Московский Авиационный Институт

(Национальный Исследовательский Университет)

Факультет информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

Лабораторная работа №5-7 по курсу

«Операционные системы»

Группа: М8О-209Б-22

Студент: Концебалов О.С.

Преподаватель: Пономарев Н.В.

Оценка: _____

Дата: 25.12.2023

Москва, 2023.

Содержание

Постановка задачи

Реализовать распределенную систему по асинхронной обработке запросов. В данной распределенной системе должно существовать 2 вида узлов: «управляющий» и «вычислительный». Необходимо объединить данные узлы в соответствии с той топологией, которая определена вариантом. Связь между узлами необходимо осуществить при помощи технологии очередей сообщений. Также в данной системе необходимо предусмотреть проверку доступности узлов в соответствии с вариантом. При убийстве («kill -9») любого вычислительного узла система должна пытаться максимально сохранять свою работоспособность, а именно все дочерние узлы убитого узла могут стать недоступными, но родительские узлы должны сохранить свою работоспособность.

Управляющий узел отвечает за ввод команд от пользователя и отправку этих команд на вычислительные узлы. Список основных поддерживаемых команд:

Создание нового вычислительного узла

Формат команды: create id [parent] id – целочисленный идентификатор нового вычислительного узла parent – целочисленный идентификатор родительского узла. Если топологией не предусмотрено введение данного параметра, то его необходимо игнорировать (если его ввели)

Формат вывода:

«Ok: pid», где pid – идентификатор процесса для созданного вычислительного узла

«Error: Already exists» - вычислительный узел с таким идентификатором уже существует

«Error: Parent not found» - нет такого родительского узла с таким идентификатором

«Error: Parent is unavailable» - родительский узел существует, но по каким-то причинам с ним не удается связаться

«Error: [Custom error]» - любая другая обрабатываемая ошибка

Исполнение команды на вычислительном узле

Формат команды: exec id [params] id – целочисленный идентификатор вычислительного узла, на который отправляется команда

Формат вывода:

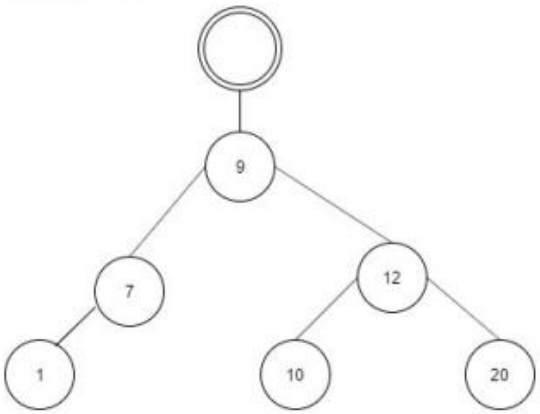
«Ok:id: [result]», где result – результат выполненной команды

«Error:id: Not found» - вычислительный узел с таким идентификатором не найден

«Error:id: Node is unavailable» - по каким-то причинам не удается связаться с вычислительным узлом

«Error:id: [Custom error]» - любая другая обрабатываемая ошибка

Топология 3



Все вычислительные узлы хранятся в бинарном дереве поиска. [parent] — является необязательным параметром.

Типы команд для вычислительных узлов

Набор команд 1 (подсчет суммы n чисел)

Формат команды: exec id n k_1 ... k_n

id – целочисленный идентификатор вычислительного узла, на который отправляется команда

n – количество складываемых чисел (от 1 до 10_8)

k_1 ... k_n – складываемые числа

Пример:

> exec 10 3 1 2 3

Ok:10: 6

Тип проверки доступности узлов

Команда проверки 1

Формат команды: pingall

Вывод всех недоступных узлов вывести разделенные через точку запятую.

Пример:

> pingall

Ok: -1 // Все узлы доступны

> pingall

Ok: 7;10;15 // узлы 7, 10, 15 — недоступны

Общие сведения о программе

Программа представляет из себя файлы main.cpp, node.cpp с реализаций обрабатывающего и вычислительного узлов. Для удобной сборки и запуска программы был написан Makefile

Общий метод и алгоритм решения

Для решения используется технология очереди сообщений ZeroMQ, а именно – модель Request/Reply, где клиент отправляет запрос, а сервер на него отвечает. Сокеты ZMQ типа REQ могут подключаться к нескольким серверам. В нашем случае, максимально к двум (число дочерних узлов бинарного дерева). При обработке запроса CREATE проверяется, создан ли корень (управляющий узел). Если нет, то он создается, иначе – запрос передается корню, тот либо сразу его обрабатывает, либо передает запрос дочерним узлам (вычислительным узлам) по правилу поиска в бинарном дереве. При обработке остальных запросов (EXEC, KILL, PINGALL) аналогично проверяется существование корня, затем запрос передается нужному узлу. При убийстве узла удаляется все его поддерево. При обработке любого запроса требуется получить ответ – статус выполнения запроса, который печатается в стандартный вывод. Если ответ не поступает по истечении заданного тайм-аута, в стандартный вывод печатается ошибка.

