stack=0x79ba815fc000, stack_size=0x7fff80, tls=0x79ba81dfc6c0} =>
↪  {parent_tid=[214428]}, 88) = 214428
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
futex(0x79ba835ff990, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 214425, NULL,
↪  FUTEX_BITSET_MATCH_ANY) = 0
futex(0x79ba82dfe990, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 214426, NULL,
↪  FUTEX_BITSET_MATCH_ANY) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba81dfc990, parent_tid=0x79ba81dfc990, exit_signal=0,
↪  stack=0x79ba815fc000, stack_size=0x7fff80, tls=0x79ba81dfc6c0} =>
↪  {parent_tid=[214429]}, 88) = 214429
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba825fd990, parent_tid=0x79ba825fd990, exit_signal=0,
↪  stack=0x79ba81dfd000, stack_size=0x7fff80, tls=0x79ba825fd6c0} =>
↪  {parent_tid=[214430]}, 88) = 214430
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba82dfe990, parent_tid=0x79ba82dfe990, exit_signal=0,
↪  stack=0x79ba825fe000, stack_size=0x7fff80, tls=0x79ba82dfe6c0} =>
↪  {parent_tid=[214431]}, 88) = 214431
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba835ff990, parent_tid=0x79ba835ff990, exit_signal=0,
↪  stack=0x79ba82dff000, stack_size=0x7fff80, tls=0x79ba835ff6c0} =>
↪  {parent_tid=[214432]}, 88) = 214432
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
futex(0x79ba81dfc990, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 214429, NULL,
↪  FUTEX_BITSET_MATCH_ANY) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba835ff990, parent_tid=0x79ba835ff990, exit_signal=0,
↪  stack=0x79ba82dff000, stack_size=0x7fff80, tls=0x79ba835ff6c0} =>
↪  {parent_tid=[214433]}, 88) = 214433
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
```

```
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV⌋
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba82dfe990, parent_tid=0x79ba82dfe990, exit_signal=0,
↪  stack=0x79ba825fe000, stack_size=0x7fff80, tls=0x79ba82dfe6c0} =>
↪  {parent_tid=[214434]}, 88) = 214434
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV⌋
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba825fd990, parent_tid=0x79ba825fd990, exit_signal=0,
↪  stack=0x79ba81dfd000, stack_size=0x7fff80, tls=0x79ba825fd6c0} =>
↪  {parent_tid=[214435]}, 88) = 214435
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV⌋
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba81dfc990, parent_tid=0x79ba81dfc990, exit_signal=0,
↪  stack=0x79ba815fc000, stack_size=0x7fff80, tls=0x79ba81dfc6c0} =>
↪  {parent_tid=[214436]}, 88) = 214436
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
futex(0x79ba835ff990, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 214433, NULL,
↪  FUTEX_BITSET_MATCH_ANY) = 0
futex(0x79ba81dfc990, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 214436, NULL,
↪  FUTEX_BITSET_MATCH_ANY) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV⌋
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba81dfc990, parent_tid=0x79ba81dfc990, exit_signal=0,
↪  stack=0x79ba815fc000, stack_size=0x7fff80, tls=0x79ba81dfc6c0} =>
↪  {parent_tid=[214437]}, 88) = 214437
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV⌋
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba825fd990, parent_tid=0x79ba825fd990, exit_signal=0,
↪  stack=0x79ba81dfd000, stack_size=0x7fff80, tls=0x79ba825fd6c0} =>
↪  {parent_tid=[214438]}, 88) = 214438
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV⌋
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba82dfe990, parent_tid=0x79ba82dfe990, exit_signal=0,
↪  stack=0x79ba825fe000, stack_size=0x7fff80, tls=0x79ba82dfe6c0} =>
↪  {parent_tid=[214439]}, 88) = 214439
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV⌋
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba835ff990, parent_tid=0x79ba835ff990, exit_signal=0,
↪  stack=0x79ba82dff000, stack_size=0x7fff80, tls=0x79ba835ff6c0} =>
