Московский Авиационный Институт

(Национальный Исследовательский Университет)

Факультет информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

**Лабораторная работа №6-8 по курсу**

**«Операционные системы»**

Студент: Гаптулхаков Руслан Рамилевич

Группа: М8О–208Б–19

Вариант: 45

Преподаватель: Миронов Евгений Сергеевич

Оценка:

Дата: 20.04.2021

Подпись:

Москва, 2021

1. **Постановка задачи**

**Цель работы**

Целью является приобретение практических навыков в:

* Управлении серверами сообщений (№6)
* Применение отложенных вычислений (№7)
* Интеграция программных систем друг с другом (№8)

**Задание**

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.

Вариант 45: топология – 4, команда для вычислительных узлов - exec id start/stop/time, проверка доступности узлов – heartbit ms.

1. **Общие сведения о программе**

Для работы с очередями используется ZMQ, программа собирается при помощи Makefile. Управляющий узел – server, вычислительные узлы – client. В программе используются следующие системные вызовы:

1. kill – убивает процесс с pid – первый аргумент и посылает сигнал – второй аргумент.
2. socket.setsockopt – устанавливает флаги для сокета.
3. zmq::context\_t – создает ZMQ контекст.
4. zmq::socket\_t – создает ZMQ сокет.
5. zmq::message\_t – создает ZMQ сообщение.
6. socket.send – отправляет ZMQ сообщение на socket.
7. socket.bind – принимает соединие к сокету.
8. execv – выполняет указанный файл.
9. fork – создает копию процесса.

Программа поддерживает следующие команды:

* create [id] – создать новый узел [id].
* remove [id] – удалить узел [id]. Все дочерние узлы будут также удалены.
* exec [id] [subcmd] – таймер, который имеет четыре состояния start/stop/time
* exec [id] check [key] – запросить значение переменной [key] в словаре узла [id].
* heartbit [time ms] - каждый узел пишет о своей роботоспособности раз в time.
* exit – выйти из программы.

1. **Общий метод и алгоритм решения.**

Создаем сервер – исполняющий узел, дальше делаем fork, в дочернем процессе при помощи execv запускаем client, а с родителя с сервера отсылаем сообщение, внутри клиента также создаются сокеты – левый и правый и на них отправляются сообщения с родителя, а родитель получает сообщения от детей и так по всему дереву. Исполняющий узел получает сообщение выполняет команду и отправляет ответ вверх по дереву до управляющего узла.

1. **Основные файлы программы**

**Makefile:**

GPP = g++

GPPFLAGS = -std=c++11 -g -O2

SERVNANE = server

CLIENTNAME = client

all: server.o client.o f\_server.o tree.o

    $(GPP) $(GPPFLAGS) server.o f\_server.o tree.o -o $(SERVNANE) -lzmq

    $(GPP) $(GPPFLAGS) client.o f\_server.o -o $(CLIENTNAME) -lzmq

server.o: f\_server.h tree.h

    $(GPP) $(GPPFLAGS) -c server.cpp -o server.o

client.o: f\_server.h

    $(GPP) $(GPPFLAGS) -c client.cpp -o client.o

f\_server.o: f\_server.h

    $(GPP) $(GPPFLAGS) -c f\_server.cpp -o f\_server.o

tree.o: tree.h

    $(GPP) $(GPPFLAGS) -c tree.cpp -o tree.o

clean:

    rm \*.o $(CLIENTNAME) $(SERVNANE)

**server.cpp:**

#include <iostream>

#include <zmq.h>

#include <string>

#include <vector>

#include <signal.h>

#include <sstream>

#include <algorithm>

#include "f\_server.h"

#include "tree.h"

bool is\_number(std::string val)

{

    try

    {

        int tmp = std::stoi(val);

        return true;

    }

    catch(std::exception& e)

    {

        std::cout << "Error: " << e.what() << "\n";

        return false;

    }

}

void help()

{

    std::cout << "----------------------------------" << std::endl

    << "create [id]" << std::endl

    << "remove [id]" << std::endl

    << "exec [id] [cmd - start/stop/time]" << std::endl

    << "heartbeat [time (ms)]" << std::endl

    << "exit" << std::endl

    << "----------------------------------" << std::endl;

}

bool vct\_contains(std::vector<int>& vct, int val)

{

    for(size\_t i = 0; i < vct.size(); ++i)

    {

        if(vct[i] == val) { return true; }

    }

    return false;

}

void create(IdTree& ids, zmq::socket\_t& main\_socket, int& port,

            size\_t& child\_pid, int& child\_id)

{

    size\_t node\_id = 0;

    std::string str = "";

    std::string result = "";

    std::cin >> str;

    if(!is\_number(str)) { return; }

    node\_id = std::stoi(str);

    if(child\_pid == 0)

    {

        child\_pid = fork();

        if(child\_pid == -1)

        {

            std::cout << "Unable to create first worker node\n";

            child\_pid = 0;

            exit(1);

        }

        else if(child\_pid == 0)

        {

            create\_node(node\_id, port);

        }

        else

        {

            child\_id = node\_id;

            send\_message(main\_socket,"pid");

            result = recieve\_message(main\_socket);

        }

    }

    else

    {

        std::string msg\_s = "create " + std::to\_string(node\_id);

        send\_message(main\_socket, msg\_s);

        result = recieve\_message(main\_socket);