Код программы

./run.cpp

```
#include <bits/stdc++.h>
#include "zmq.h"

const int DEFAULT_PORT = 5050;
int n = 2;

bool send_message(void* socket, const std::string& message_string)
{
   int rc;
   zmq_msg_t msg;
   rc = zmq_msg_init_size(&msg, message_string.size());

   if (rc != 0) {
      std::cerr << "SEND_MESSAGE ERROR: Error while initializing mes-
sage: " << zmq_strerror(errno)
                << std::endl;
      return false;
   }

   memcpy(zmq_msg_data(&msg), message_string.c_str(), mes-
sage_string.size());
   rc = zmq_msg_send(&msg, socket, 0);

   if (rc == -1) {
      std::cerr << "SEND_MESSAGE ERROR: Error while sending message: "
<< zmq_strerror(errno)
                << std::endl;
      zmq_msg_close(&msg);
      return false;
   }

   zmq_msg_close(&msg);
   return true;
}

std::string receive_message(void* socket)
{
   int rc;
   zmq_msg_t msg;
```

```
    rc = zmq_msg_init(&msg);

    if (rc != 0) {
        std::cerr << "RECEIVE_MESSAGE ERROR: Error while initializing
message: " << zmq_strerror(errno)
                  << std::endl;
        return "";
    }

    rc = zmq_msg_recv(&msg, socket, 0);
    if (rc == -1) {
        std::cerr << "RECEIVE_MESSAGE ERROR: Error while recieving mes-
sage: " << zmq_strerror(errno)
                  << std::endl;
        zmq_msg_close(&msg);
        return "";
    }

    std::string received_mes-
sage(static_cast<char*>(zmq_msg_data(&msg)),
                                zmq_msg_size(&msg));

    zmq_msg_close(&msg);
    return received_message;
}

void create_node(const int& id, const int& port)
{
    char* arg0 = strdup("./node");
    char* arg1 = strdup((std::to_string(id)).c_str());
    char* arg2 = strdup((std::to_string(port)).c_str());
    char* args[] = {arg0, arg1, arg2, nullptr};

    execv("./node", args);
}

std::string get_port_name(const int& port)
{
    return "tcp://127.0.0.1:" + std::to_string(port);
}

bool is_number(const std::string& val)
```

```
{
    try {
        int tmp = stoi(val);
        return true;
    } catch (std::exception& ex) {
        std::cout << "IS_NUMBER ERROR: " << ex.what() << std::endl;
        return false;
    }
}

int main() {
    std::string command;

    int root_id = 0;
    int root_pid = 0;
    void* context = zmq_ctx_new();
    void* root_socket = zmq_socket(context, ZMQ_REQ);

    std::cout << "Commands:" << std::endl;
    std::cout << "1. create (id)" << std::endl;
    std::cout << "2. exec (id) (numbers_of_nums, k_1...k_n)" <<
std::endl;
    std::cout << "3. kill (id)" << std::endl;
    std::cout << "4. pingall" << std::endl;
    std::cout << "5. exit" << std::endl << std::endl;

    std::vector<int> node_ids;

    while (true) {
        std::cin >> command;
        int node_id = 0;

        std::string id_str = "";
        std::string reply = "";

        if (command == "create") {
            ++n;
            std::cin >> id_str;

            if (!is_number(id_str)) {
                continue;
            }
```

```
        node_id = stoi(id_str);
        node_ids.push_back(node_id);

        if (root_pid == 0) {
            zmq_bind(root_socket,
                     get_port_name(DEFAULT_PORT + node_id).c_str());

            zmq_setsockopt(root_socket, ZMQ_RCVTIMEO, NULL, n * 500);
            zmq_setsockopt(root_socket, ZMQ_SNDTIMEO, NULL, n * 500);

            root_pid = fork();
            if (root_pid == -1) {
                std::cout << "CREATE ERROR: Unable to create first
worker node\n";
                root_pid = 0;
                exit(1);
            } else if (root_pid == 0) {
                create_node(node_id, DEFAULT_PORT + node_id);
            } else {
                root_id = node_id;
                send_message(root_socket, "pid");
                reply = receive_message(root_socket);
            }
        } else {
            zmq_setsockopt(root_socket, ZMQ_RCVTIMEO, NULL, n * 500);
            zmq_setsockopt(root_socket, ZMQ_SNDTIMEO, NULL, n * 500);
            std::string request = "create " + std::to_string(node_id);
            send_message(root_socket, request);
            reply = receive_message(root_socket);
        }

        std::cout << reply << std::endl;
    }

    if (command == "kill") {
        std::cin >> id_str;

        if (root_pid == 0) {
            std::cout << "Root is dead!" << std::endl;
            continue;
        }
```

```
        if (!is_number(id_str)) {
            continue;
        }

        node_id = stoi(id_str);
        if (node_id == root_id) {
            kill(root_pid, SIGKILL);
            root_id = 0;
            root_pid = 0;
            std::cout << "Ok\n";
            continue;
        }

        std::string request = "kill " + std::to_string(node_id);
        send_message(root_socket, request);
        reply = receive_message(root_socket);
        std::cout << reply << std::endl;
    }

    if (command == "exec") {
        int number_of_nums = 0;
        std::string nums_str = "";
        std::cin >> id_str >> number_of_nums;

        for (size_t i = 0; i != number_of_nums; ++i) {
            int num;
            std::cin >> num;
            nums_str += (std::to_string(num) + "_");
        }
        nums_str.pop_back();

        if (root_pid == 0) {
            std::cout << "Root is dead!" << std::endl;
            continue;
        }
        if (!is_number(id_str)) {
            continue;
        }

        node_id = stoi(id_str);
        std::string request = "exec " + std::to_string(node_id) + " "
+ nums_str;
```

```
            send_message(root_socket, request);
            reply = receive_message(root_socket);
            std::cout << reply << std::endl;
        }

        if (command == "pingall") {
            if (root_pid == 0) {
                std::cout << "Root is dead!" << std::endl;
                continue;
            }

            std::string reply_1 = "";

            for (size_t i = 0; i != node_ids.size(); ++i) {
                if (!is_number(std::to_string(node_ids[i]))) {
                    continue;
                }

                std::string request = "ping " +
std::to_string(node_ids[i]);
                send_message(root_socket, request);
                reply_1 = receive_message(root_socket);

                if (reply_1 == "Ok: 1") {
                    reply += std::to_string(node_ids[i]) + ';';
                } else {
                    continue;
                }
            }

            if (reply.empty()) {
                reply = "Ok: -1";
            } else {
                reply = "Ok: " + reply;
            }
            reply.pop_back();

            std::cout << reply << std::endl;
        }

        if (command == "exit") {
            int t = system("killall -9 node");
```

```
            break;
        }
    }

    zmq_close(root_socket);
    zmq_ctx_destroy(context);

    return 0;
}
```

