

**МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)**

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

**Лабораторная работа №4
по курсу «Операционные системы»**

**Выполнил: Е. Г. Туймуков
Группа: М8О-208БВ-24
Преподаватель: Е. С. Миронов**

Москва, 2025

Условие

Требуется создать динамические библиотеки, которые реализуют заданный вариантом функционал. Далее использовать данные библиотеки 2-мя способами:

1. Во время компиляции (на этапе «линковки»/linking)
2. Во время исполнения программы. Библиотеки загружаются в память с помощью интерфейса ОС для работы с динамическими библиотеками

В конечном итоге, в лабораторной работе необходимо получить следующие части:

- Динамические библиотеки, реализующие контракты, которые заданы вариантом;
- Тестовая программа (программа №1), которая используют одну из библиотек, используя информацию полученные на этапе компиляции;
- Тестовая программа (программа №2), которая загружает библиотеки, используя только их относительные пути и контракты.

Провести анализ двух типов использования библиотек.

Пользовательский ввод для обоих программ должен быть организован следующим образом: Если пользователь вводит команду «0», то программа переключает одну реализацию контрактов на другую (необходимо только для программы №2). Можно реализовать лабораторную работу без данной функции, но максимальная оценка в этом случае будет «хорошо»;

1. «1 arg1 arg2 … argN», где после «1» идут аргументы для первой функции, предусмотренной контрактами. После ввода команды происходит вызов первой функции, и на экране появляется результат её выполнения;
2. «2 arg1 arg2 … argM», где после «2» идут аргументы для второй функции, предусмотренной контрактами. После ввода команды происходит вызов второй функции, и на экране появляется результат её выполнения.

Цель работы: Целью является приобретение практических навыков в: Создание динамических библиотек, Создание программ, которые используют функции динамических библиотек

Задание: Подсчёт наибольшего общего делителя для двух натуральных чисел(Int GCF(int A, int B)):

- Алгоритм Евклида
- Наивный алгоритм. Пытаться разделить числа на все числа, что меньше А и В.

Подсчет площади плоской геометрической фигуры по двум сторонам(Float Square(float A, float B)):

- Фигура прямоугольник
- Фигура прямоугольный треугольник

Вариант: 24

Метод решения

Программа демонстрирует работу с динамическими библиотеками двумя различными способами: привязкой на этапе компиляции и загрузкой во время выполнения. Реализованы две отдельные динамические библиотеки, содержащие разные варианты реализации одинаковых контрактов. Контракт включает две функции: вычисление наибольшего общего делителя двух чисел и вычисление площади геометрической фигуры. Первая библиотека использует алгоритм Евклида и считает площадь прямоугольника, вторая — наивный алгоритм поиска НОД и вычисляет площадь прямоугольного треугольника.

Ключевые компоненты:

* `contract.h` — определяет интерфейс (контракт), который обязаны реализовать все версии библиотек. * `libimpl1` и `libimpl2` — динамические библиотеки (`.dylib`), реализующие одинаковые функции разными способами. * `program1` — программа, которая линкуется с одной из библиотек на этапе компиляции. Все вызовы функций происходят через заранее известные символы, подгружаемые линковщиком. * `program2` — программа, загружающая библиотеки динамически при помощи системных вызовов `dlopen` и `dlsym`, что позволяет переключаться между реализациями библиотек прямо во время работы программы.

В первой программе работа с библиотекой происходит статически: компоновщик включается в процесс сборки и связывает вызовы функций с конкретной библиотекой. Эта схема проста, но не позволяет менять реализацию без перекомпиляции.

Вторая программа использует механизм динамической загрузки. Она не привязана к конкретной библиотеке заранее — вместо этого путь к библиотеке передаётся в `dlopen`, после чего указатели на функции извлекаются с помощью `dlsym`. Благодаря этому пользователь может в процессе выполнения вводить команду переключения (например, «`0`»), и программа будет закрывать текущую библиотеку и загружать другую. Это обеспечивает гибкость и расширяемость: можно подменять реализацию, добавлять новые библиотеки без изменения кода и без пересборки приложения.

На уровне ОС используются стандартные функции работы с динамическими загрузками:

* macOS/Linux: `dlopen`, `dlsym`, `dlclose` — для загрузки библиотек и доступа к символам, * механизм линковки компилятора — для статической привязки функций.

Такой подход демонстрирует различия между статическим и динамическим способом использования библиотек, показывает тонкости управления указателями на функции, необходимость проверок ошибок загрузки и выгрузки библиотек, а также подчёркивает роль контрактов как стабильного программного интерфейса, позволяющего подменять реализацию, не меняя вызывающий код.

Описание программы

Программа демонстрирует два способа использования динамических библиотек: привязку при компиляции и загрузку во время выполнения. Реализованы две библиотеки, каждая из которых содержит функции вычисления НОД и площади фигуры, но с разными алгоритмами. Первая использует алгоритм Евклида и считает площадь прямоугольника, вторая — наивный алгоритм и площадь прямоугольного треугольника. Основная программа при статической линковке вызывает функции напрямую из выбранной реализации, тогда как во втором варианте происходит загрузка библиотек через `dlopen`, получение адресов функций через `dlsym` и переключение между реализациями по команде пользователя. Таким образом обеспечивается возможность подмены функциональности без перекомпиляции программы и гибкое использование разных вариантов реализации через единый контракт.