    }

    if (result.substr(0,2) == "Ok") { ids.insert(node\_id); }

    std::cout << result << "\n";

}

//?????

void remove(IdTree& ids, zmq::socket\_t& main\_socket, int& port,

            size\_t& child\_pid, int& child\_id)

{

    size\_t node\_id = 0;

    std::string str = "";

    std::cin >> str;

    if(!is\_number(str)) { return; }

    node\_id = std::stoi(str);

    if(ids.find(node\_id) == nullptr)

    {

        std::cout << "Error: id doesn't exist\n";

        return;

    }

    if(child\_pid == 0)

    {

        std::cout << "Error:Not found\n";

        return;

    }

    if(node\_id == child\_id)

    {

        //Нормальное завершнение процесса

        kill(child\_pid, SIGTERM);

        // Фатальное завершение процесса

        kill(child\_pid, SIGKILL);

        child\_id = 0;

        child\_pid = 0;

        ids.erase(node\_id);

        std::cout << "Ok\n";

        return;

    }

    std::string message\_string = "remove " + std::to\_string(node\_id);

    send\_message(main\_socket, message\_string);

    std::string recieved\_message = recieve\_message(main\_socket);

    if(recieved\_message.substr(0, std::min<int>(recieved\_message.size(), 2)) == "Ok")

    {

        ids.erase(node\_id);

    }

    std::cout << recieved\_message << "\n";

}

void exec(IdTree& ids, zmq::socket\_t& main\_socket, int& port,

          size\_t& child\_pid, int& child\_id)

{

    std::string id\_str = "";

    std::string subcmd = "";

    int id = 0;

    std:: cin >> id\_str >> subcmd;

    if(!is\_number(id\_str) ) { return; }

    //std::cout << subcmd << '\n';

    if((subcmd != "start") && (subcmd != "stop") && (subcmd != "time")) {

        std::cout << "Error: invalid write subcommand\n";

        return;

    }

    id = std::stoi(id\_str);

    if(ids.find(id) == nullptr)

    {

        std::cout << "Please\n";

        std::cout << "Error: id doesn't exist\n";

        std::cout << "Please\n";

        return;

    }

    std::string message\_string = "exec " + std::to\_string(id) + " ";

    message\_string += subcmd;

    send\_message(main\_socket, message\_string);

    std::string recieved\_message = recieve\_message(main\_socket);

    std::cout << recieved\_message << "\n";

}

void heartbit(IdTree& ids, zmq::socket\_t& main\_socket, int& port,

          size\_t& child\_pid, int& child\_id)

{

    int time = 0;

    std::string str\_time = "";

    bool all\_available = true;

    int rec\_id = 0;

    std::cin >>  str\_time;

    if(!is\_number(str\_time)) { return; }

    time = std::stoi(str\_time);

    if(time <= 0)

    {

        std::cout << "Error: time must be > 0\n";

        return;

    }

    std::vector<int> from\_tree = ids.get\_nodes();

    std::vector<int> recieved\_ids;

    for(size\_t i = 0; i < from\_tree.size(); ++i)

    {

        std::string msg\_tree = "heartbit " + std::to\_string(time);

        msg\_tree += " " + std::to\_string(from\_tree[i]);

        send\_message(main\_socket, msg\_tree);

        std::string recieved = recieve\_message(main\_socket);

        if(recieved != "Error: Node is not available")

        {

            try { recieved\_ids.push\_back(std::stoi(recieved)); }

            catch(...) { std::cout << "Error: stoi\n"; }

        }

    }

    for(size\_t i = 0; i < from\_tree.size(); ++i)

    {

        if(!vct\_contains(recieved\_ids, from\_tree[i]))

        {

            all\_available = false;

            std::cout << "Heartbit: node " << from\_tree[i] << " is unavailable\n";

            ids.erase(from\_tree[i]);

        }

    }

    if(all\_available) { std::cout << "Ok\n"; }

}

int main()

{

    IdTree ids;

    std::string cmd;

    size\_t child\_pid = 0;

    int child\_id = 0;

    int linger = 0;

    zmq::context\_t context(1);

    zmq::socket\_t main\_socket(context, ZMQ\_REQ);

    main\_socket.setsockopt(ZMQ\_SNDTIMEO, 2000);

    // Задержка, как долго находится сообщение для узла после закрытия узла.

    main\_socket.setsockopt(ZMQ\_LINGER, &linger, sizeof(linger));

    int port = bind\_socket(main\_socket);

    help();

    while(true)

    {

        try

        {

            std::cin >> cmd;

            if(cmd == "create")

            {

                create(ids, main\_socket, port, child\_pid, child\_id);

            }

            else if(cmd == "remove")

            {

                remove(ids, main\_socket, port, child\_pid, child\_id);

            }

            else if(cmd == "exec")

            {

                exec(ids, main\_socket, port, child\_pid, child\_id);

            }

            else if(cmd == "heartbit")

            {

                heartbit(ids, main\_socket, port, child\_pid, child\_id);

            }

            else if(cmd == "help")

            {

                help();

            }

            else if(cmd == "exit")

            {

                system("killall -9 client");

                break;

            }

        }

        catch(...)