./node.cpp

```
#include <bits/stdc++.h>
#include "zmq.h"

const int DEFAULT_PORT = 5050;
int nl = 2, nr = 2;

bool send_message(void* socket, const std::string& message_string)
{
    int rc;
    zmq_msg_t msg;
    rc = zmq_msg_init_size(&msg, message_string.size());
    if (rc != 0) {
        std::cerr << "SEND_MESSAGE ERROR: Error while initializing mes-
sage: " << zmq_strerror(errno)
                  << std::endl;
        return false;
    }

    memcpy(zmq_msg_data(&msg), message_string.c_str(), mes-
sage_string.size());
    rc = zmq_msg_send(&msg, socket, 0);

    if (rc == -1) {
        std::cerr << "SEND_MESSAGE ERROR: Error while sending message: "
<< zmq_strerror(errno)
                  << std::endl;
        zmq_msg_close(&msg);
        return false;
    }
```

```
    zmq_msg_close(&msg);
    return true;
}

std::string receive_message(void* socket)
{
    int rc;
    zmq_msg_t msg;
    rc = zmq_msg_init(&msg);

    if (rc != 0) {
        std::cerr << "RECEIVE_MESSAGE ERROR: Error while initializing
message: " << zmq_strerror(errno)
                  << std::endl;
        return "";
    }

    rc = zmq_msg_recv(&msg, socket, 0);

    if (rc == -1) {
        std::cerr << "RECEIVE_MESSAGE ERROR: Error while recieving mes-
sage: " << zmq_strerror(errno)
                  << std::endl;
        zmq_msg_close(&msg);
        return "";
    }

    std::string received_mes-
sage(static_cast<char*>(zmq_msg_data(&msg)),
                                  zmq_msg_size(&msg));
    zmq_msg_close(&msg);
    return received_message;
}

void create_node(const int& id, const int& port)
{
    char* arg0 = strdup("./node");
    char* arg1 = strdup((std::to_string(id)).c_str());
    char* arg2 = strdup((std::to_string(port)).c_str());
    char* args[] = {arg0, arg1, arg2, nullptr};
    execv("./node", args);
}
```

```
std::string get_port_name(const int& port)
{
   return "tcp://127.0.0.1:" + std::to_string(port);
}

void create_helper(void* parent_socket, void* socket, int& create_id,
int& id,
                   int& pid)
{
   if (pid == -1) {
      send_message(parent_socket, "CREATE_HELPER ERROR: Cannot fork");
      pid = 0;
   } else if (pid == 0) {
      create_node(create_id, DEFAULT_PORT + create_id);
   } else {
      id = create_id;
      send_message(socket, "pid");
      send_message(parent_socket, receive_message(socket));
   }
}

void exec_helper(void* parent_socket, void* socket, int& id, int& pid,
                 std::string& request_string)
{
   if (pid == 0) {
      std::string receive_message = "EXEC_HELPER ERROR:" +
std::to_string(id);
      receive_message += ": Not found";
      send_message(parent_socket, receive_message);
   } else {
      send_message(socket, request_string);
      std::string str = receive_message(socket);
      if (str == "") str = "EXEC_HELPER ERROR: Node is unavailable";
      send_message(parent_socket, str);
   }
}

void ping_helper(void* parent_socket, void* socket, int& id, int& pid,
                 std::string& request_string)
{
   if (pid == 0) {
```

```
      std::string receive_message = "PING_HELPER ERROR:" +
std::to_string(id);
      receive_message += ": Not found";
      send_message(parent_socket, receive_message);
   } else {
      send_message(socket, request_string);
      std::string str = receive_message(socket);
      if (str == "") {
         str = "Ok: 0";
      }
      send_message(parent_socket, str);
   }
}

void exec(std::istringstream& command_stream, void* parent_socket,
          void* left_socket, void* right_socket, int& left_pid, int&
right_pid,
          int& id, std::string& request_string)
{
   std::string nums_string;
   int exec_id;
   int sum = 0;

   command_stream >> exec_id;

   if (exec_id == id) {
      command_stream >> nums_string;
      std::string reply = "";
      int index = 0;

      std::string substr_num = "";
      for (size_t i = 0; i != nums_string.length(); ++i) {
        if (nums_string[i] != '_') {
            substr_num += nums_string[i];
        } else {
            sum += std::stoi(substr_num);
            substr_num = "";
        }
      }

      sum += std::stoi(substr_num);
      reply = "Ok:" + std::to_string(id) + ":" + std::to_string(sum);
```

```
        send_message(parent_socket, reply);
    } else if (exec_id < id) {
        exec_helper(parent_socket, left_socket, exec_id, left_pid,
                    request_string);
    } else {
        exec_helper(parent_socket, right_socket, exec_id, right_pid,
                    request_string);
    }
}

void ping(std::istringstream& command_stream, void* parent_socket,
          void* left_socket, void* right_socket, int& left_pid, int&
right_pid,
          int& id, std::string& request_string)
{
    int ping_id;
    std::string reply;

    command_stream >> ping_id;

    if (ping_id == id) {
        reply = "Ok: 1";
        send_message(parent_socket, reply);
    } else if (ping_id < id) {
        ping_helper(parent_socket, left_socket, ping_id, left_pid,
                    request_string);
    } else {
        ping_helper(parent_socket, right_socket, ping_id, right_pid,
                    request_string);
    }
}

void kill_node(void* parent_socket, void* socket, int& delete_id, int&
id,
               int& pid, std::string& request_string)
{
    if (id == 0) {
        send_message(parent_socket, "Error: Not found");
    } else if (id == delete_id) {
        send_message(socket, "kill_children");
        receive_message(socket);
        kill(pid, SIGKILL);
```

```
      id = 0;
      pid = 0;
      send_message(parent_socket, "Ok");
   } else {
      send_message(socket, request_string);
      send_message(parent_socket, receive_message(socket));
   }
}

void kill_children(void* parent_socket, void* left_socket, void*
right_socket,
                   int& left_pid, int& right_pid)
{
   if (left_pid == 0 && right_pid == 0) {
      send_message(parent_socket, "Ok");
   } else {
      if (left_pid != 0) {
         send_message(left_socket, "kill_children");
         receive_message(left_socket);
         kill(left_pid, SIGKILL);
      }
      if (right_pid != 0) {
         send_message(right_socket, "kill_children");
         receive_message(right_socket);
         kill(right_pid, SIGKILL);
      }
      send_message(parent_socket, "Ok");
   }
}

int main(int argc, char** argv) {
   int id = std::stoi(argv[1]);
   int parent_port = std::stoi(argv[2]);
   void* context = zmq_ctx_new();
   void* parent_socket = zmq_socket(context, ZMQ_REP);
   void* left_socket = zmq_socket(context, ZMQ_REQ);
   void* right_socket = zmq_socket(context, ZMQ_REQ);

   zmq_connect(parent_socket, get_port_name(parent_port).c_str());
   zmq_setsockopt(parent_socket, ZMQ_RCVTIMEO, NULL, 500);
   zmq_setsockopt(parent_socket, ZMQ_SNDTIMEO, NULL, 500);
```

```
   int left_id = 0, right_id = 0, left_pid = 0, right_pid = 0;

   while (true) {
      std::string request = receive_message(parent_socket);
      std::istringstream command_stream(request);
      std::string command;

      command_stream >> command;

      if (command == "id") {
         std::string reply = "Ok: " + std::to_string(id);
         send_message(parent_socket, reply);
      }

      if (command == "pid") {
         std::string reply = "Ok: " + std::to_string(getpid());
         send_message(parent_socket, reply);
      }

      if (command == "create") {
         int create_id;
         command_stream >> create_id;

         if (create_id == id) {
            std::string reply = "Error: Already exists";
            send_message(parent_socket, reply);
         } else if (create_id < id) {
            ++nl;

            if (left_pid == 0) {
               zmq_bind(left_socket,
                        get_port_name(DEFAULT_PORT + cre-
ate_id).c_str());

               zmq_setsockopt(left_socket, ZMQ_RCVTIMEO, NULL, nl *
500);
               zmq_setsockopt(left_socket, ZMQ_SNDTIMEO, NULL, nl *
500);
               left_pid = fork();
               create_helper(parent_socket, left_socket, create_id,
left_id,
                             left_pid);
```

```
            } else {
               send_message(left_socket, request);
               std::string reply = receive_message(left_socket);

               if (reply == "") {
                  reply = "Error: Node is unavaliable";
               } else {
                  zmq_setsockopt(left_socket, ZMQ_RCVTIMEO, NULL, nl *
500);
                  zmq_setsockopt(left_socket, ZMQ_SNDTIMEO, NULL, nl *
500);
               }

               send_message(parent_socket, reply);
            }
         } else {
            ++nr;

            if (right_pid == 0) {
               zmq_bind(right_socket,
                        get_port_name(DEFAULT_PORT + cre-
ate_id).c_str());

               zmq_setsockopt(right_socket, ZMQ_RCVTIMEO, NULL, nr *
500);
               zmq_setsockopt(right_socket, ZMQ_SNDTIMEO, NULL, nr *
500);
               right_pid = fork();
               create_helper(parent_socket, right_socket, create_id,
right_id,
                             right_pid);
            } else {
               send_message(right_socket, request);
               std::string reply = receive_message(right_socket);

               if (reply == "") {
                  reply = "Error: Node is unavaliable";
               } else {
                  zmq_setsockopt(right_socket, ZMQ_RCVTIMEO, NULL, nr
* 500);
                  zmq_setsockopt(right_socket, ZMQ_SNDTIMEO, NULL, nr
* 500);
```

```
            }

            send_message(parent_socket, reply);
         }
      }
   }

   if (command == "exec") {
      exec(command_stream, parent_socket, left_socket,
right_socket,
           left_pid, right_pid, id, request);
   }

   if (command == "ping") {
      ping(command_stream, parent_socket, left_socket,
right_socket,
           left_pid, right_pid, id, request);
   }

   if (command == "kill") {
      int delete_id;
      command_stream >> delete_id;

      if (delete_id < id) {
         kill_node(parent_socket, left_socket, delete_id, left_id,
left_pid,
                   request);
      } else {
         kill_node(parent_socket, right_socket, delete_id,
right_id,
                   right_pid, request);
      }
   }

   if (command == "kill_children") {
      kill_children(parent_socket, left_socket, right_socket,
left_pid,
                    right_pid);
   }
}

zmq_close(left_socket);
```

```
    zmq_close(right_socket);
    zmq_ctx_destroy(context);

    return 0;
}
```

Использование утилиты strace

execve("./server", ["./server"], 0x7ffd1ef5f518 /* 60 vars */) = 0

brk(NULL) = 0x55e7c2782000

arch_prctl(0x3001 /* ARCH_??? */, 0x7ffcf05462d0) = -1 EINVAL (Invalid argument)

mmap(NULL, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f6415c7a000

access("/etc/ld.so.preload", R_OK) = -1 ENOENT (No such file or directory)

openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3

newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=67103, ...}, AT_EMPTY_PATH) = 0

mmap(NULL, 67103, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f6415c69000

close(3) = 0

openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libzmq.so.5", 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\240\233\1\0\0\0\0\0"..., 832) =
832

newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=634936, ...}, AT_EMPTY_PATH) =
0

mmap(NULL, 636784, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
0x7f6415bcd000

mmap(0x7f6415be5000, 397312, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x18000) = 0x7f6415be5000

mmap(0x7f6415c46000, 106496, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x79000) = 0x7f6415c46000

mmap(0x7f6415c60000, 36864, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x92000) = 0x7f6415c60000