↪  {parent_tid=[214440]}, 88) = 214440
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
futex(0x79ba81dfc990, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 214437, NULL,
↪  FUTEX_BITSET_MATCH_ANY) = 0
```

```
futex(0x79ba825fd990, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 214438, NULL,
↪  FUTEX_BITSET_MATCH_ANY) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV⌋
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba835ff990, parent_tid=0x79ba835ff990, exit_signal=0,
↪  stack=0x79ba82dff000, stack_size=0x7fff80, tls=0x79ba835ff6c0} =>
↪  {parent_tid=[214441]}, 88) = 214441
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV⌋
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba82dfe990, parent_tid=0x79ba82dfe990, exit_signal=0,
↪  stack=0x79ba825fe000, stack_size=0x7fff80, tls=0x79ba82dfe6c0} =>
↪  {parent_tid=[214442]}, 88) = 214442
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV⌋
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba825fd990, parent_tid=0x79ba825fd990, exit_signal=0,
↪  stack=0x79ba81dfd000, stack_size=0x7fff80, tls=0x79ba825fd6c0} =>
↪  {parent_tid=[214443]}, 88) = 214443
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV⌋
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba81dfc990, parent_tid=0x79ba81dfc990, exit_signal=0,
↪  stack=0x79ba815fc000, stack_size=0x7fff80, tls=0x79ba81dfc6c0} =>
↪  {parent_tid=[214444]}, 88) = 214444
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
futex(0x79ba835ff990, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 214441, NULL,
↪  FUTEX_BITSET_MATCH_ANY) = 0
futex(0x79ba82dfe990, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 214442, NULL,
↪  FUTEX_BITSET_MATCH_ANY) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV⌋
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba81dfc990, parent_tid=0x79ba81dfc990, exit_signal=0,
↪  stack=0x79ba815fc000, stack_size=0x7fff80, tls=0x79ba81dfc6c0} =>
↪  {parent_tid=[214445]}, 88) = 214445
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV⌋
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba825fd990, parent_tid=0x79ba825fd990, exit_signal=0,
↪  stack=0x79ba81dfd000, stack_size=0x7fff80, tls=0x79ba825fd6c0} =>
↪  {parent_tid=[214446]}, 88) = 214446
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV⌋
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba82dfe990, parent_tid=0x79ba82dfe990, exit_signal=0,
↪  stack=0x79ba825fe000, stack_size=0x7fff80, tls=0x79ba82dfe6c0} =>
↪  {parent_tid=[214447]}, 88) = 214447
```

```
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba835ff990, parent_tid=0x79ba835ff990, exit_signal=0,
↪  stack=0x79ba82dff000, stack_size=0x7fff80, tls=0x79ba835ff6c0} =>
↪  {parent_tid=[214448]}, 88) = 214448
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
futex(0x79ba81dfc990, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 214445, NULL,
↪  FUTEX_BITSET_MATCH_ANY) = 0
futex(0x79ba825fd990, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 214446, NULL,
↪  FUTEX_BITSET_MATCH_ANY) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba835ff990, parent_tid=0x79ba835ff990, exit_signal=0,
↪  stack=0x79ba82dff000, stack_size=0x7fff80, tls=0x79ba835ff6c0} =>
↪  {parent_tid=[214449]}, 88) = 214449
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba82dfe990, parent_tid=0x79ba82dfe990, exit_signal=0,
↪  stack=0x79ba825fe000, stack_size=0x7fff80, tls=0x79ba82dfe6c0} =>
↪  {parent_tid=[214450]}, 88) = 214450
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba825fd990, parent_tid=0x79ba825fd990, exit_signal=0,
↪  stack=0x79ba81dfd000, stack_size=0x7fff80, tls=0x79ba825fd6c0} =>
↪  {parent_tid=[214451]}, 88) = 214451
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba81dfc990, parent_tid=0x79ba81dfc990, exit_signal=0,
↪  stack=0x79ba815fc000, stack_size=0x7fff80, tls=0x79ba81dfc6c0} =>