        {

            std::cout << "something gone wrong\n";

        }

    }

    return 0;

}

**client.cpp:**

#include <iostream>

#include <zmq.hpp>

#include <string>

#include <sstream>

#include <exception>

#include <signal.h>

#include "f\_server.h"

auto start\_clock = std::chrono::high\_resolution\_clock::now();

auto stop\_clock = std::chrono::high\_resolution\_clock::now();

auto time\_clock = 0;

bool flag\_clock = false;

void rl\_create(zmq::socket\_t& parent\_socket, zmq::socket\_t& socket,

            int& port, int& create\_id, int& id, int& pid)

{

    if(pid == -1)

    {

        send\_message(parent\_socket, "Error: Cannot fork");

        pid = 0;

    }

    else if(pid == 0)

    {

        create\_node(create\_id,port);

    }

    else

    {

        id = create\_id;

        send\_message(socket, "pid");

        send\_message(parent\_socket, recieve\_message(socket));

    }

}

void create(std::istringstream& command\_stream, zmq::socket\_t& parent\_socket,

            zmq::socket\_t& left\_socket, zmq::socket\_t& right\_socket,

            int& left\_pid, int& left\_id, int& right\_pid, int& right\_id, int& id,

            int& left\_port, int& right\_port, std::string& request\_string)

{

    int create\_id;

    command\_stream >> create\_id;

    std::cout << create\_id << '\n';

    if(create\_id == id)

    {

        std::string message\_string = "Error: Already exists";

        send\_message(parent\_socket, message\_string);

    }

    else if(create\_id < id)

    {

        if(left\_pid == 0)

        {

            left\_pid = fork();

            rl\_create(parent\_socket, left\_socket, left\_port,

                   create\_id, left\_id, left\_pid);

        }

        else

        {

            send\_message(left\_socket, request\_string);

            send\_message(parent\_socket, recieve\_message(left\_socket));

        }

    }

    else

    {

        if(right\_pid == 0)

        {

            right\_pid = fork();

            rl\_create(parent\_socket, right\_socket, right\_port,

                   create\_id, right\_id, right\_pid);

        }

        else

        {

            send\_message(right\_socket, request\_string);

            send\_message(parent\_socket, recieve\_message(right\_socket));

        }

    }

}

void rl\_remove(zmq::socket\_t& parent\_socket, zmq::socket\_t& socket,

               int& port, 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");

        recieve\_message(socket);

        kill(pid,SIGTERM);

        kill(pid,SIGKILL);

        id = 0;

        pid = 0;

        send\_message(parent\_socket, "Ok");

    }

    else

    {

        send\_message(socket, request\_string);

        send\_message(parent\_socket, recieve\_message(socket));

    }

}

void remove(std::istringstream& command\_stream, zmq::socket\_t& parent\_socket,

            zmq::socket\_t& left\_socket, zmq::socket\_t& right\_socket,

            int& left\_pid, int& left\_id, int& right\_pid, int& right\_id, int& id,

            int& left\_port, int& right\_port, std::string& request\_string)

{

    int delete\_id;

    command\_stream >> delete\_id;

    if (delete\_id < id)

    {

        rl\_remove(parent\_socket, left\_socket, left\_port,

                  delete\_id, left\_id, left\_pid, request\_string);

    }

    else

    {

        rl\_remove(parent\_socket, right\_socket, right\_port,

                  delete\_id, right\_id, right\_pid, request\_string);

    }

}

void rl\_exec(zmq::socket\_t& parent\_socket, zmq::socket\_t& socket,

             int& id, int& pid, std::string& request\_string)

{

    if(pid == 0)

    {

        std::string recieve\_message = "Error:" + std::to\_string(id);

        recieve\_message += ": Not found";

        send\_message(parent\_socket, recieve\_message);

    }

    else

    {

        send\_message(socket, request\_string);

        send\_message(parent\_socket, recieve\_message(socket));

    }

}

void exec(std::istringstream& command\_stream, zmq::socket\_t& parent\_socket,

          zmq::socket\_t& left\_socket, zmq::socket\_t& right\_socket,

          int& left\_pid, int& left\_id, int& right\_pid, int& right\_id, int& id,

          int& left\_port, int& right\_port, std::string& request\_string)

{

    int exec\_id;

    command\_stream >> exec\_id;

    if(exec\_id == id)

    {

        std::string subcmd;

        command\_stream >> subcmd;

        std::string recieve\_message = "OK: " + std::to\_string(id) + " " + subcmd;

        std::cout << "begin\n";

        if(subcmd == "start") {

            start\_clock = std::chrono::high\_resolution\_clock::now();

            flag\_clock = true;

        }

        else if(subcmd == "stop")

        {

            if(flag\_clock)

            {

                stop\_clock = std::chrono::high\_resolution\_clock::now();

                time\_clock += std::chrono::duration\_cast<std::chrono::milliseconds>(stop\_clock - start\_clock).count();

                flag\_clock = false;

            }

        } else if(subcmd == "time")

        {

            recieve\_message += ": " + std::to\_string(time\_clock);

        }

        send\_message(parent\_socket, recieve\_message);

    }

    else if(exec\_id < id)

    {

        rl\_exec(parent\_socket, left\_socket, exec\_id,

                left\_pid, request\_string);

    }

    else

    {

        rl\_exec(parent\_socket, right\_socket, exec\_id,

                right\_pid, request\_string);