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\0\3\0>\0\1\0\0\0\0\0\0\0\0\0\0\0"..., 832) = 832

newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=2260296, ...}, AT_EMPTY_PATH) = 0

mmap(NULL, 2275520, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f6415800000

mprotect(0x7f641589a000, 1576960, PROT_NONE) = 0

mmap(0x7f641589a000, 1118208, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x9a000) = 0x7f641589a000

mmap(0x7f64159ab000, 454656, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1ab000) = 0x7f64159ab000

mmap(0x7f6415a1b000, 57344, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x21a000) = 0x7f6415a1b000

mmap(0x7f6415a29000, 10432, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f6415a29000

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\0"..., 832) = 832

newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=125488, ...}, AT_EMPTY_PATH) = 0

mmap(NULL, 127720, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f6415bad000

mmap(0x7f6415bb0000, 94208, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x3000) = 0x7f6415bb0000

mmap(0x7f6415bc7000, 16384, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1a000) = 0x7f6415bc7000

mmap(0x7f6415bcb000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1d000) = 0x7f6415bcb000

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\0\3\0>\0\1\0\0\0P\237\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

pread64(3, "\4\0\0\0 \0\0\0\5\0\0\0GNU\0\2\0\0\300\4\0\0\0\3\0\0\0\0\0\0\0"..., 48, 848) = 48

pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0 =\340\2563\265?\356\25x\261\27\313A#\350"..., 68, 896) = 68

newfstatat(3, "", {st_mode=S_IFREG|0755, st_size=2216304, ...}, AT_EMPTY_PATH) = 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, 2260560, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f6415400000

mmap(0x7f6415428000, 1658880, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x28000) = 0x7f6415428000

mmap(0x7f64155bd000, 360448, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1bd000) = 0x7f64155bd000

mmap(0x7f6415615000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x214000) = 0x7f6415615000

mmap(0x7f641561b000, 52816, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f641561b000

close(3) = 0

openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libbsd.so.0", 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\0"..., 832) = 832

newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=89096, ...}, AT_EMPTY_PATH) = 0

mmap(NULL, 94432, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f6415b95000

mprotect(0x7f6415b99000, 69632, PROT_NONE) = 0

mmap(0x7f6415b99000, 53248, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x4000) = 0x7f6415b99000

mmap(0x7f6415ba6000, 12288, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x11000) = 0x7f6415ba6000

mmap(0x7f6415baa000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x14000) = 0x7f6415baa000

mmap(0x7f6415bac000, 224, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f6415bac000

close(3) = 0

openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libsodium.so.23", 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\0"..., 832) = 832

newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=355040, ...}, AT_EMPTY_PATH) = 0

mmap(NULL, 357440, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f6415b3d000

mprotect(0x7f6415b49000, 303104, PROT_NONE) = 0

mmap(0x7f6415b49000, 229376, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0xc000) = 0x7f6415b49000

mmap(0x7f6415b81000, 69632, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x44000) = 0x7f6415b81000

mmap(0x7f6415b93000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x55000) = 0x7f6415b93000

close(3) = 0

mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f6415b3b000

openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libpgm-5.3.so.0", 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\340L\0\0\0\0\0\0"..., 832) = 832

newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=310264, ...}, AT_EMPTY_PATH) = 0

mmap(NULL, 329808, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f6415aea000

mmap(0x7f6415aee000, 172032, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x4000) = 0x7f6415aee000

mmap(0x7f6415b18000, 118784, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x2e000) = 0x7f6415b18000

mmap(0x7f6415b35000, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x4a000) = 0x7f6415b35000

mmap(0x7f6415b37000, 14416, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f6415b37000

close(3) = 0

openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libnorm.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 \255\0\0\0\0\0\0"..., 832) = 832

newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=497824, ...}, AT_EMPTY_PATH) =
0

mmap(NULL, 1223168, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
0x7f64156d5000

mprotect(0x7f64156df000, 446464, PROT_NONE) = 0

mmap(0x7f64156df000, 286720, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0xa000) = 0x7f64156df000

mmap(0x7f6415725000, 155648, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x50000) = 0x7f6415725000

mmap(0x7f641574c000, 16384, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x76000) = 0x7f641574c000

mmap(0x7f6415750000, 719360, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f6415750000

close(3) = 0

openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libgssapi_krb5.so.2",
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\0"..., 832) = 832

newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=338648, ...}, AT_EMPTY_PATH) =
0

mmap(NULL, 340960, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
0x7f6415a96000

mprotect(0x7f6415aa1000, 282624, PROT_NONE) = 0

mmap(0x7f6415aa1000, 229376, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0xb000) = 0x7f6415aa1000

mmap(0x7f6415ad9000, 49152, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x43000) = 0x7f6415ad9000

mmap(0x7f6415ae6000, 16384, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x4f000) = 0x7f6415ae6000

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\0"..., 832) = 832

newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=940560, ...}, AT_EMPTY_PATH) =
0

mmap(NULL, 942344, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
0x7f6415319000

mmap(0x7f6415327000, 507904, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0xe000) = 0x7f6415327000

mmap(0x7f64153a3000, 372736, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x8a000) = 0x7f64153a3000

mmap(0x7f64153fe000, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0xe4000) = 0x7f64153fe000

close(3) = 0

openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libmd.so.0", O_RDONLY|O_CLOEXEC) =
3

read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\0\0\0\0\0\0\0\0"..., 832) = 832

newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=47472, ...}, AT_EMPTY_PATH) = 0

mmap(NULL, 49384, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
0x7f6415a89000

mmap(0x7f6415a8b000, 28672, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x2000) = 0x7f6415a8b000

mmap(0x7f6415a92000, 8192, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x9000) = 0x7f6415a92000

mmap(0x7f6415a94000, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0xa000) = 0x7f6415a94000

close(3) = 0

openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libpthread.so.0",
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\0"..., 832) = 832

newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=21448, ...}, AT_EMPTY_PATH) = 0

mmap(NULL, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f6415a87000

mmap(NULL, 16424, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
0x7f6415a82000

mmap(0x7f6415a83000, 4096, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1000) = 0x7f6415a83000

mmap(0x7f6415a84000, 4096, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x2000) = 0x7f6415a84000

mmap(0x7f6415a85000, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x2000) = 0x7f6415a85000

close(3) = 0

openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libkrb5.so.3", 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\0"..., 832) = 832

newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=827936, ...}, AT_EMPTY_PATH) =
0

mmap(NULL, 830576, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
0x7f641524e000

mprotect(0x7f641526f000, 634880, PROT_NONE) = 0

mmap(0x7f641526f000, 380928, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x21000) = 0x7f641526f000

mmap(0x7f64152cc000, 249856, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x7e000) = 0x7f64152cc000

mmap(0x7f641530a000, 61440, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0xbb000) = 0x7f641530a000

close(3) = 0

openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libk5crypto.so.3",
O_RDONLY|O_CLOEXEC) = 3

read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\0\0\0\0\0\0\0\0"..., 832) = 832

newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=182864, ...}, AT_EMPTY_PATH) =
0

mmap(NULL, 188472, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
0x7f6415a53000

mprotect(0x7f6415a57000, 163840, PROT_NONE) = 0

mmap(0x7f6415a57000, 110592, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x4000) = 0x7f6415a57000

mmap(0x7f6415a72000, 49152, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1f000) = 0x7f6415a72000

mmap(0x7f6415a7f000, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x2b000) = 0x7f6415a7f000

mmap(0x7f6415a81000, 56, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f6415a81000

close(3) = 0

openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libcom_err.so.2",
O_RDONLY|O_CLOEXEC) = 3

read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\0\0\0\0\0\0\0\0"..., 832) = 832

newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=18504, ...}, AT_EMPTY_PATH) = 0

mmap(NULL, 20552, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
0x7f6415a4d000

mmap(0x7f6415a4f000, 4096, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x2000) = 0x7f6415a4f000

mmap(0x7f6415a50000, 4096, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x3000) = 0x7f6415a50000

mmap(0x7f6415a51000, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x3000) = 0x7f6415a51000