↪  {parent_tid=[214452]}, 88) = 214452
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
futex(0x79ba835ff990, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 214449, NULL,
↪  FUTEX_BITSET_MATCH_ANY) = 0
futex(0x79ba81dfc990, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 214452, NULL,
↪  FUTEX_BITSET_MATCH_ANY) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba81dfc990, parent_tid=0x79ba81dfc990, exit_signal=0,
↪  stack=0x79ba815fc000, stack_size=0x7fff80, tls=0x79ba81dfc6c0} =>
↪  {parent_tid=[214453]}, 88) = 214453
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
```

```
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV⌋
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba825fd990, parent_tid=0x79ba825fd990, exit_signal=0,
↪  stack=0x79ba81dfd000, stack_size=0x7fff80, tls=0x79ba825fd6c0} =>
↪  {parent_tid=[214454]}, 88) = 214454
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV⌋
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba82dfe990, parent_tid=0x79ba82dfe990, exit_signal=0,
↪  stack=0x79ba825fe000, stack_size=0x7fff80, tls=0x79ba82dfe6c0} =>
↪  {parent_tid=[214455]}, 88) = 214455
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV⌋
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba835ff990, parent_tid=0x79ba835ff990, exit_signal=0,
↪  stack=0x79ba82dff000, stack_size=0x7fff80, tls=0x79ba835ff6c0} =>
↪  {parent_tid=[214456]}, 88) = 214456
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
futex(0x79ba81dfc990, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 214453, NULL,
↪  FUTEX_BITSET_MATCH_ANY) = 0
futex(0x79ba82dfe990, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 214455, NULL,
↪  FUTEX_BITSET_MATCH_ANY) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV⌋
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba835ff990, parent_tid=0x79ba835ff990, exit_signal=0,
↪  stack=0x79ba82dff000, stack_size=0x7fff80, tls=0x79ba835ff6c0} =>
↪  {parent_tid=[214457]}, 88) = 214457
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV⌋
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba82dfe990, parent_tid=0x79ba82dfe990, exit_signal=0,
↪  stack=0x79ba825fe000, stack_size=0x7fff80, tls=0x79ba82dfe6c0} =>
↪  {parent_tid=[214458]}, 88) = 214458
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV⌋
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba825fd990, parent_tid=0x79ba825fd990, exit_signal=0,
↪  stack=0x79ba81dfd000, stack_size=0x7fff80, tls=0x79ba825fd6c0} =>
↪  {parent_tid=[214459]}, 88) = 214459
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV⌋
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba81dfc990, parent_tid=0x79ba81dfc990, exit_signal=0,
↪  stack=0x79ba815fc000, stack_size=0x7fff80, tls=0x79ba81dfc6c0} =>
↪  {parent_tid=[214460]}, 88) = 214460
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
futex(0x79ba835ff990, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 214457, NULL,
↪  FUTEX_BITSET_MATCH_ANY) = 0
```

```
futex(0x79ba82dfe990, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 214458, NULL,
↪  FUTEX_BITSET_MATCH_ANY) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV⌋
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba81dfc990, parent_tid=0x79ba81dfc990, exit_signal=0,
↪  stack=0x79ba815fc000, stack_size=0x7fff80, tls=0x79ba81dfc6c0} =>
↪  {parent_tid=[214461]}, 88) = 214461
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV⌋
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba825fd990, parent_tid=0x79ba825fd990, exit_signal=0,
↪  stack=0x79ba81dfd000, stack_size=0x7fff80, tls=0x79ba825fd6c0} =>
↪  {parent_tid=[214462]}, 88) = 214462
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV⌋
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba82dfe990, parent_tid=0x79ba82dfe990, exit_signal=0,
↪  stack=0x79ba825fe000, stack_size=0x7fff80, tls=0x79ba82dfe6c0} =>