    }

}

bool repl(std::string& str, std::string& from, std::string& to)

{

    size\_t start\_pos = str.find(from);

    if(start\_pos == std::string::npos)

    {

        return false;

    }

    str.replace(start\_pos, from.length(), to);

    return true;

}

void heartbit(std::istringstream& command\_stream, zmq::socket\_t& parent\_socket,

              zmq::socket\_t& left\_socket, zmq::socket\_t& right\_socket, int& left\_pid,

              int& right\_pid, int& left\_id, int& right\_id,

              int& id, std::string& request\_string)

{

    bool flag = true;

    int time = 0;

    int my\_id = 0;

    command\_stream >> time >> my\_id;

    left\_socket.setsockopt(ZMQ\_RCVTIMEO, 4\*time);

    right\_socket.setsockopt(ZMQ\_RCVTIMEO, 4\*time);

    if(my\_id == id)

    {

        send\_message(parent\_socket, std::to\_string(id));

    }

    else if(my\_id < id)

    {

        send\_message(left\_socket, request\_string);

        send\_message(parent\_socket, recieve\_message(left\_socket));

    }

    else if(my\_id > id)

    {

        send\_message(right\_socket, request\_string);

        send\_message(parent\_socket, recieve\_message(right\_socket));

    }

}

void kill\_children(zmq::socket\_t& parent\_socket, zmq::socket\_t& left\_socket,

                   zmq::socket\_t& 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");

            recieve\_message(left\_socket);

            kill(left\_pid,SIGTERM);

            kill(left\_pid,SIGKILL);

        }

        if(right\_pid != 0)

        {

            send\_message(right\_socket, "kill\_children");

            recieve\_message(right\_socket);

            kill(right\_pid,SIGTERM);

            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]);

    zmq::context\_t context(3);

    zmq::socket\_t parent\_socket(context, ZMQ\_REP);

    parent\_socket.connect(get\_port\_name(parent\_port));

    int left\_pid = 0;

    int right\_pid = 0;

    int left\_id = 0;

    int right\_id = 0;

    int linger = 0;

    zmq::socket\_t left\_socket(context, ZMQ\_REQ);

    zmq::socket\_t right\_socket(context, ZMQ\_REQ);

    left\_socket.setsockopt(ZMQ\_SNDTIMEO, 2000);

    left\_socket.setsockopt(ZMQ\_LINGER, &linger, sizeof(linger));

    right\_socket.setsockopt(ZMQ\_SNDTIMEO, 2000);

    right\_socket.setsockopt(ZMQ\_LINGER, &linger, sizeof(linger));

    int left\_port = bind\_socket(left\_socket);

    int right\_port = bind\_socket(right\_socket);

    while(true)

    {

        std::string request\_string = recieve\_message(parent\_socket);

        std::istringstream command\_stream(request\_string);

        std::string command;

        command\_stream >> command;

        if(command == "id")

        {

            std::string parent\_string = "Ok:" + std::to\_string(id);

            send\_message(parent\_socket, parent\_string);

        }

        else if(command == "pid")

        {

            std::string parent\_string = "Ok:" + std::to\_string(getpid());

            send\_message(parent\_socket, parent\_string);

        }

        else if(command == "create")

        {

            create(command\_stream, parent\_socket, left\_socket, right\_socket,

                   left\_pid, left\_id, right\_pid, right\_id, id, left\_port,

                   right\_port, request\_string);

        }

        else if(command == "remove")

        {

            remove(command\_stream, parent\_socket, left\_socket, right\_socket,

                   left\_pid, left\_id, right\_pid, right\_id, id, left\_port,

                   right\_port, request\_string);

        }

        else if(command == "exec")

        {

            exec(command\_stream, parent\_socket, left\_socket, right\_socket,

                 left\_pid, left\_id, right\_pid, right\_id, id, left\_port,

                 right\_port, request\_string);

        }

        else if(command == "heartbit")

        {

            heartbit(command\_stream, parent\_socket, left\_socket, right\_socket,

                     left\_pid, right\_pid, left\_id, right\_id, id, request\_string);

        }

        else if(command == "kill\_children")

        {

            kill\_children(parent\_socket, left\_socket,

                          right\_socket,  left\_pid, right\_pid);

        }

        if(parent\_port == 0) { break; }

    }

}

**tree.cpp:**

#include<iostream>

#include <vector>

#include <algorithm>

#include "tree.h"

int pow2(int val)

{

    return (1 << val);

}

IdTree::~IdTree()

{

    delete\_node(root);

}

void IdTree::insert(int id)

{

    root = insert(root, id);

}

void IdTree::erase(int id)

{

    root = remove(root, id);

}

TreeNode\* IdTree::find(int id)

{

    return find(root, id);

}

void IdTree::delete\_node(TreeNode\* node)

{

    if(node == nullptr) { return; }

    delete\_node(node->right);

    delete\_node(node->left);

    delete node;

}

std::vector<int> IdTree::get\_nodes() const

{

    std::vector<int> result;

    get\_nodes(root, result);

    return result;

}

void IdTree::get\_nodes(TreeNode\* node, std::vector<int>& v) const

{

    if(node == nullptr) { return; }

    get\_nodes(node->left,v);

    v.push\_back(node->data);

    get\_nodes(node->right, v);