close(3) = 0

openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libkrb5support.so.0",
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\0"..., 832) = 832

newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=52016, ...}, AT_EMPTY_PATH) = 0

mmap(NULL, 54224, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
0x7f6415a3f000

mprotect(0x7f6415a42000, 36864, PROT_NONE) = 0

mmap(0x7f6415a42000, 24576, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x3000) = 0x7f6415a42000

mmap(0x7f6415a48000, 8192, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x9000) = 0x7f6415a48000

mmap(0x7f6415a4b000, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0xb000) = 0x7f6415a4b000

close(3) = 0

openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libkeyutils.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\0"..., 832) = 832

newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=22600, ...}, AT_EMPTY_PATH) = 0

mmap(NULL, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f6415a3d000

mmap(NULL, 24592, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
0x7f6415a36000

mmap(0x7f6415a38000, 8192, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x2000) = 0x7f6415a38000

mmap(0x7f6415a3a000, 4096, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x4000) = 0x7f6415a3a000

mmap(0x7f6415a3b000, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x4000) = 0x7f6415a3b000

close(3)   = 0

openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libresolv.so.2",
O_RDONLY|O_CLOEXEC) = 3

read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\0\0\0\0\0\0\0\0"..., 832) = 832

newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=68552, ...}, AT_EMPTY_PATH) = 0

mmap(NULL, 80456, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
0x7f64156c1000

mmap(0x7f64156c4000, 40960, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x3000) = 0x7f64156c4000

mmap(0x7f64156ce000, 12288, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0xd000) = 0x7f64156ce000

mmap(0x7f64156d1000, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0xf000) = 0x7f64156d1000

mmap(0x7f64156d3000, 6728, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f64156d3000

close(3)   = 0

mmap(NULL, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f6415a34000

mmap(NULL, 12288, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f6415a31000

arch_prctl(ARCH_SET_FS, 0x7f6415a319c0) = 0

set_tid_address(0x7f6415a31c90)   = 4598

set_robust_list(0x7f6415a31ca0, 24)   = 0

rseq(0x7f6415a32360, 0x20, 0, 0x53053053) = 0

mprotect(0x7f6415615000, 16384, PROT_READ) = 0

mprotect(0x7f64156d1000, 4096, PROT_READ) = 0

mprotect(0x7f6415a3b000, 4096, PROT_READ) = 0

mprotect(0x7f6415a4b000, 4096, PROT_READ) = 0

```
mprotect(0x7f6415a51000, 4096, PROT_READ) = 0
mprotect(0x7f6415a7f000, 4096, PROT_READ) = 0
mprotect(0x7f641530a000, 53248, PROT_READ) = 0
mprotect(0x7f6415a85000, 4096, PROT_READ) = 0
mprotect(0x7f6415a94000, 4096, PROT_READ) = 0
mprotect(0x7f64153fe000, 4096, PROT_READ) = 0
mprotect(0x7f6415ae6000, 8192, PROT_READ) = 0
mprotect(0x7f6415bcb000, 4096, PROT_READ) = 0
mmap(NULL, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f6415a2f000
mprotect(0x7f6415a1b000, 45056, PROT_READ) = 0
mprotect(0x7f641574c000, 12288, PROT_READ) = 0
mprotect(0x7f6415b35000, 4096, PROT_READ) = 0
mprotect(0x7f6415b93000, 4096, PROT_READ) = 0
mprotect(0x7f6415baa000, 4096, PROT_READ) = 0
mprotect(0x7f6415c60000, 32768, PROT_READ) = 0
mprotect(0x55e7c1f4c000, 4096, PROT_READ) = 0
mprotect(0x7f6415cb4000, 8192, PROT_READ) = 0
prlimit64(0, RLIMIT_STACK, NULL, {rlim_cur=8192*1024,
rlim_max=RLIM64_INFINITY}) = 0
munmap(0x7f6415c69000, 67103)           = 0
getrandom("\x39\x7b\x85\x9c\xb5\x4a\x9e\x2a", 8, GRND_NONBLOCK) = 8
brk(NULL)                               = 0x55e7c2782000
brk(0x55e7c27a3000)                     = 0x55e7c27a3000
futex(0x7f6415a2977c, FUTEX_WAKE_PRIVATE, 2147483647) = 0
openat(AT_FDCWD, "/sys/devices/system/cpu/online", O_RDONLY|O_CLOEXEC) = 3
read(3, "0-5\n", 1024)                  = 4
close(3)                                = 0
```

```
openat(AT_FDCWD, "/sys/devices/system/cpu",
O_RDONLY|O_NONBLOCK|O_CLOEXEC|O_DIRECTORY) = 3

newfstatat(3, "", {st_mode=S_IFDIR|0755, st_size=0, ...}, AT_EMPTY_PATH) = 0

getdents64(3, 0x55e7c2793ee0 /* 22 entries */, 32768) = 640

getdents64(3, 0x55e7c2793ee0 /* 0 entries */, 32768) = 0

close(3)                                = 0

getpid()                                = 4598

sched_getaffinity(4598, 128, [0, 1, 2, 3, 4, 5]) = 16

newfstatat(AT_FDCWD, "/etc/nsswitch.conf", {st_mode=S_IFREG|0644, st_size=542,
...}, 0) = 0

newfstatat(AT_FDCWD, "/", {st_mode=S_IFDIR|0755, st_size=4096, ...}, 0) = 0

openat(AT_FDCWD, "/etc/nsswitch.conf", O_RDONLY|O_CLOEXEC) = 3

newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=542, ...}, AT_EMPTY_PATH) = 0

read(3, "# /etc/nsswitch.conf\n#\n# Example"..., 4096) = 542

read(3, "", 4096)                       = 0

newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=542, ...}, AT_EMPTY_PATH) = 0

close(3)                                = 0

openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3

newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=67103, ...}, AT_EMPTY_PATH) = 0

mmap(NULL, 67103, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f6415c69000

close(3)                                = 0

openat(AT_FDCWD, "/lib/x86_64-linux-gnu/glibc-hwcaps/x86-64-v3/libnss_db.so.2",
O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)

newfstatat(AT_FDCWD, "/lib/x86_64-linux-gnu/glibc-hwcaps/x86-64-v3",
0x7ffcf05432c0, 0) = -1 ENOENT (No such file or directory)

openat(AT_FDCWD, "/lib/x86_64-linux-gnu/glibc-hwcaps/x86-64-v2/libnss_db.so.2",
O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)

newfstatat(AT_FDCWD, "/lib/x86_64-linux-gnu/glibc-hwcaps/x86-64-v2",
0x7ffcf05432c0, 0) = -1 ENOENT (No such file or directory)
```

openat(AT_FDCWD, "/lib/x86_64-linux-gnu/tls/x86_64/x86_64/libnss_db.so.2", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)

newfstatat(AT_FDCWD, "/lib/x86_64-linux-gnu/tls/x86_64/x86_64", 0x7ffcf05432c0, 0) = -1 ENOENT (No such file or directory)

openat(AT_FDCWD, "/lib/x86_64-linux-gnu/tls/x86_64/libnss_db.so.2", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)

newfstatat(AT_FDCWD, "/lib/x86_64-linux-gnu/tls/x86_64", 0x7ffcf05432c0, 0) = -1 ENOENT (No such file or directory)

openat(AT_FDCWD, "/lib/x86_64-linux-gnu/tls/x86_64/libnss_db.so.2", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)

newfstatat(AT_FDCWD, "/lib/x86_64-linux-gnu/tls/x86_64", 0x7ffcf05432c0, 0) = -1 ENOENT (No such file or directory)

openat(AT_FDCWD, "/lib/x86_64-linux-gnu/tls/libnss_db.so.2", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)

newfstatat(AT_FDCWD, "/lib/x86_64-linux-gnu/tls", 0x7ffcf05432c0, 0) = -1 ENOENT (No such file or directory)