Результаты

Разработанная программа демонстрирует работу с динамическими библиотеками двумя способами: через линковку во время компиляции и через загрузку во время выполнения. Основные результаты:

- Реализованы две независимые библиотеки с разными алгоритмами вычисления НОД и площади фигуры.
- Программа с компиляционной линковкой корректно вызывает функции выбранной реализации.
- Вторая программа успешно загружает библиотеки через `dlopen` и переключается между ними по команде пользователя.
- Обеспечена взаимозаменяемость реализаций благодаря единому контракту.
- Обработаны ошибки загрузки библиотек и получения символов, обеспечена стабильная работа при переключении.

Выводы

Ходе выполнения лабораторной работы я разработал две динамические библиотеки и программы, использующие их различными способами. Я понял разницу между статическим связыванием, когда реализация выбирается на этапе компиляции, и динамической загрузкой, позволяющей подменять функциональность прямо во время выполнения программы. Работа с `dlopen` и `dlsym` показала, как можно гибко управлять доступом к функциям библиотеки, не привязываясь к конкретной реализации заранее.

Создание двух вариантов библиотек с единым контрактом помогло увидеть ценность абстракции: программы остаются неизменными, даже если реализация функций меняется. Я также убедился в важности обработки ошибок при загрузке библиотек и получении символов, так как некорректная работа этих механизмов напрямую влияет на стабильность программы.

Кроме того, я освоил принципы организации кода при работе с динамическими библиотеками: разделение контрактов, реализаций и программ, а также использование CMake для сборки проекта. Это позволило выстроить понятную структуру и обеспечить удобство подключения разных библиотек.

В результате данной лабораторной работы я научился использовать динамические библиотеки в двух режимах, понял особенности их загрузки и связывания, а также получил практический опыт разработки модульной и расширяемой архитектуры, поддерживающей замену реализаций без изменения основного кода.

Исходная программа

```
#pragma once

extern "C" {
int GCF(int A,int B);
float Square(float A,float B);
}

#include "contract.h"
#include <cmath>

extern "C" int GCF(int A,int B) {
while (B != 0) {
int t = B;
B = A % B;
A = t;
}
return A;
}

extern "C" float Square(float A,float B) {
return A * B;
}

#include "contract.h"
#include <cmath>

extern "C" int GCF(int A,int B) {
int g = 1;
int m = std::fmin(A,B);
for (int i = 1; i <= m; i++) {
if (A % i == 0 && B % i == 0) g = i;
}
return g;
}

extern "C" float Square(float A,float B) {
return (A * B) / 2.0f;
}

#include <iostream>
#include "contract.h"
```

```
int main() {
std::cout <<"Program 1 (compile-time linking)" <<std::endl;
while (true) {
int cmd;
std::cin >>cmd;
if (!std::cin) break;

if (cmd == 1) {
int a,b;
std::cin >>a >>b;
std::cout <<GCF(a,b) <<std::endl;
}
else if (cmd == 2) {
float a,b;
std::cin >>a >>b;
std::cout <<Square(a,b) <<std::endl;
}
}
}
```

```
#include <iostream>
#include <dlfcn.h>
#include <string>

// указатели на функции
int (*GCF_func)(int,int) = nullptr;
float (*Square_func)(float,float) = nullptr;

void* handle = nullptr;

void loadLib(const char* path) {
if (handle) dlclose(handle);
handle = dlopen(path,RTLD_LAZY);

if (!handle) {
std::cerr <<"dlopen error: " <<derror() <<std::endl;
exit(1);
}

GCF_func = (int (*)(int,int)) dlsym(handle,"GCF");
```

```
Square_func = (float (*)(float,float)) dlsym(handle,"Square");
}

int main() {
std::cout <<"Program 2 (runtime dynamic loading)" <<std::endl;

// старгаем с первой либы
loadLib("./libimpl1.dylib");

while (true) {
int cmd;
std::cin >>cmd;
if (!std::cin) break;

if (cmd == 0) {
// переключаем
static bool toggle = false;
toggle = !toggle;
loadLib(toggle ? "./libimpl2.dylib" : "./libimpl1.dylib");
std::cout <<"Library switched" <<std::endl;
}
else if (cmd == 1) {
int a,b;
std::cin >>a >>b;
std::cout <<GCF_func(a,b) <<std::endl;
}
else if (cmd == 2) {
float a,b;
std::cin >>a >>b;
std::cout <<Square_func(a,b) <<std::endl;
}
}

if (handle) dlclose(handle);
}
```