↪  {parent_tid=[214463]}, 88) = 214463
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV⌋
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba835ff990, parent_tid=0x79ba835ff990, exit_signal=0,
↪  stack=0x79ba82dff000, stack_size=0x7fff80, tls=0x79ba835ff6c0} =>
↪  {parent_tid=[214464]}, 88) = 214464
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
futex(0x79ba81dfc990, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 214461, NULL,
↪  FUTEX_BITSET_MATCH_ANY) = 0
futex(0x79ba825fd990, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 214462, NULL,
↪  FUTEX_BITSET_MATCH_ANY) = 0
futex(0x79ba82dfe990, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 214463, NULL,
↪  FUTEX_BITSET_MATCH_ANY) = 0
futex(0x79ba835ff990, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 214464, NULL,
↪  FUTEX_BITSET_MATCH_ANY) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV⌋
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba835ff990, parent_tid=0x79ba835ff990, exit_signal=0,
↪  stack=0x79ba82dff000, stack_size=0x7fff80, tls=0x79ba835ff6c0} =>
↪  {parent_tid=[214475]}, 88) = 214475
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV⌋
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba82dfe990, parent_tid=0x79ba82dfe990, exit_signal=0,
↪  stack=0x79ba825fe000, stack_size=0x7fff80, tls=0x79ba82dfe6c0} =>
↪  {parent_tid=[214476]}, 88) = 214476
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
```

```
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV␣
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba825fd990, parent_tid=0x79ba825fd990, exit_signal=0,
↪  stack=0x79ba81dfd000, stack_size=0x7fff80, tls=0x79ba825fd6c0} =>
↪  {parent_tid=[214477]}, 88) = 214477
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV␣
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba81dfc990, parent_tid=0x79ba81dfc990, exit_signal=0,
↪  stack=0x79ba815fc000, stack_size=0x7fff80, tls=0x79ba81dfc6c0} =>
↪  {parent_tid=[214478]}, 88) = 214478
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
futex(0x79ba835ff990, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 214475, NULL,
↪  FUTEX_BITSET_MATCH_ANY) = 0
futex(0x79ba82dfe990, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 214476, NULL,
↪  FUTEX_BITSET_MATCH_ANY) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV␣
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba81dfc990, parent_tid=0x79ba81dfc990, exit_signal=0,
↪  stack=0x79ba815fc000, stack_size=0x7fff80, tls=0x79ba81dfc6c0} =>
↪  {parent_tid=[214479]}, 88) = 214479
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV␣
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba825fd990, parent_tid=0x79ba825fd990, exit_signal=0,
↪  stack=0x79ba81dfd000, stack_size=0x7fff80, tls=0x79ba825fd6c0} =>
↪  {parent_tid=[214480]}, 88) = 214480
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV␣
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba82dfe990, parent_tid=0x79ba82dfe990, exit_signal=0,
↪  stack=0x79ba825fe000, stack_size=0x7fff80, tls=0x79ba82dfe6c0} =>
↪  {parent_tid=[214481]}, 88) = 214481
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV␣
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba835ff990, parent_tid=0x79ba835ff990, exit_signal=0,
↪  stack=0x79ba82dff000, stack_size=0x7fff80, tls=0x79ba835ff6c0} =>
↪  {parent_tid=[214482]}, 88) = 214482
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
futex(0x79ba81dfc990, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 214479, NULL,
↪  FUTEX_BITSET_MATCH_ANY) = 0
futex(0x79ba825fd990, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 214480, NULL,
↪  FUTEX_BITSET_MATCH_ANY) = 0
futex(0x79ba82dfe990, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 214481, NULL,
↪  FUTEX_BITSET_MATCH_ANY) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
```

```
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV␣