openat(AT_FDCWD, "/lib/x86_64-linux-gnu/x86_64/x86_64/libnss_db.so.2", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)

newfstatat(AT_FDCWD, "/lib/x86_64-linux-gnu/x86_64/x86_64", 0x7ffcf05432c0, 0) = -1 ENOENT (No such file or directory)

openat(AT_FDCWD, "/lib/x86_64-linux-gnu/x86_64/libnss_db.so.2", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)

newfstatat(AT_FDCWD, "/lib/x86_64-linux-gnu/x86_64", 0x7ffcf05432c0, 0) = -1 ENOENT (No such file or directory)

openat(AT_FDCWD, "/lib/x86_64-linux-gnu/x86_64/libnss_db.so.2", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)

newfstatat(AT_FDCWD, "/lib/x86_64-linux-gnu/x86_64", 0x7ffcf05432c0, 0) = -1 ENOENT (No such file or directory)

openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libnss_db.so.2", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)

newfstatat(AT_FDCWD, "/lib/x86_64-linux-gnu", {st_mode=S_IFDIR|0755, st_size=98304, ...}, 0) = 0

openat(AT_FDCWD, "/usr/lib/x86_64-linux-gnu/glibc-hwcaps/x86-64-v3/libnss_db.so.2", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)

newfstatat(AT_FDCWD, "/usr/lib/x86_64-linux-gnu/glibc-hwcaps/x86-64-v3", 0x7ffcf05432c0, 0) = -1 ENOENT (No such file or directory)

openat(AT_FDCWD, "/usr/lib/x86_64-linux-gnu/glibc-hwcaps/x86-64-v2/libnss_db.so.2", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)

newfstatat(AT_FDCWD, "/usr/lib/x86_64-linux-gnu/glibc-hwcaps/x86-64-v2", 0x7ffcf05432c0, 0) = -1 ENOENT (No such file or directory)

openat(AT_FDCWD, "/usr/lib/x86_64-linux-gnu/tls/x86_64/x86_64/libnss_db.so.2", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)

newfstatat(AT_FDCWD, "/usr/lib/x86_64-linux-gnu/tls/x86_64/x86_64", 0x7ffcf05432c0, 0) = -1 ENOENT (No such file or directory)

openat(AT_FDCWD, "/usr/lib/x86_64-linux-gnu/tls/x86_64/libnss_db.so.2", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)

newfstatat(AT_FDCWD, "/usr/lib/x86_64-linux-gnu/tls/x86_64", 0x7ffcf05432c0, 0) = -1 ENOENT (No such file or directory)

openat(AT_FDCWD, "/usr/lib/x86_64-linux-gnu/tls/x86_64/libnss_db.so.2", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)

newfstatat(AT_FDCWD, "/usr/lib/x86_64-linux-gnu/tls/x86_64", 0x7ffcf05432c0, 0) = -1 ENOENT (No such file or directory)

openat(AT_FDCWD, "/usr/lib/x86_64-linux-gnu/tls/libnss_db.so.2", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)

newfstatat(AT_FDCWD, "/usr/lib/x86_64-linux-gnu/tls", 0x7ffcf05432c0, 0) = -1 ENOENT (No such file or directory)

openat(AT_FDCWD, "/usr/lib/x86_64-linux-gnu/x86_64/x86_64/libnss_db.so.2", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)

newfstatat(AT_FDCWD, "/usr/lib/x86_64-linux-gnu/x86_64/x86_64", 0x7ffcf05432c0, 0) = -1 ENOENT (No such file or directory)

openat(AT_FDCWD, "/usr/lib/x86_64-linux-gnu/x86_64/libnss_db.so.2", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)

newfstatat(AT_FDCWD, "/usr/lib/x86_64-linux-gnu/x86_64", 0x7ffcf05432c0, 0) = -1 ENOENT (No such file or directory)

openat(AT_FDCWD, "/usr/lib/x86_64-linux-gnu/x86_64/libnss_db.so.2", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)

newfstatat(AT_FDCWD, "/usr/lib/x86_64-linux-gnu/x86_64", 0x7ffcf05432c0, 0) = -1 ENOENT (No such file or directory)

openat(AT_FDCWD, "/usr/lib/x86_64-linux-gnu/libnss_db.so.2", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)

newfstatat(AT_FDCWD, "/usr/lib/x86_64-linux-gnu", {st_mode=S_IFDIR|0755, st_size=98304, ...}, 0) = 0

openat(AT_FDCWD, "/lib/glibc-hwcaps/x86-64-v3/libnss_db.so.2", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)

newfstatat(AT_FDCWD, "/lib/glibc-hwcaps/x86-64-v3", 0x7ffcf05432c0, 0) = -1 ENOENT (No such file or directory)

openat(AT_FDCWD, "/lib/glibc-hwcaps/x86-64-v2/libnss_db.so.2", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)

newfstatat(AT_FDCWD, "/lib/glibc-hwcaps/x86-64-v2", 0x7ffcf05432c0, 0) = -1 ENOENT (No such file or directory)

openat(AT_FDCWD, "/lib/tls/x86_64/x86_64/libnss_db.so.2", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)

newfstatat(AT_FDCWD, "/lib/tls/x86_64/x86_64", 0x7ffcf05432c0, 0) = -1 ENOENT (No such file or directory)

openat(AT_FDCWD, "/lib/tls/x86_64/libnss_db.so.2", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)

newfstatat(AT_FDCWD, "/lib/tls/x86_64", 0x7ffcf05432c0, 0) = -1 ENOENT (No such file or directory)

openat(AT_FDCWD, "/lib/tls/x86_64/libnss_db.so.2", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)

newfstatat(AT_FDCWD, "/lib/tls/x86_64", 0x7ffcf05432c0, 0) = -1 ENOENT (No such file or directory)

openat(AT_FDCWD, "/lib/tls/libnss_db.so.2", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)

newfstatat(AT_FDCWD, "/lib/tls", 0x7ffcf05432c0, 0) = -1 ENOENT (No such file or directory)

openat(AT_FDCWD, "/lib/x86_64/x86_64/libnss_db.so.2", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)

newfstatat(AT_FDCWD, "/lib/x86_64/x86_64", 0x7ffcf05432c0, 0) = -1 ENOENT (No such file or directory)

openat(AT_FDCWD, "/lib/x86_64/libnss_db.so.2", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)

newfstatat(AT_FDCWD, "/lib/x86_64", 0x7ffcf05432c0, 0) = -1 ENOENT (No such file or directory)

openat(AT_FDCWD, "/lib/x86_64/libnss_db.so.2", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)

newfstatat(AT_FDCWD, "/lib/x86_64", 0x7ffcf05432c0, 0) = -1 ENOENT (No such file or directory)

openat(AT_FDCWD, "/lib/libnss_db.so.2", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)

newfstatat(AT_FDCWD, "/lib", {st_mode=S_IFDIR|0755, st_size=4096, ...}, 0) = 0

openat(AT_FDCWD, "/usr/lib/glibc-hwcaps/x86-64-v3/libnss_db.so.2", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)

newfstatat(AT_FDCWD, "/usr/lib/glibc-hwcaps/x86-64-v3", 0x7ffcf05432c0, 0) = -1 ENOENT (No such file or directory)

openat(AT_FDCWD, "/usr/lib/glibc-hwcaps/x86-64-v2/libnss_db.so.2", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)

newfstatat(AT_FDCWD, "/usr/lib/glibc-hwcaps/x86-64-v2", 0x7ffcf05432c0, 0) = -1 ENOENT (No such file or directory)

openat(AT_FDCWD, "/usr/lib/tls/x86_64/x86_64/libnss_db.so.2", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)

newfstatat(AT_FDCWD, "/usr/lib/tls/x86_64/x86_64", 0x7ffcf05432c0, 0) = -1 ENOENT (No such file or directory)

openat(AT_FDCWD, "/usr/lib/tls/x86_64/libnss_db.so.2", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)