Логи


```
10341 mmap(0x78d6e65fe000,8192,PROT_READ|PROT_WRITE,MAP_PRIVATE|MAP_FIXED|MAP_DENYWRIT
= 0x78d6e65fe000
10341 close(3) = 0
10341 mmap(NULL,8192,PROT_READ|PROT_WRITE,MAP_PRIVATE|MAP_ANONYMOUS,-1,0) =
0x78d6e6935000
10341 mmap(NULL,12288,PROT_READ|PROT_WRITE,MAP_PRIVATE|MAP_ANONYMOUS,-1,0)
= 0x78d6e6932000
10341 arch_prctl(ARCH_SET_FS,0x78d6e6932740) = 0
10341 set_tid_address(0x78d6e6932a10) = 10341
10341 set_robust_list(0x78d6e6932a20,24) = 0
10341 rseq(0x78d6e6933060,0x20,0,0x53053053) = 0
10341 mprotect(0x78d6e63ff000,16384,PROT_READ) = 0
10341 mprotect(0x78d6e65fe000,4096,PROT_READ) = 0
10341 mprotect(0x78d6e6963000,4096,PROT_READ) = 0
10341 mprotect(0x78d6e686c000,45056,PROT_READ) = 0
10341 mprotect(0x615728746000,4096,PROT_READ) = 0
10341 mprotect(0x78d6e69bb000,8192,PROT_READ) = 0
10341 prlimit64(0,RLIMIT_STACK,NULL,{rlim_cur=8192*1024,rlim_max=RLIM64_INFINITY})
= 0
10341 munmap(0x78d6e6965000,94467) = 0
10341 futex(0x78d6e687a7bc,FUTEX_WAKE_PRIVATE,2147483647) = 0
10341 getrandom("\x9e\x15\x a8\xd6\x30\x e3\x e4\x1c",8,GRND_NONBLOCK) = 8
10341 brk(NULL) = 0x615731163000
10341 brk(0x615731184000) = 0x615731184000
10341 fstat(1,{st_dev=makedev(0,0x1a),st_ino=3,st_mode=S_IFCHR|0620,st_nlink=1,st_uid
/* 2025-10-13T13:04:48+0300 */,st_atime_nsec=0,st_mtime=1760349888 /* 2025-10-13T13:0
*/,st_mtime_nsec=0,st_ctime=1760348474 /* 2025-10-13T12:41:14.222497184+0300
*/,st_ctime_nsec=222497184}) = 0
10341 write(1,"Before sorting:\n",16) = 16
10341 write(1,"746 358 274 817 780 93 905 419 265 147 \n",40) = 40
10341 rt_sigaction(SIGRT_1,{sa_handler=0x78d6e6299530,sa_mask=[],sa_flags=SA_RESTORER
= 0
10341 rt_sigprocmask(SIG_UNBLOCK,[RTMIN RT_1],NULL,8) = 0
10341 mmap(NULL,8392704,PROT_NONE,MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK,-1,0)
= 0x78d6e59ff000
10341 mprotect(0x78d6e5a00000,8388608,PROT_READ|PROT_WRITE) = 0
10341 rt_sigprocmask(SIG_BLOCK,~[],[],8) = 0
10341 clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SY
=>{parent_tid=[10342]},88) = 10342
10341 rt_sigprocmask(SIG_SETMASK,[],NULL,8) = 0
10342 rseq(0x78d6e61ffe0,0x20,0,0x53053053 <unfinished ...>
10341 futex(0x78d6e61ff990,FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME,10342,NULL,FUTEX_BI
<unfinished ...>
10342 <... rseq resumed> = 0
10342 set_robust_list(0x78d6e61ff9a0,24) = 0
10342 rt_sigprocmask(SIG_SETMASK,[],NULL,8) = 0
10342 write(1,"Thread started. Active threads: 1\n",34) = 34
10342 mmap(NULL,134217728,PROT_NONE,MAP_PRIVATE|MAP_ANONYMOUS,-1,0) = 0x78d6dd800000
```

```
10342 munmap(0x78d6dd800000,41943040) = 0
10342 munmap(0x78d6e4000000,25165824) = 0
10342 mprotect(0x78d6e0000000,135168,PROT_READ|PROT_WRITE) = 0
10342 mmap(NULL,8392704,PROT_NONE,MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK,-1,0)
= 0x78d6e51fe000
10342 mprotect(0x78d6e51ff000,8388608,PROT_READ|PROT_WRITE) = 0
10342 rt_sigprocmask(SIG_BLOCK,~[],[],8) = 0
10342 clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SY
=>{parent_tid=[10343]},88) = 10343
10343 rseq(0x78d6e59fefe0,0x20,0,0x53053053 <unfinished ...>
10342 rt_sigprocmask(SIG_SETMASK,[],<unfinished ...>
10343 <... rseq resumed>) = 0
10342 <... rt_sigprocmask resumed>NULL,8) = 0
10343 set_robust_list(0x78d6e59fe9a0,24 <unfinished ...>
10342 futex(0x78d6e59fe990,FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME,10343,NULL,FUTEX_BI
<unfinished ...>
10343 <... set_robust_list resumed>) = 0
10343 rt_sigprocmask(SIG_SETMASK,[],NULL,8) = 0
10343 write(1,"Thread started. Active threads: 2\n",34) = 34
10343 mmap(NULL,134217728,PROT_NONE,MAP_PRIVATE|MAP_ANONYMOUS,-1,0) = 0x78d6d8000000
10343 munmap(0x78d6dc000000,67108864) = 0
10343 mprotect(0x78d6d8000000,135168,PROT_READ|PROT_WRITE) = 0
10343 mmap(NULL,8392704,PROT_NONE,MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK,-1,0)
= 0x78d6e49fd000
10343 mprotect(0x78d6e49fe000,8388608,PROT_READ|PROT_WRITE) = 0