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba835ff990, parent_tid=0x79ba835ff990, exit_signal=0,
↪  stack=0x79ba82dff000, stack_size=0x7fff80, tls=0x79ba835ff6c0} =>
↪  {parent_tid=[214483]}, 88) = 214483
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV␣
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba82dfe990, parent_tid=0x79ba82dfe990, exit_signal=0,
↪  stack=0x79ba825fe000, stack_size=0x7fff80, tls=0x79ba82dfe6c0} =>
↪  {parent_tid=[214484]}, 88) = 214484
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV␣
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba825fd990, parent_tid=0x79ba825fd990, exit_signal=0,
↪  stack=0x79ba81dfd000, stack_size=0x7fff80, tls=0x79ba825fd6c0} =>
↪  {parent_tid=[214485]}, 88) = 214485
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV␣
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba81dfc990, parent_tid=0x79ba81dfc990, exit_signal=0,
↪  stack=0x79ba815fc000, stack_size=0x7fff80, tls=0x79ba81dfc6c0} =>
↪  {parent_tid=[214486]}, 88) = 214486
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
futex(0x79ba835ff990, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 214483, NULL,
↪  FUTEX_BITSET_MATCH_ANY) = 0
futex(0x79ba82dfe990, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 214484, NULL,
↪  FUTEX_BITSET_MATCH_ANY) = 0
futex(0x79ba825fd990, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 214485, NULL,
↪  FUTEX_BITSET_MATCH_ANY) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV␣
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba81dfc990, parent_tid=0x79ba81dfc990, exit_signal=0,
↪  stack=0x79ba815fc000, stack_size=0x7fff80, tls=0x79ba81dfc6c0} =>
↪  {parent_tid=[214488]}, 88) = 214488
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV␣
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba825fd990, parent_tid=0x79ba825fd990, exit_signal=0,
↪  stack=0x79ba81dfd000, stack_size=0x7fff80, tls=0x79ba825fd6c0} =>
↪  {parent_tid=[214489]}, 88) = 214489
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV␣
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba82dfe990, parent_tid=0x79ba82dfe990, exit_signal=0,
↪  stack=0x79ba825fe000, stack_size=0x7fff80, tls=0x79ba82dfe6c0} =>
↪  {parent_tid=[214491]}, 88) = 214491
```

```
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV⌋
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba835ff990, parent_tid=0x79ba835ff990, exit_signal=0,
↪  stack=0x79ba82dff000, stack_size=0x7fff80, tls=0x79ba835ff6c0} =>
↪  {parent_tid=[214492]}, 88) = 214492
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
futex(0x79ba81dfc990, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 214488, NULL,
↪  FUTEX_BITSET_MATCH_ANY) = 0
futex(0x79ba825fd990, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 214489, NULL,
↪  FUTEX_BITSET_MATCH_ANY) = 0
futex(0x79ba82dfe990, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 214491, NULL,
↪  FUTEX_BITSET_MATCH_ANY) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV⌋
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba835ff990, parent_tid=0x79ba835ff990, exit_signal=0,
↪  stack=0x79ba82dff000, stack_size=0x7fff80, tls=0x79ba835ff6c0} =>
↪  {parent_tid=[214504]}, 88) = 214504
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV⌋
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba82dfe990, parent_tid=0x79ba82dfe990, exit_signal=0,
↪  stack=0x79ba825fe000, stack_size=0x7fff80, tls=0x79ba82dfe6c0} =>
↪  {parent_tid=[214505]}, 88) = 214505
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV⌋
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba825fd990, parent_tid=0x79ba825fd990, exit_signal=0,
↪  stack=0x79ba81dfd000, stack_size=0x7fff80, tls=0x79ba825fd6c0} =>
↪  {parent_tid=[214506]}, 88) = 214506
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV⌋