newfstatat(AT_FDCWD, "/usr/lib/tls/x86_64", 0x7ffcf05432c0, 0) = -1 ENOENT (No such file or directory)

openat(AT_FDCWD, "/usr/lib/tls/x86_64/libnss_db.so.2", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)

newfstatat(AT_FDCWD, "/usr/lib/tls/x86_64", 0x7ffcf05432c0, 0) = -1 ENOENT (No such file or directory)

openat(AT_FDCWD, "/usr/lib/tls/libnss_db.so.2", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)

newfstatat(AT_FDCWD, "/usr/lib/tls", 0x7ffcf05432c0, 0) = -1 ENOENT (No such file or directory)

openat(AT_FDCWD, "/usr/lib/x86_64/x86_64/libnss_db.so.2", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)

newfstatat(AT_FDCWD, "/usr/lib/x86_64/x86_64", 0x7ffcf05432c0, 0) = -1 ENOENT (No such file or directory)

openat(AT_FDCWD, "/usr/lib/x86_64/libnss_db.so.2", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)

newfstatat(AT_FDCWD, "/usr/lib/x86_64", 0x7ffcf05432c0, 0) = -1 ENOENT (No such file or directory)

openat(AT_FDCWD, "/usr/lib/x86_64/libnss_db.so.2", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)

newfstatat(AT_FDCWD, "/usr/lib/x86_64", 0x7ffcf05432c0, 0) = -1 ENOENT (No such file or directory)

openat(AT_FDCWD, "/usr/lib/libnss_db.so.2", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)

newfstatat(AT_FDCWD, "/usr/lib", {st_mode=S_IFDIR|0755, st_size=4096, ...}, 0) = 0

munmap(0x7f6415c69000, 67103)   = 0

openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3

newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=67103, ...}, AT_EMPTY_PATH) = 0

mmap(NULL, 67103, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f6415c69000

close(3)   = 0

openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libnss_db-2.35.so", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)

openat(AT_FDCWD, "/usr/lib/x86_64-linux-gnu/libnss_db-2.35.so", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)

openat(AT_FDCWD, "/lib/libnss_db-2.35.so", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)

openat(AT_FDCWD, "/usr/lib/libnss_db-2.35.so", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)

munmap(0x7f6415c69000, 67103) = 0

openat(AT_FDCWD, "/etc/protocols", O_RDONLY|O_CLOEXEC) = 3

newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=2932, ...}, AT_EMPTY_PATH) = 0

lseek(3, 0, SEEK_SET) = 0

read(3, "# Internet (IP) protocols\n#\n# Up"..., 4096) = 2932

read(3, "", 4096) = 0

close(3) = 0

eventfd2(0, EFD_CLOEXEC) = 3

fcntl(3, F_GETFL) = 0x2 (flags O_RDWR)

fcntl(3, F_SETFL, O_RDWR|O_NONBLOCK) = 0

fcntl(3, F_GETFL) = 0x802 (flags O_RDWR|O_NONBLOCK)

fcntl(3, F_SETFL, O_RDWR|O_NONBLOCK) = 0

getpid() = 4598

getpid() = 4598

getrandom("\x36\x6b\xcc\x21\xa1\x7c\x07\xb7\xbe\xb9\x04\xb9\x28\x4c\xab\x31", 16, 0) = 16

getrandom("\x4a\x8a\xef\xac\x44\xaa\x91\xce\x9c\x67\xcc\x70\x86\xdf\x87\x2a", 16, 0) = 16

eventfd2(0, EFD_CLOEXEC) = 4

fcntl(4, F_GETFL) = 0x2 (flags O_RDWR)

fcntl(4, F_SETFL, O_RDWR|O_NONBLOCK) = 0

fcntl(4, F_GETFL) = 0x802 (flags O_RDWR|O_NONBLOCK)

fcntl(4, F_SETFL, O_RDWR|O_NONBLOCK) = 0

getpid()   = 4598

epoll_create1(EPOLL_CLOEXEC)   = 5

epoll_ctl(5, EPOLL_CTL_ADD, 4, {events=0, data={u32=3262726752, u64=94453888533088}}) = 0

epoll_ctl(5, EPOLL_CTL_MOD, 4, {events=EPOLLIN, data={u32=3262726752, u64=94453888533088}}) = 0

getpid()   = 4598

rt_sigaction(SIGRT_1, {sa_handler=0x7f6415491870, sa_mask=[], sa_flags=SA_RESTORER|SA_ONSTACK|SA_RESTART|SA_SIGINFO, sa_restorer=0x7f6415442520}, 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) = 0x7f6414a4d000

mprotect(0x7f6414a4e000, 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_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID, child_tid=0x7f641524d910, parent_tid=0x7f641524d910, exit_signal=0, stack=0x7f6414a4d000, stack_size=0x7ffc80, tls=0x7f641524d640}strace: Process 4599 attached

 => {parent_tid=[4599]}, 88) = 4599

[pid 4599] rseq(0x7f641524dfe0, 0x20, 0, 0x53053053 <unfinished ...>

[pid 4598] rt_sigprocmask(SIG_SETMASK, [], <unfinished ...>

[pid 4599] <... rseq resumed>)   = 0

[pid 4598] <... rt_sigprocmask resumed>NULL, 8) = 0

[pid 4599] set_robust_list(0x7f641524d920, 24 <unfinished ...>

[pid 4598] eventfd2(0, EFD_CLOEXEC <unfinished ...>

[pid 4599] <... set_robust_list resumed>) = 0

[pid 4598] <... eventfd2 resumed>)   = 6

[pid 4599] rt_sigprocmask(SIG_SETMASK, [], <unfinished ...>

[pid 4598] fcntl(6, F_GETFL <unfinished ...>

[pid 4599] <... rt_sigprocmask resumed>NULL, 8) = 0

[pid 4598] <... fcntl resumed>)   = 0x2 (flags O_RDWR)

[pid 4599] rt_sigprocmask(SIG_BLOCK, ~[RTMIN RT_1], <unfinished ...>

[pid 4598] fcntl(6, F_SETFL, O_RDWR|O_NONBLOCK <unfinished ...>

[pid 4599] <... rt_sigprocmask resumed>NULL, 8) = 0

[pid 4598] <... fcntl resumed>)   = 0

[pid 4599] sched_getparam(4599, <unfinished ...>

[pid 4598] fcntl(6, F_GETFL <unfinished ...>

[pid 4599] <... sched_getparam resumed>[0]) = 0

[pid 4598] <... fcntl resumed>)   = 0x802 (flags O_RDWR|O_NONBLOCK)

[pid 4599] sched_getscheduler(4599 <unfinished ...>

[pid 4598] fcntl(6, F_SETFL, O_RDWR|O_NONBLOCK <unfinished ...>

[pid 4599] <... sched_getscheduler resumed>) = 0 (SCHED_OTHER)

[pid 4598] <... fcntl resumed>)   = 0

[pid 4599] sched_setscheduler(4599, SCHED_OTHER, [0] <unfinished ...>

[pid 4598] getpid(<unfinished ...>

[pid 4599] <... sched_setscheduler resumed>) = 0

[pid 4598] <... getpid resumed>)   = 4598

[pid 4599] prctl(PR_SET_NAME, "ZMQbg/Reaper" <unfinished ...>

[pid 4598] epoll_create1(EPOLL_CLOEXEC <unfinished ...>

[pid 4599] <... prctl resumed>)   = 0

[pid 4598] <... epoll_create1 resumed>) = 7

[pid 4599] epoll_wait(5, <unfinished ...>

[pid 4598] epoll_ctl(7, EPOLL_CTL_ADD, 6, {events=0, data={u32=3262747808, u64=94453888554144}}) = 0

[pid 4598] epoll_ctl(7, EPOLL_CTL_MOD, 6, {events=EPOLLIN, data={u32=3262747808, u64=94453888554144}}) = 0