10343 rt_sigprocmask(SIG_BLOCK,~[],[],8) = 0
10343 clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SY
=>{parent_tid=[10344]},88) = 10344
10344 rseq(0x78d6e51fdf0,0x20,0,0x53053053 <unfinished ...>
10343 rt_sigprocmask(SIG_SETMASK,[],<unfinished ...>
10344 <... rseq resumed>) = 0
10343 <... rt_sigprocmask resumed>NULL,8) = 0
10344 set_robust_list(0x78d6e51fd9a0,24 <unfinished ...>
10343 futex(0x78d6e51fd990,FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME,10344,NULL,FUTEX_BI
<unfinished ...>
10344 <... set_robust_list resumed>) = 0
10344 rt_sigprocmask(SIG_SETMASK,[],NULL,8) = 0
10344 write(1,"Thread started. Active threads: 3\n",34) = 34
10344 mmap(0x78d6dc000000,67108864,PROT_NONE,MAP_PRIVATE|MAP_ANONYMOUS,-1,0)
= 0x78d6dc000000
10344 mprotect(0x78d6dc000000,135168,PROT_READ|PROT_WRITE) = 0
10344 mmap(NULL,8392704,PROT_NONE,MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK,-1,0)
= 0x78d6e41fc000
10344 mprotect(0x78d6e41fd000,8388608,PROT_READ|PROT_WRITE) = 0
10344 rt_sigprocmask(SIG_BLOCK,~[],[],8) = 0
10344 clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SY
=>{parent_tid=[10345]},88) = 10345
10345 rseq(0x78d6e49fcfe0,0x20,0,0x53053053 <unfinished ...>
```

```
10344 rt_sigprocmask(SIG_SETMASK, [] , <unfinished ...>
10345 <... rseq resumed>) = 0
10344 <... rt_sigprocmask resumed>NULL,8) = 0
10345 set_robust_list(0x78d6e49fc9a0,24 <unfinished ...>
10344 futex(0x78d6e49fc990,FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME,10345,NULL,FUTEX_BI
<unfinished ...>
10345 <... set_robust_list resumed>) = 0
10345 rt_sigprocmask(SIG_SETMASK, [] ,NULL,8) = 0
10345 write(1,"Thread started. Active threads: 4\n",34) = 34
10345 mmap(NULL,134217728,PROT_NONE,MAP_PRIVATE|MAP_ANONYMOUS,-1,0) = 0x78d6d0000000
10345 munmap(0x78d6d4000000,67108864) = 0
10345 mprotect(0x78d6d0000000,135168,PROT_READ|PROT_WRITE) = 0
10345 mmap(NULL,8392704,PROT_NONE,MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK,-1,0)
= 0x78d6d77ff000
10345 mprotect(0x78d6d7800000,8388608,PROT_READ|PROT_WRITE) = 0
10345 rt_sigprocmask(SIG_BLOCK,~[],[],8) = 0
10345 clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SY
=>{parent_tid=[10346]},88) = 10346
10346 rseq(0x78d6d7ffffe0,0x20,0,0x53053053 <unfinished ...>
10345 rt_sigprocmask(SIG_SETMASK, [] , <unfinished ...>
10346 <... rseq resumed>) = 0
10345 <... rt_sigprocmask resumed>NULL,8) = 0
10346 set_robust_list(0x78d6d7fff9a0,24 <unfinished ...>
10345 futex(0x78d6d7fff990,FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME,10346,NULL,FUTEX_BI
<unfinished ...>
10346 <... set_robust_list resumed>) = 0
10346 rt_sigprocmask(SIG_SETMASK, [] ,NULL,8) = 0
10346 write(1,"Thread started. Active threads: 5\n",34) = 34
10346 mmap(0x78d6d4000000,67108864,PROT_NONE,MAP_PRIVATE|MAP_ANONYMOUS,-1,0)
= 0x78d6cc000000
10346 mprotect(0x78d6cc000000,135168,PROT_READ|PROT_WRITE) = 0
10346 mmap(NULL,8392704,PROT_NONE,MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK,-1,0)
= 0x78d6d6ffe000
10346 mprotect(0x78d6d6fff000,8388608,PROT_READ|PROT_WRITE) = 0
10346 rt_sigprocmask(SIG_BLOCK,~[],[],8) = 0
10346 clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SY
<unfinished ...>
10347 rseq(0x78d6d77fef0,0x20,0,0x53053053 <unfinished ...>
10346 <... clone3 resumed>=>{parent_tid=[10347]},88) = 10347
10347 <... rseq resumed>) = 0
10346 rt_sigprocmask(SIG_SETMASK, [] , <unfinished ...>
10347 set_robust_list(0x78d6d77fe9a0,24 <unfinished ...>
10346 <... rt_sigprocmask resumed>NULL,8) = 0
10347 <... set_robust_list resumed>) = 0
10347 rt_sigprocmask(SIG_SETMASK, [] , <unfinished ...>
10346 futex(0x78d6d77fe990,FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME,10347,NULL,FUTEX_BI
<unfinished ...>
10347 <... rt_sigprocmask resumed>NULL,8) = 0
```

```
10347 write(1,"Thread started. Active threads: 6\n",34) = 34
10347 mmap(NULL,134217728,PROT_NONE,MAP_PRIVATE|MAP_ANONYMOUS,-1,0) = 0x78d6c4000000