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba81dfc990, parent_tid=0x79ba81dfc990, exit_signal=0,
↪  stack=0x79ba815fc000, stack_size=0x7fff80, tls=0x79ba81dfc6c0} =>
↪  {parent_tid=[214507]}, 88) = 214507
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
futex(0x79ba835ff990, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 214504, NULL,
↪  FUTEX_BITSET_MATCH_ANY) = 0
futex(0x79ba82dfe990, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 214505, NULL,
↪  FUTEX_BITSET_MATCH_ANY) = 0
futex(0x79ba825fd990, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 214506, NULL,
↪  FUTEX_BITSET_MATCH_ANY) = 0
futex(0x79ba81dfc990, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 214507, NULL,
↪  FUTEX_BITSET_MATCH_ANY) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
```

```
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba81dfc990, parent_tid=0x79ba81dfc990, exit_signal=0,
↪  stack=0x79ba815fc000, stack_size=0x7fff80, tls=0x79ba81dfc6c0} =>
↪  {parent_tid=[214508]}, 88) = 214508
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba825fd990, parent_tid=0x79ba825fd990, exit_signal=0,
↪  stack=0x79ba81dfd000, stack_size=0x7fff80, tls=0x79ba825fd6c0} =>
↪  {parent_tid=[214509]}, 88) = 214509
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba82dfe990, parent_tid=0x79ba82dfe990, exit_signal=0,
↪  stack=0x79ba825fe000, stack_size=0x7fff80, tls=0x79ba82dfe6c0} =>
↪  {parent_tid=[214510]}, 88) = 214510
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba835ff990, parent_tid=0x79ba835ff990, exit_signal=0,
↪  stack=0x79ba82dff000, stack_size=0x7fff80, tls=0x79ba835ff6c0} =>
↪  {parent_tid=[214511]}, 88) = 214511
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
futex(0x79ba81dfc990, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 214508, NULL,
↪  FUTEX_BITSET_MATCH_ANY) = 0
futex(0x79ba825fd990, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 214509, NULL,
↪  FUTEX_BITSET_MATCH_ANY) = 0
futex(0x79ba835ff990, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 214511, NULL,
↪  FUTEX_BITSET_MATCH_ANY) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba835ff990, parent_tid=0x79ba835ff990, exit_signal=0,
↪  stack=0x79ba82dff000, stack_size=0x7fff80, tls=0x79ba835ff6c0} =>
↪  {parent_tid=[214512]}, 88) = 214512
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba82dfe990, parent_tid=0x79ba82dfe990, exit_signal=0,
↪  stack=0x79ba825fe000, stack_size=0x7fff80, tls=0x79ba82dfe6c0} =>
↪  {parent_tid=[214513]}, 88) = 214513
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba825fd990, parent_tid=0x79ba825fd990, exit_signal=0,
↪  stack=0x79ba81dfd000, stack_size=0x7fff80, tls=0x79ba825fd6c0} =>
↪  {parent_tid=[214514]}, 88) = 214514
```

```
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba81dfc990, parent_tid=0x79ba81dfc990, exit_signal=0,
↪  stack=0x79ba815fc000, stack_size=0x7fff80, tls=0x79ba81dfc6c0} =>
↪  {parent_tid=[214515]}, 88) = 214515
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
futex(0x79ba835ff990, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 214512, NULL,
↪  FUTEX_BITSET_MATCH_ANY) = 0
futex(0x79ba82dfe990, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 214513, NULL,
↪  FUTEX_BITSET_MATCH_ANY) = 0
futex(0x79ba81dfc990, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 214515, NULL,
↪  FUTEX_BITSET_MATCH_ANY) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba81dfc990, parent_tid=0x79ba81dfc990, exit_signal=0,
↪  stack=0x79ba815fc000, stack_size=0x7fff80, tls=0x79ba81dfc6c0} =>
↪  {parent_tid=[214516]}, 88) = 214516
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba825fd990, parent_tid=0x79ba825fd990, exit_signal=0,
↪  stack=0x79ba81dfd000, stack_size=0x7fff80, tls=0x79ba825fd6c0} =>
↪  {parent_tid=[214517]}, 88) = 214517
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba82dfe990, parent_tid=0x79ba82dfe990, exit_signal=0,