[pid 4598] mmap(NULL, 8392704, PROT_NONE,
MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1, 0) = 0x7f641424c000

[pid 4598] mprotect(0x7f641424d000, 8388608, PROT_READ|PROT_WRITE) = 0

[pid 4598] rt_sigprocmask(SIG_BLOCK, ~[], [], 8) = 0

[pid 4598]
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_
THREAD|CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE
_CHILD_CLEARTID, child_tid=0x7f6414a4c910, parent_tid=0x7f6414a4c910,
exit_signal=0, stack=0x7f641424c000, stack_size=0x7ffc80, tls=0x7f6414a4c640}strace:
Process 4600 attached

 => {parent_tid=[4600]}, 88) = 4600

[pid 4600] rseq(0x7f6414a4cfe0, 0x20, 0, 0x53053053 <unfinished ...>

[pid 4598] rt_sigprocmask(SIG_SETMASK, [], <unfinished ...>

[pid 4600] <... rseq resumed>)   = 0

[pid 4598] <... rt_sigprocmask resumed>NULL, 8) = 0

[pid 4600] set_robust_list(0x7f6414a4c920, 24 <unfinished ...>

[pid 4598] eventfd2(0, EFD_CLOEXEC)   = 8

[pid 4600] <... set_robust_list resumed>) = 0

[pid 4598] fcntl(8, F_GETFL <unfinished ...>

[pid 4600] rt_sigprocmask(SIG_SETMASK, [], <unfinished ...>

[pid 4598] <... fcntl resumed>)   = 0x2 (flags O_RDWR)

[pid 4600] <... rt_sigprocmask resumed>NULL, 8) = 0

[pid 4598] fcntl(8, F_SETFL, O_RDWR|O_NONBLOCK <unfinished ...>

[pid 4600] rt_sigprocmask(SIG_BLOCK, ~[RTMIN RT_1], <unfinished ...>

[pid 4598] <... fcntl resumed>)   = 0

[pid 4600] <... rt_sigprocmask resumed>NULL, 8) = 0

[pid 4598] fcntl(8, F_GETFL <unfinished ...>

[pid 4600] sched_getparam(4600, <unfinished ...>

[pid 4598] <... fcntl resumed>)   = 0x802 (flags O_RDWR|O_NONBLOCK)

[pid 4600] <... sched_getparam resumed>[0]) = 0

[pid 4598] fcntl(8, F_SETFL, O_RDWR|O_NONBLOCK <unfinished ...>

[pid 4600] sched_getscheduler(4600 <unfinished ...>

[pid 4598] <... fcntl resumed>)   = 0

[pid 4600] <... sched_getscheduler resumed>) = 0 (SCHED_OTHER)

[pid 4598] getpid(<unfinished ...>

[pid 4600] sched_setscheduler(4600, SCHED_OTHER, [0] <unfinished ...>

[pid 4598] <... getpid resumed>)   = 4598

[pid 4600] <... sched_setscheduler resumed>) = 0

[pid 4598] newfstatat(1, "", <unfinished ...>

[pid 4600] prctl(PR_SET_NAME, "ZMQbg/IO/0" <unfinished ...>

[pid 4598] <... newfstatat resumed>{st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0), ...}, AT_EMPTY_PATH) = 0

[pid 4600] <... prctl resumed>)   = 0

[pid 4598] write(1, "Commands:\n", 10 <unfinished ...>

[pid 4600] epoll_wait(7, Commands:

 <unfinished ...>

[pid 4598] <... write resumed>)   = 10

[pid 4598] write(1, "1. create (id)\n", 151. create (id)

) = 15

[pid 4598] write(1, "2. exec (id) (numbers_of_nums, k"..., 422. exec (id) (numbers_of_nums, k_1...k_n)

) = 42

[pid 4598] write(1, "3. kill (id)\n", 133. kill (id)

) = 13

[pid 4598] write(1, "4. pingall\n", 114. pingall

) = 11

[pid 4598] write(1, "5. exit\n", 85. exit

) = 8

[pid 4598] write(1, "\n", 1

) = 1

[pid 4598] newfstatat(0, "", {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0), ...}, AT_EMPTY_PATH) = 0

[pid 4598] read(0, pingall

close(7) = 0

close(6) = 0

close(5) = 0

close(4) = 0

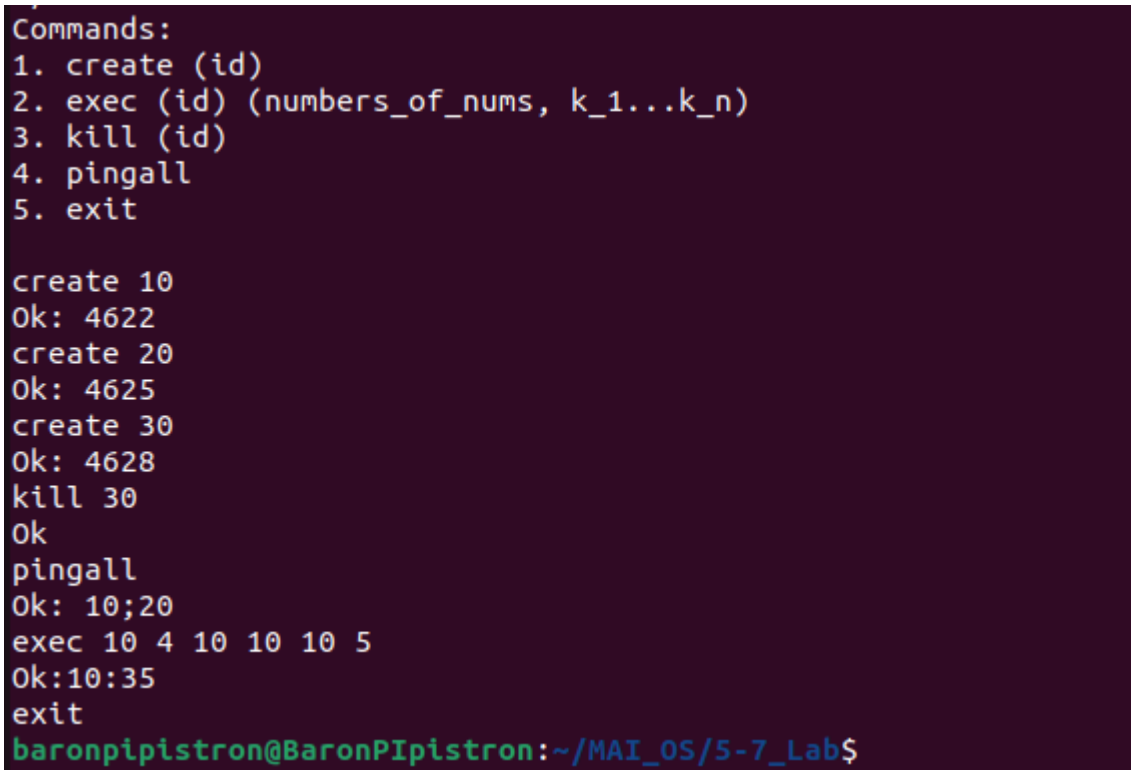
close(3) = 0

lseek(0, -1, SEEK_CUR) = -1 ESPIPE (Illegal seek)

exit_group(0) = ?

+++ exited with 0 +++

Демонстрация работы программы



Вывод

В ходе лабораторной изучил основы работы с очередями сообщений ZeroMQ и реализовал программу с использованием этой библиотеки. Когда параллельных вычислений становится недостаточно, на помощь приходят распределённые вычисления (распределение вычислений осуществляется уже не между потоками процессора, а между отдельными ЭВМ). Очереди сообщений используются для взаимодействия нескольких машин в одной большой сети. Опыт работы с ZeroMQ может пригодиться при настройке собственной системы распределённых вычислений. В целом лаба понравилась, было бы еще приятнее выполнять, если бы сроки так не поджимали.

Работа на 9 из 10, 100% пригодится в дальнейшем