10347 munmap(0x78d6c8000000,67108864) = 0
10347 mprotect(0x78d6c4000000,135168,PROT_READ|PROT_WRITE) = 0
10347 mmap(NULL,8392704,PROT_NONE,MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK,-1,0)
= 0x78d6d67fd000
10347 mprotect(0x78d6d67fe000,8388608,PROT_READ|PROT_WRITE) = 0
10347 rt_sigprocmask(SIG_BLOCK,~[],[],8) = 0
10347 clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SY
=>{parent_tid=[10348]},88) = 10348
10348 rseq(0x78d6d6ffdfe0,0x20,0,0x53053053 <unfinished ...>
10347 rt_sigprocmask(SIG_SETMASK,[],<unfinished ...>
10348 <... rseq resumed>) = 0
10347 <... rt_sigprocmask resumed>NULL,8) = 0
10348 set_robust_list(0x78d6d6ffd9a0,24 <unfinished ...>
10347 futex(0x78d6d6ffd990,FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME,10348,NULL,FUTEX_BI
<unfinished ...>
10348 <... set_robust_list resumed>) = 0
10348 rt_sigprocmask(SIG_SETMASK,[],NULL,8) = 0
10348 write(1,"Thread started. Active threads: 7\n",34) = 34
10348 mmap(0x78d6c8000000,67108864,PROT_NONE,MAP_PRIVATE|MAP_ANONYMOUS,-1,0)
= 0x78d6c8000000
10348 mprotect(0x78d6c8000000,135168,PROT_READ|PROT_WRITE) = 0
10348 mmap(NULL,8392704,PROT_NONE,MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK,-1,0)
= 0x78d6d5ffc000
10348 mprotect(0x78d6d5ffd000,8388608,PROT_READ|PROT_WRITE) = 0
10348 rt_sigprocmask(SIG_BLOCK,~[],[],8) = 0
10348 clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SY
<unfinished ...>
10349 rseq(0x78d6d67fcfe0,0x20,0,0x53053053 <unfinished ...>
10348 <... clone3 resumed>=>{parent_tid=[10349]},88) = 10349
10349 <... rseq resumed>) = 0
10348 rt_sigprocmask(SIG_SETMASK,[],<unfinished ...>
10349 set_robust_list(0x78d6d67fc9a0,24 <unfinished ...>
10348 <... rt_sigprocmask resumed>NULL,8) = 0
10349 <... set_robust_list resumed>) = 0
10348 futex(0x78d6d67fc990,FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME,10349,NULL,FUTEX_BI
<unfinished ...>
10349 rt_sigprocmask(SIG_SETMASK,[],NULL,8) = 0
10349 write(1,"Thread started. Active threads: 8\n",34) = 34
10349 mmap(NULL,134217728,PROT_NONE,MAP_PRIVATE|MAP_ANONYMOUS,-1,0) = 0x78d6bc000000
10349 munmap(0x78d6c0000000,67108864) = 0
10349 mprotect(0x78d6bc000000,135168,PROT_READ|PROT_WRITE) = 0
10349 mmap(NULL,8392704,PROT_NONE,MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK,-1,0)
= 0x78d6d57fb000
10349 mprotect(0x78d6d57fc000,8388608,PROT_READ|PROT_WRITE) = 0
10349 rt_sigprocmask(SIG_BLOCK,~[],[],8) = 0
10349 clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SY
```

```
<unfinished ...>
10350 rseq(0x78d6d5ffbf0,0x20,0,0x53053053 <unfinished ...>
10349 <... clone3 resumed=>=>{parent_tid=[10350]},88) = 10350
10350 <... rseq resumed> = 0
10349 rt_sigprocmask(SIG_SETMASK, [], <unfinished ...>
10350 set_robust_list(0x78d6d5ffbf9a0,24) = 0
10349 <... rt_sigprocmask resumed>NULL,8) = 0
10350 rt_sigprocmask(SIG_SETMASK, [], <unfinished ...>
10349 futex(0x78d6d5ffbf990,FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME,10350,NULL,FUTEX_BI
<unfinished ...>
10350 <... rt_sigprocmask resumed>NULL,8) = 0
10350 write(1,"Thread started. Active threads: 9\n",34) = 34
10350 openat(AT_FDCWD,"/sys/devices/system/cpu/online",O_RDONLY|O_CLOEXEC)
= 3
10350 read(3,"0-19\n",1024) = 5
10350 close(3) = 0
10350 mmap(0x78d6c0000000,67108864,PROT_NONE,MAP_PRIVATE|MAP_ANONYMOUS,-1,0)
= 0x78d6c0000000
10350 mprotect(0x78d6c0000000,135168,PROT_READ|PROT_WRITE) = 0
10350 mmap(NULL,8392704,PROT_NONE,MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK,-1,0)
= 0x78d6d4ffa000
10350 mprotect(0x78d6d4ffb000,8388608,PROT_READ|PROT_WRITE) = 0
10350 rt_sigprocmask(SIG_BLOCK,~[],[],8) = 0
10350 clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SY
<unfinished ...>
10351 rseq(0x78d6d57faf0,0x20,0,0x53053053 <unfinished ...>
10350 <... clone3 resumed=>=>{parent_tid=[10351]},88) = 10351
10351 <... rseq resumed> = 0
10350 rt_sigprocmask(SIG_SETMASK, [], <unfinished ...>
10351 set_robust_list(0x78d6d57fa9a0,24 <unfinished ...>