↪  stack=0x79ba825fe000, stack_size=0x7fff80, tls=0x79ba82dfe6c0} =>
↪  {parent_tid=[214518]}, 88) = 214518
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba835ff990, parent_tid=0x79ba835ff990, exit_signal=0,
↪  stack=0x79ba82dff000, stack_size=0x7fff80, tls=0x79ba835ff6c0} =>
↪  {parent_tid=[214519]}, 88) = 214519
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
futex(0x79ba81dfc990, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 214516, NULL,
↪  FUTEX_BITSET_MATCH_ANY) = 0
futex(0x79ba825fd990, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 214517, NULL,
↪  FUTEX_BITSET_MATCH_ANY) = 0
futex(0x79ba82dfe990, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 214518, NULL,
↪  FUTEX_BITSET_MATCH_ANY) = 0
futex(0x79ba835ff990, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 214519, NULL,
↪  FUTEX_BITSET_MATCH_ANY) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
```

```
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba835ff990, parent_tid=0x79ba835ff990, exit_signal=0,
↪  stack=0x79ba82dff000, stack_size=0x7fff80, tls=0x79ba835ff6c0} =>
↪  {parent_tid=[214520]}, 88) = 214520
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba82dfe990, parent_tid=0x79ba82dfe990, exit_signal=0,
↪  stack=0x79ba825fe000, stack_size=0x7fff80, tls=0x79ba82dfe6c0} =>
↪  {parent_tid=[214521]}, 88) = 214521
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba825fd990, parent_tid=0x79ba825fd990, exit_signal=0,
↪  stack=0x79ba81dfd000, stack_size=0x7fff80, tls=0x79ba825fd6c0} =>
↪  {parent_tid=[214522]}, 88) = 214522
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba81dfc990, parent_tid=0x79ba81dfc990, exit_signal=0,
↪  stack=0x79ba815fc000, stack_size=0x7fff80, tls=0x79ba81dfc6c0} =>
↪  {parent_tid=[214523]}, 88) = 214523
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
futex(0x79ba835ff990, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 214520, NULL,
↪  FUTEX_BITSET_MATCH_ANY) = 0
futex(0x79ba82dfe990, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 214521, NULL,
↪  FUTEX_BITSET_MATCH_ANY) = 0
futex(0x79ba825fd990, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 214522, NULL,
↪  FUTEX_BITSET_MATCH_ANY) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba81dfc990, parent_tid=0x79ba81dfc990, exit_signal=0,
↪  stack=0x79ba815fc000, stack_size=0x7fff80, tls=0x79ba81dfc6c0} =>
↪  {parent_tid=[214524]}, 88) = 214524
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba825fd990, parent_tid=0x79ba825fd990, exit_signal=0,
↪  stack=0x79ba81dfd000, stack_size=0x7fff80, tls=0x79ba825fd6c0} =>
↪  {parent_tid=[214525]}, 88) = 214525
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba82dfe990, parent_tid=0x79ba82dfe990, exit_signal=0,
↪  stack=0x79ba825fe000, stack_size=0x7fff80, tls=0x79ba82dfe6c0} =>
↪  {parent_tid=[214526]}, 88) = 214526
```

```
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSV
↪  SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
↪  child_tid=0x79ba835ff990, parent_tid=0x79ba835ff990, exit_signal=0,
↪  stack=0x79ba82dff000, stack_size=0x7fff80, tls=0x79ba835ff6c0} =>
↪  {parent_tid=[214527]}, 88) = 214527
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
futex(0x79ba81dfc990, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 214524, NULL,
↪  FUTEX_BITSET_MATCH_ANY) = 0
futex(0x79ba825fd990, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 214525, NULL,
↪  FUTEX_BITSET_MATCH_ANY) = 0
futex(0x79ba82dfe990, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 214526, NULL,
↪  FUTEX_BITSET_MATCH_ANY) = 0
write(1, "Algorithm ended on 24 iterations"..., 33) = 33
write(1, "Algorithm was done in: 1403 ms\n", 31) = 31
munmap(0x79ba83ca7000, 200704)          = 0
lseek(0, -1, SEEK_CUR)                  = -1 ESPIPE (Недопустимая операция
↪  смещения)
exit_group(0)                           = ?
+++ exited with 0 +++
```