}

TreeNode\* IdTree::find(TreeNode\* r, int id)

{

    if(r == nullptr || r->data == id) { return r; }

    if(id < r->data) { return find(r->left, id); }

    if(id > r->data) { return find(r->right, id); }

    return nullptr;

}

int IdTree::height(TreeNode\* node)

{

    int h = 0;

    if(node != nullptr)

    {

        int l\_height = height(node->left);

        int r\_height = height(node->right);

        int max\_height = std::max(l\_height, r\_height);

        h = max\_height + 1;

    }

    return h;

}

int IdTree::diff(TreeNode\* node)

{

    int l\_height = height(node->left);

    int r\_height = height(node->right);

    int b\_factor = l\_height - r\_height;

    return b\_factor;

}

TreeNode\* IdTree::rr\_rotation(TreeNode\* parent)

{

    TreeNode\* node;

    node = parent->right;

    parent->right = node->left;

    node->left = parent;

    return node;

}

TreeNode\* IdTree::ll\_rotation(TreeNode\* parent)

{

    TreeNode\* node;

    node = parent->left;

    parent->left = node->right;

    node->right = parent;

    return node;

}

TreeNode\* IdTree::lr\_rotation(TreeNode\* parent)

{

    TreeNode\* node;

    node = parent->left;

    parent->left = rr\_rotation(node);

    return ll\_rotation(parent);

}

TreeNode\* IdTree::rl\_rotation(TreeNode\* parent)

{

    TreeNode\* node;

    node = parent->right;

    parent->right = ll\_rotation(node);

    return rr\_rotation(parent);

}

TreeNode\* IdTree::balance(TreeNode\* node)

{

    int bal\_factor = diff(node);

    if(bal\_factor > 1)

    {

        if(diff(node->left) > 0) { node = ll\_rotation(node); }

        else { node = lr\_rotation(node); }

    }

    else if(bal\_factor < -1)

    {

        if(diff(node->right) > 0) { node = rl\_rotation(node); }

        else { node = rr\_rotation(node); }

    }

    return node;

}

TreeNode\* IdTree::find\_min(TreeNode\* node)

{

    if(node == nullptr) { return nullptr; }

    else if(node->left == nullptr) { return node; }

    else { return find\_min(node->left); }

}

TreeNode\* IdTree::insert(TreeNode\* root, int val)

{

    if(root == nullptr)

    {

        root = new TreeNode;

        root->data = val;

        root->left = nullptr;

        root->right = nullptr;

        return root;

    }

    else if(val < root->data)

    {

        root->left = insert(root->left, val);

        //root = balance(root);

    }

    else if(val >= root->data)

    {

        root->right = insert(root->right, val);

        //root = balance(root);

    }

    return root;

}

TreeNode\* IdTree::remove(TreeNode\* root\_node, int val)

{

    TreeNode\* node;

    if(root\_node == nullptr) { return nullptr; }

    else if(val < root\_node->data) { /\*root\_node->left =\*/ remove(root\_node->left, val); }

    else if(val >root\_node->data) { /\*root\_node->right =\*/ remove(root\_node->right, val); }

    else if(root\_node->left && root\_node->right)

    {

        node = find\_min(root\_node->right);

        root\_node->data = node->data;

        root\_node->right = remove(root\_node->right, root\_node->data);

    }

    else

    {

        node = root\_node;

        if(root\_node->left == nullptr) { root\_node = root\_node->right; }

        else if(root\_node->right == nullptr) { root\_node = root\_node->left; }

        //delete node;

    }

    if(root\_node == nullptr) { return root\_node; }

    //root\_node = balance(root\_node);

    return nullptr;

}

**tree.h:**

#pragma once

#include <vector>

struct TreeNode

{

    int data;

    TreeNode\* left;

    TreeNode\* right;

};

class IdTree

{

public:

    void insert(int);

    void erase(int);

    TreeNode\* find(int);

    std::vector<int> get\_nodes() const;

    ~IdTree();

private:

    TreeNode\* root = nullptr;

    int height(TreeNode\*);

    int diff(TreeNode\*);

    TreeNode\* rr\_rotation(TreeNode\*);

    TreeNode\* ll\_rotation(TreeNode\*);

    TreeNode\* lr\_rotation(TreeNode\*);

    TreeNode\* rl\_rotation(TreeNode\*);

    TreeNode\* balance(TreeNode \*);

    TreeNode\* insert(TreeNode \*, int);

    TreeNode\* remove(TreeNode\* t, int);

    TreeNode\* find(TreeNode\*, int);

    TreeNode\* find\_min(TreeNode\*);

    void get\_nodes(TreeNode\*,std::vector<int>&) const;

    void delete\_node(TreeNode\*);

};

**f\_server.cpp:**

#include <iostream>

#include "f\_server.h"

int send\_message(zmq::socket\_t& socket, const std::string& message\_string)

{

    zmq::message\_t message(message\_string.size());

    memcpy(message.data(), message\_string.c\_str(), message\_string.size());

    return socket.send(message);

}

std::string recieve\_message(zmq::socket\_t& socket)