10350 <... rt_sigprocmask resumed>NULL,8) = 0
10351 <... set_robust_list resumed> = 0
10350 futex(0x78d6d57fa990,FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME,10351,NULL,FUTEX_BI
<unfinished ...>
10351 rt_sigprocmask(SIG_SETMASK, [],NULL,8) = 0
10351 write(1,"Thread started. Active threads: 10\n",35) = 35
10351 mmap(NULL,134217728,PROT_NONE,MAP_PRIVATE|MAP_ANONYMOUS,-1,0) = 0x78d6b4000000
10351 munmap(0x78d6b8000000,67108864) = 0
10351 mprotect(0x78d6b4000000,135168,PROT_READ|PROT_WRITE) = 0
10351 mmap(NULL,8392704,PROT_NONE,MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK,-1,0)
= 0x78d6d47f9000
10351 mprotect(0x78d6d47fa000,8388608,PROT_READ|PROT_WRITE) = 0
10351 rt_sigprocmask(SIG_BLOCK,~[],[],8) = 0
10351 clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SY
=>{parent_tid=[10352]},88) = 10352
10352 rseq(0x78d6d4ff9fe0,0x20,0,0x53053053 <unfinished ...>
10351 rt_sigprocmask(SIG_SETMASK, [], <unfinished ...>
10352 <... rseq resumed> = 0
```

```
10351 <... rt_sigprocmask resumed>NULL,8) = 0
10352 set_robust_list(0x78d6d4ff99a0,24 <unfinished ...>
10351 futex(0x78d6d4ff9990,FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME,10352,NULL,FUTEX_BI
<unfinished ...>
10352 <... set_robust_list resumed>)      = 0
10352 rt_sigprocmask(SIG_SETMASK,[],NULL,8) = 0
10352 write(1,"Thread started. Active threads: 11\n",35) = 35
10352 mmap(0x78d6b8000000,67108864,PROT_NONE,MAP_PRIVATE|MAP_ANONYMOUS,-1,0)
= 0x78d6b8000000
10352 mprotect(0x78d6b8000000,135168,PROT_READ|PROT_WRITE) = 0
10352 mmap(NULL,8392704,PROT_NONE,MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK,-1,0)
= 0x78d6b37ff000
10352 mprotect(0x78d6b3800000,8388608,PROT_READ|PROT_WRITE) = 0
10352 rt_sigprocmask(SIG_BLOCK,~[],[],8) = 0
10352 clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SY
=>{parent_tid=[10353]},88) = 10353
10353 rseq(0x78d6b3ffffe0,0x20,0,0x53053053 <unfinished ...>
10352 rt_sigprocmask(SIG_SETMASK,[], <unfinished ...>
10353 <... rseq resumed>)            = 0
10352 <... rt_sigprocmask resumed>NULL,8) = 0
10353 set_robust_list(0x78d6b3fff9a0,24 <unfinished ...>
10352 futex(0x78d6b3fff990,FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME,10353,NULL,FUTEX_BI
<unfinished ...>
10353 <... set_robust_list resumed>)      = 0
10353 rt_sigprocmask(SIG_SETMASK,[],NULL,8) = 0
10353 write(1,"Thread started. Active threads: 12\n",35) = 35
10353 mmap(NULL,134217728,PROT_NONE,MAP_PRIVATE|MAP_ANONYMOUS,-1,0) = 0x78d6ab600000
10353 munmap(0x78d6ab600000,10485760)   = 0
10353 munmap(0x78d6b0000000,56623104)   = 0
10353 mprotect(0x78d6ac000000,135168,PROT_READ|PROT_WRITE) = 0
10353 mmap(NULL,8392704,PROT_NONE,MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK,-1,0)
= 0x78d6b2ffe000
10353 mprotect(0x78d6b2ffff000,8388608,PROT_READ|PROT_WRITE) = 0
10353 rt_sigprocmask(SIG_BLOCK,~[],[],8) = 0
10353 clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SY
=>{parent_tid=[10354]},88) = 10354
10354 rseq(0x78d6b37fe0,0x20,0,0x53053053 <unfinished ...>
10353 rt_sigprocmask(SIG_SETMASK,[], <unfinished ...>
10354 <... rseq resumed>)            = 0
10353 <... rt_sigprocmask resumed>NULL,8) = 0
10354 set_robust_list(0x78d6b37fe9a0,24 <unfinished ...>
10353 futex(0x78d6b37fe990,FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME,10354,NULL,FUTEX_BI
<unfinished ...>
10354 <... set_robust_list resumed>)      = 0
10354 rt_sigprocmask(SIG_SETMASK,[],NULL,8) = 0
10354 write(1,"Thread started. Active threads: 13\n",35) = 35
10354 mmap(NULL,134217728,PROT_NONE,MAP_PRIVATE|MAP_ANONYMOUS,-1,0) = 0x78d6a4000000
10354 munmap(0x78d6a8000000,67108864)   = 0
```

```
10354 mprotect(0x78d6a4000000,135168,PROT_READ|PROT_WRITE) = 0
10354 mmap(NULL,8392704,PROT_NONE,MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK,-1,0)
= 0x78d6b27fd000
10354 mprotect(0x78d6b27fe000,8388608,PROT_READ|PROT_WRITE) = 0