{

    zmq::message\_t message;

    bool ok = false;

    try { ok = socket.recv(&message); }

    catch(...) { ok = false; }

    std::string recieved\_message(static\_cast<char\*>(message.data()), message.size());

    if(recieved\_message.empty() || !ok) { return "Error: Node is not available";}

    return recieved\_message;

}

std::string get\_port\_name(int port) { return "tcp://127.0.0.1:" + std::to\_string(port); }

int bind\_socket(zmq::socket\_t& socket)

{

    int port = PORT;

    while (true)

    {

        try

        {

            socket.bind(get\_port\_name(port));

            break;

        } catch(...) { ++port; }

    }

    return port;

}

void create\_node(int id, int port)

{

    char\* arg0 = strdup(CLIENTNAME);

    char\* arg1 = strdup((std::to\_string(id)).c\_str());

    char\* arg2 = strdup((std::to\_string(port)).c\_str());

    char\* args[] = {arg0, arg1, arg2, NULL};

    execv(CLIENTNAME, args);

}

**f\_server.h:**

#pragma once

#include <string>

//#include <zconf.h>

#include <zmq.hpp>

#include <unistd.h>

#ifndef PORT

#define PORT 1234

#endif

#ifndef CLIENTNAME

#define CLIENTNAME "./client"

#endif

int send\_message(zmq::socket\_t&,const std::string&);

std::string recieve\_message(zmq::socket\_t&);

std::string get\_port\_name(int);

int bind\_socket(zmq::socket\_t&);

void create\_node(int,int);

1. **Пример работы**

**rusya@DESKTOP-93JGKCU:/mnt/c/Users/rusya/Desktop/Lab8\_OS$** ./server

----------------------------------

create [id]

remove [id]

exec [id] [cmd - start/stop/time]

heartbeat [time (ms)]

exit

----------------------------------

create 3

Ok:1626

create 4

4

Ok:1631

create 5

5

5

Ok:1636

create 2

2

Ok:1641

create 1

1

1

Ok:1646

exec 5 start

OK: 5 start

exec 5 stop

OK: 5 stop

exec 5 time

OK: 5 time: 6957

exec 3 time

OK: 3 time: 0

exec 3 start

OK: 3 start

exec 3 stop

OK: 3 stop

exec 3 time

OK: 3 time: 5221

exec 1 start

OK: 1 start

exec 1 stop

OK: 1 stop

exec 1 time

OK: 1 time: 4692

heartbit 1000

Ok

remove 1

Ok

Exit

**rusya@DESKTOP-93JGKCU:/mnt/c/Users/rusya/Desktop/Lab678\_OS$** strace -o strace\_log.txt ./server

execve("./server", ["./server"], 0x7ffe73559000 /\* 27 vars \*/) = 0

brk(NULL) = 0x560f7c10a000

arch\_prctl(0x3001 /\* ARCH\_??? \*/, 0x7ffeaf80ebe0) = -1 EINVAL (Invalid argument)

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

fstat(3, {st\_mode=S\_IFREG|0644, st\_size=33170, ...}) = 0

mmap(NULL, 33170, PROT\_READ, MAP\_PRIVATE, 3, 0) = 0x7ff4a092c000

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`z\1\0\0\0\0\0"..., 832) = 832

fstat(3, {st\_mode=S\_IFREG|0644, st\_size=675776, ...}) = 0

mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7ff4a092a000

mmap(NULL, 678128, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7ff4a0884000

mmap(0x7ff4a089a000, 430080, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x16000) = 0x7ff4a089a000

mmap(0x7ff4a0903000, 126976, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x7f000) = 0x7ff4a0903000

mmap(0x7ff4a0922000, 32768, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x9d000) = 0x7ff4a0922000

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\240\341\t\0\0\0\0\0"..., 832) = 832

fstat(3, {st\_mode=S\_IFREG|0644, st\_size=1952928, ...}) = 0

mmap(NULL, 1968128, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7ff4a06a3000

mprotect(0x7ff4a0739000, 1286144, PROT\_NONE) = 0

mmap(0x7ff4a0739000, 983040, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x96000) = 0x7ff4a0739000

mmap(0x7ff4a0829000, 299008, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x186000) = 0x7ff4a0829000

mmap(0x7ff4a0873000, 57344, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1cf000) = 0x7ff4a0873000

mmap(0x7ff4a0881000, 10240, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7ff4a0881000

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\3405\0\0\0\0\0\0"..., 832) = 832

fstat(3, {st\_mode=S\_IFREG|0644, st\_size=104984, ...}) = 0

mmap(NULL, 107592, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7ff4a0688000

mmap(0x7ff4a068b000, 73728, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x3000) = 0x7ff4a068b000

mmap(0x7ff4a069d000, 16384, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x15000) = 0x7ff4a069d000

mmap(0x7ff4a06a1000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x18000) = 0x7ff4a06a1000