10354 rt_sigprocmask(SIG_BLOCK,~[],[],8) = 0
10354 clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SY
<unfinished ...>
10355 rseq(0x78d6b2ffdfe0,0x20,0,0x53053053 <unfinished ...>
10354 <... clone3 resumed=>={parent_tid=[10355]},88) = 10355
10355 <... rseq resumed>) = 0
10354 rt_sigprocmask(SIG_SETMASK,[],<unfinished ...>
10355 set_robust_list(0x78d6b2ffd9a0,24 <unfinished ...>
10354 <... rt_sigprocmask resumed>NULL,8) = 0
10355 <... set_robust_list resumed>) = 0
10354 futex(0x78d6b2ffd990,FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME,10355,NULL,FUTEX_BI
<unfinished ...>
10355 rt_sigprocmask(SIG_SETMASK,[],NULL,8) = 0
10355 write(1,"Thread started. Active threads: 14\n",35) = 35
10355 mmap(0x78d6a8000000,67108864,PROT_NONE,MAP_PRIVATE|MAP_ANONYMOUS,-1,0)
= 0x78d6a8000000
10355 mprotect(0x78d6a8000000,135168,PROT_READ|PROT_WRITE) = 0
10355 mmap(NULL,8392704,PROT_NONE,MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK,-1,0)
= 0x78d6b1ffc000
10355 mprotect(0x78d6b1ffd000,8388608,PROT_READ|PROT_WRITE) = 0
10355 rt_sigprocmask(SIG_BLOCK,~[],[],8) = 0
10355 clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SY
<unfinished ...>
10356 rseq(0x78d6b27fcfe0,0x20,0,0x53053053 <unfinished ...>
10355 <... clone3 resumed=>={parent_tid=[10356]},88) = 10356
10356 <... rseq resumed>) = 0
10355 rt_sigprocmask(SIG_SETMASK,[],<unfinished ...>
10356 set_robust_list(0x78d6b27fc9a0,24 <unfinished ...>
10355 <... rt_sigprocmask resumed>NULL,8) = 0
10356 <... set_robust_list resumed>) = 0
10355 futex(0x78d6b27fc990,FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME,10356,NULL,FUTEX_BI
<unfinished ...>
10356 rt_sigprocmask(SIG_SETMASK,[],NULL,8) = 0
10356 write(1,"Thread started. Active threads: 15\n",35) = 35
10356 mmap(NULL,134217728,PROT_NONE,MAP_PRIVATE|MAP_ANONYMOUS,-1,0) = 0x78d69c000000
10356 munmap(0x78d6a0000000,67108864) = 0
10356 mprotect(0x78d69c000000,135168,PROT_READ|PROT_WRITE) = 0
10356 mmap(NULL,8392704,PROT_NONE,MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK,-1,0)
= 0x78d6b17fb000
10356 mprotect(0x78d6b17fc000,8388608,PROT_READ|PROT_WRITE) = 0
10356 rt_sigprocmask(SIG_BLOCK,~[],[],8) = 0
10356 clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SY
=>{parent_tid=[10357]},88) = 10357
10357 rseq(0x78d6b1ffbfe0,0x20,0,0x53053053 <unfinished ...>
```

```
10356 rt_sigprocmask(SIG_SETMASK, [] , <unfinished ...>
10357 <... rseq resumed>) = 0
10356 <... rt_sigprocmask resumed>NULL,8) = 0
10357 set_robust_list(0x78d6b1ffb9a0,24 <unfinished ...>
10356 futex(0x78d6b1ffb990,FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME,10357,NULL,FUTEX_BI
<unfinished ...>
10357 <... set_robust_list resumed>) = 0
10357 rt_sigprocmask(SIG_SETMASK, [] ,NULL,8) = 0
10357 write(1,"Thread started. Active threads: 16\n",35) = 35
10357 write(1,"Thread finished. Active threads: 16\n",36) = 36
10357 mmap(0x78d6a0000000,67108864,PROT_NONE,MAP_PRIVATE|MAP_ANONYMOUS,-1,0)
= 0x78d6a0000000
10357 mprotect(0x78d6a0000000,135168,PROT_READ|PROT_WRITE) = 0
10357 rt_sigprocmask(SIG_BLOCK,~[RT_1],NULL,8) = 0
10357 madvise(0x78d6b17fb000,8368128,MADV_DONTNEED) = 0
10357 exit(0) = ?
10357 +++ exited with 0 ===
10356 <... futex resumed> = 0
10356 write(1,"Thread finished. Active threads: 15\n",36) = 36
10356 rt_sigprocmask(SIG_BLOCK,~[RT_1],NULL,8) = 0
10356 madvise(0x78d6b1ffc000,8368128,MADV_DONTNEED) = 0
10356 exit(0) = ?
10356 +++ exited with 0 ===
10355 <... futex resumed> = 0
10355 write(1,"Thread finished. Active threads: 14\n",36) = 36
10355 rt_sigprocmask(SIG_BLOCK,~[RT_1],NULL,8) = 0
10355 madvise(0x78d6b27fd000,8368128,MADV_DONTNEED) = 0
10355 exit(0) = ?
10355 +++ exited with 0 ===
10354 <... futex resumed> = 0
10354 write(1,"Thread finished. Active threads: 13\n",36) = 36
10354 rt_sigprocmask(SIG_BLOCK,~[RT_1],NULL,8) = 0
10354 madvise(0x78d6b2ffe000,8368128,MADV_DONTNEED) = 0
10354 exit(0) = ?
10354 +++ exited with 0 ===
10353 <... futex resumed> = 0
10353 write(1,"Thread finished. Active threads: 12\n",36) = 36
10353 rt_sigprocmask(SIG_BLOCK,~[RT_1],NULL,8) = 0
10353 madvise(0x78d6b37ff000,8368128,MADV_DONTNEED) = 0
10353 exit(0) = ?
10353 +++ exited with 0 ===
10352 <... futex resumed> = 0
10352 munmap(0x78d6b17fb000,8392704) = 0
10352 write(1,"Thread finished. Active threads: 11\n",36) = 36
10352 rt_sigprocmask(SIG_BLOCK,~[RT_1],NULL,8) = 0
10352 madvise(0x78d6d47f9000,8368128,MADV_DONTNEED) = 0
10352 exit(0) = ?
10352 +++ exited with 0 ===
```

```
10351 <... futex resumed>          = 0
10351 munmap(0x78d6b1ffc000,8392704) = 0
10351 write(1,"Thread finished. Active threads: 10\n",36) = 36
10351 rt_sigprocmask(SIG_BLOCK,~[RT_1],NULL,8) = 0
10351 madvise(0x78d6d4ffa000,8368128,MADV_DONTNEED) = 0
10351 exit(0)                      = ?
10351 +++ exited with 0 ===
10350 <... futex resumed>          = 0
10350 munmap(0x78d6b27fd000,8392704) = 0
10350 write(1,"Thread finished. Active threads: 9\n",35) = 35
10350 rt_sigprocmask(SIG_BLOCK,~[RT_1],NULL,8) = 0
10350 madvise(0x78d6d57fb000,8368128,MADV_DONTNEED) = 0
10350 exit(0)                      = ?
10350 +++ exited with 0 ===
10349 <... futex resumed>          = 0
10349 munmap(0x78d6b2ffe000,8392704) = 0
10349 write(1,"Thread finished. Active threads: 8\n",35) = 35
10349 rt_sigprocmask(SIG_BLOCK,~[RT_1],NULL,8) = 0
10349 madvise(0x78d6d5ffc000,8368128,MADV_DONTNEED) = 0
10349 exit(0)                      = ?
10349 +++ exited with 0 ===
10348 <... futex resumed>          = 0
10348 munmap(0x78d6b37ff000,8392704) = 0
10348 write(1,"Thread finished. Active threads: 7\n",35) = 35
10348 rt_sigprocmask(SIG_BLOCK,~[RT_1],NULL,8) = 0
10348 madvise(0x78d6d67fd000,8368128,MADV_DONTNEED) = 0
10348 exit(0)                      = ?
10348 +++ exited with 0 ===
10347 <... futex resumed>          = 0
10347 munmap(0x78d6d47f9000,8392704) = 0
10347 write(1,"Thread finished. Active threads: 6\n",35) = 35
10347 rt_sigprocmask(SIG_BLOCK,~[RT_1],NULL,8) = 0
10347 madvise(0x78d6d6ffe000,8368128,MADV_DONTNEED) = 0
10347 exit(0)                      = ?
10347 +++ exited with 0 ===
10346 <... futex resumed>          = 0
10346 munmap(0x78d6d4ffa000,8392704) = 0
10346 write(1,"Thread finished. Active threads: 5\n",35) = 35
10346 rt_sigprocmask(SIG_BLOCK,~[RT_1],NULL,8) = 0
10346 madvise(0x78d6d77ff000,8368128,MADV_DONTNEED) = 0
10346 exit(0)                      = ?
10346 +++ exited with 0 ===
10345 <... futex resumed>          = 0
10345 munmap(0x78d6d57fb000,8392704) = 0
10345 write(1,"Thread finished. Active threads: 4\n",35) = 35
10345 rt_sigprocmask(SIG_BLOCK,~[RT_1],NULL,8) = 0
10345 madvise(0x78d6e41fc000,8368128,MADV_DONTNEED) = 0
10345 exit(0)                      = ?
```

```
10345 +++ exited with 0 +++
10344 <... futex resumed> = 0
10344 munmap(0x78d6d5ffc000,8392704) = 0
10344 write(1,"Thread finished. Active threads: 3\n",35) = 35
10344 rt_sigprocmask(SIG_BLOCK,~[RT_1],NULL,8) = 0
10344 madvise(0x78d6e49fd000,8368128,MADV_DONTNEED) = 0
10344 exit(0) = ?
10344 +++ exited with 0 +++
10343 <... futex resumed> = 0
10343 munmap(0x78d6d67fd000,8392704) = 0
10343 write(1,"Thread finished. Active threads: 2\n",35) = 35
10343 rt_sigprocmask(SIG_BLOCK,~[RT_1],NULL,8) = 0
10343 madvise(0x78d6e51fe000,8368128,MADV_DONTNEED) = 0
10343 exit(0) = ?
10343 +++ exited with 0 +++
10342 <... futex resumed> = 0
10342 munmap(0x78d6d6ffe000,8392704) = 0
10342 write(1,"Thread finished. Active threads: 1\n",35) = 35
10342 rt_sigprocmask(SIG_BLOCK,~[RT_1],NULL,8) = 0
10342 madvise(0x78d6e59ff000,8368128,MADV_DONTNEED) = 0
10342 exit(0) = ?
10341 <... futex resumed> = 0
10342 +++ exited with 0 +++
10341 munmap(0x78d6d77ff000,8392704) = 0
10341 write(1,"After sorting:\n",15) = 15
10341 write(1,"0 0 4 9 14 22 23 24 25 26 \n",27) = 27
10341 write(1,"Time: 0.0184601 seconds\n",24) = 24
10341 exit_group(0) = ?
10341 +++ exited with 0 ++
```