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\0\360q\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\20\0\0\0\5\0\0\0GNU\0\2\0\0\300\4\0\0\0\3\0\0\0\0\0\0\0", 32, 848) = 32

pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\t\233\222%\274\260\320\31\331\326\10\204\276X>\263"..., 68, 880) = 68

fstat(3, {st\_mode=S\_IFREG|0755, st\_size=2029224, ...}) = 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

pread64(3, "\4\0\0\0\20\0\0\0\5\0\0\0GNU\0\2\0\0\300\4\0\0\0\3\0\0\0\0\0\0\0", 32, 848) = 32

pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\t\233\222%\274\260\320\31\331\326\10\204\276X>\263"..., 68, 880) = 68

mmap(NULL, 2036952, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7ff4a0496000

mprotect(0x7ff4a04bb000, 1847296, PROT\_NONE) = 0

mmap(0x7ff4a04bb000, 1540096, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x25000) = 0x7ff4a04bb000

mmap(0x7ff4a0633000, 303104, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x19d000) = 0x7ff4a0633000

mmap(0x7ff4a067e000, 24576, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1e7000) = 0x7ff4a067e000

mmap(0x7ff4a0684000, 13528, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7ff4a0684000

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\200\302\0\0\0\0\0\0"..., 832) = 832

fstat(3, {st\_mode=S\_IFREG|0644, st\_size=355016, ...}) = 0

mmap(NULL, 357384, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7ff4a043e000

mmap(0x7ff4a044a000, 229376, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xc000) = 0x7ff4a044a000

mmap(0x7ff4a0482000, 73728, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x44000) = 0x7ff4a0482000

mmap(0x7ff4a0494000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x55000) = 0x7ff4a0494000

close(3) = 0

openat(AT\_FDCWD, "/lib/x86\_64-linux-gnu/libpgm-5.2.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\240L\0\0\0\0\0\0"..., 832) = 832

fstat(3, {st\_mode=S\_IFREG|0644, st\_size=302056, ...}) = 0

mmap(NULL, 321584, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7ff4a03ef000

mmap(0x7ff4a03f3000, 163840, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x4000) = 0x7ff4a03f3000

mmap(0x7ff4a041b000, 118784, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x2c000) = 0x7ff4a041b000

mmap(0x7ff4a0438000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x48000) = 0x7ff4a0438000

mmap(0x7ff4a043a000, 14384, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7ff4a043a000

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\0\257\0\0\0\0\0\0"..., 832) = 832

fstat(3, {st\_mode=S\_IFREG|0644, st\_size=690344, ...}) = 0

mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7ff4a03ed000

mmap(NULL, 1420000, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7ff4a0292000

mmap(0x7ff4a029c000, 421888, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xa000) = 0x7ff4a029c000

mmap(0x7ff4a0303000, 217088, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x71000) = 0x7ff4a0303000

mmap(0x7ff4a0338000, 16384, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xa5000) = 0x7ff4a0338000

mmap(0x7ff4a033c000, 723680, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7ff4a033c000

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\0P\321\0\0\0\0\0\0"..., 832) = 832

fstat(3, {st\_mode=S\_IFREG|0644, st\_size=309712, ...}) = 0

mmap(NULL, 312128, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7ff4a0245000

mmap(0x7ff4a0250000, 204800, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xb000) = 0x7ff4a0250000

mmap(0x7ff4a0282000, 49152, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x3d000) = 0x7ff4a0282000

mmap(0x7ff4a028e000, 16384, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x48000) = 0x7ff4a028e000

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\220\201\0\0\0\0\0\0"..., 832) = 832

pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\345Ga\367\265T\320\374\301V)Yf]\223\337"..., 68, 824) = 68

fstat(3, {st\_mode=S\_IFREG|0755, st\_size=157224, ...}) = 0

pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\345Ga\367\265T\320\374\301V)Yf]\223\337"..., 68, 824) = 68

mmap(NULL, 140408, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7ff4a0222000

mmap(0x7ff4a0229000, 69632, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x7000) = 0x7ff4a0229000

mmap(0x7ff4a023a000, 20480, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x18000) = 0x7ff4a023a000

mmap(0x7ff4a023f000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1c000) = 0x7ff4a023f000

mmap(0x7ff4a0241000, 13432, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7ff4a0241000

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\0\3\0>\0\1\0\0\0\300\363\0\0\0\0\0\0"..., 832) = 832

fstat(3, {st\_mode=S\_IFREG|0644, st\_size=1369352, ...}) = 0

mmap(NULL, 1368336, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7ff4a00d3000

mmap(0x7ff4a00e2000, 684032, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xf000) = 0x7ff4a00e2000

mmap(0x7ff4a0189000, 618496, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xb6000) = 0x7ff4a0189000

mmap(0x7ff4a0220000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x14c000) = 0x7ff4a0220000

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 ?\2\0\0\0\0\0"..., 832) = 832

fstat(3, {st\_mode=S\_IFREG|0644, st\_size=902016, ...}) = 0

mmap(NULL, 904640, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7ff49fff6000

mprotect(0x7ff4a0018000, 700416, PROT\_NONE) = 0

mmap(0x7ff4a0018000, 397312, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x22000) = 0x7ff4a0018000

mmap(0x7ff4a0079000, 299008, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x83000) = 0x7ff4a0079000

mmap(0x7ff4a00c3000, 65536, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xcc000) = 0x7ff4a00c3000

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\0\3\0>\0\1\0\0\0\240D\0\0\0\0\0\0"..., 832) = 832

fstat(3, {st\_mode=S\_IFREG|0644, st\_size=191040, ...}) = 0

mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7ff49fff4000

mmap(NULL, 196696, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7ff49ffc3000

mprotect(0x7ff49ffc7000, 172032, PROT\_NONE) = 0

mmap(0x7ff49ffc7000, 114688, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x4000) = 0x7ff49ffc7000

mmap(0x7ff49ffe3000, 53248, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x20000) = 0x7ff49ffe3000

mmap(0x7ff49fff1000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x2d000) = 0x7ff49fff1000

mmap(0x7ff49fff3000, 88, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7ff49fff3000

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\0\3\0>\0\1\0\0\0\200$\0\0\0\0\0\0"..., 832) = 832

fstat(3, {st\_mode=S\_IFREG|0644, st\_size=22600, ...}) = 0

mmap(NULL, 24744, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7ff49ffbc000

mmap(0x7ff49ffbe000, 8192, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x2000) = 0x7ff49ffbe000

mmap(0x7ff49ffc0000, 4096, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x4000) = 0x7ff49ffc0000

mmap(0x7ff49ffc1000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x4000) = 0x7ff49ffc1000

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\3605\0\0\0\0\0\0"..., 832) = 832

fstat(3, {st\_mode=S\_IFREG|0644, st\_size=56096, ...}) = 0

mmap(NULL, 58344, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7ff49ffad000

mmap(0x7ff49ffb0000, 28672, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x3000) = 0x7ff49ffb0000

mmap(0x7ff49ffb7000, 12288, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xa000) = 0x7ff49ffb7000

mmap(0x7ff49ffba000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xc000) = 0x7ff49ffba000

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"..., 832) = 832

fstat(3, {st\_mode=S\_IFREG|0644, st\_size=22600, ...}) = 0

mmap(NULL, 24592, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7ff49ffa6000

mmap(0x7ff49ffa8000, 8192, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x2000) = 0x7ff49ffa8000

mmap(0x7ff49ffaa000, 4096, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x4000) = 0x7ff49ffaa000

mmap(0x7ff49ffab000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x4000) = 0x7ff49ffab000

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

fstat(3, {st\_mode=S\_IFREG|0644, st\_size=101320, ...}) = 0

mmap(NULL, 113280, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7ff49ff8a000

mprotect(0x7ff49ff8e000, 81920, PROT\_NONE) = 0

mmap(0x7ff49ff8e000, 65536, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x4000) = 0x7ff49ff8e000

mmap(0x7ff49ff9e000, 12288, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x14000) = 0x7ff49ff9e000

mmap(0x7ff49ffa2000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x17000) = 0x7ff49ffa2000

mmap(0x7ff49ffa4000, 6784, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7ff49ffa4000

close(3) = 0

openat(AT\_FDCWD, "/lib/x86\_64-linux-gnu/libdl.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 \22\0\0\0\0\0\0"..., 832) = 832

fstat(3, {st\_mode=S\_IFREG|0644, st\_size=18816, ...}) = 0

mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7ff49ff88000

mmap(NULL, 20752, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7ff49ff82000

mmap(0x7ff49ff83000, 8192, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1000) = 0x7ff49ff83000

mmap(0x7ff49ff85000, 4096, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x3000) = 0x7ff49ff85000

mmap(0x7ff49ff86000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x3000) = 0x7ff49ff86000

close(3) = 0

mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7ff49ff80000

mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7ff49ff7e000

arch\_prctl(ARCH\_SET\_FS, 0x7ff49ff81600) = 0

mprotect(0x7ff4a067e000, 12288, PROT\_READ) = 0

mprotect(0x7ff49ff86000, 4096, PROT\_READ) = 0

mprotect(0x7ff49ffa2000, 4096, PROT\_READ) = 0

mprotect(0x7ff49ffab000, 4096, PROT\_READ) = 0

mprotect(0x7ff49ffba000, 4096, PROT\_READ) = 0

mprotect(0x7ff4a023f000, 4096, PROT\_READ) = 0

mprotect(0x7ff49ffc1000, 4096, PROT\_READ) = 0

mprotect(0x7ff49fff1000, 4096, PROT\_READ) = 0

mprotect(0x7ff4a00c3000, 57344, PROT\_READ) = 0

mprotect(0x7ff4a0220000, 4096, PROT\_READ) = 0

mprotect(0x7ff4a028e000, 8192, PROT\_READ) = 0

mprotect(0x7ff4a06a1000, 4096, PROT\_READ) = 0

mprotect(0x7ff4a0873000, 45056, PROT\_READ) = 0

mprotect(0x7ff4a0338000, 12288, PROT\_READ) = 0

mprotect(0x7ff4a0438000, 4096, PROT\_READ) = 0

mprotect(0x7ff4a0494000, 4096, PROT\_READ) = 0

mprotect(0x7ff4a0922000, 28672, PROT\_READ) = 0

mprotect(0x560f7a782000, 4096, PROT\_READ) = 0

mprotect(0x7ff4a0962000, 4096, PROT\_READ) = 0

munmap(0x7ff4a092c000, 33170) = 0

set\_tid\_address(0x7ff49ff818d0) = 212

set\_robust\_list(0x7ff49ff818e0, 24) = 0

rt\_sigaction(SIGRTMIN, {sa\_handler=0x7ff4a0229bf0, sa\_mask=[], sa\_flags=SA\_RESTORER|SA\_SIGINFO, sa\_restorer=0x7ff4a02373c0}, NULL, 8) = 0

rt\_sigaction(SIGRT\_1, {sa\_handler=0x7ff4a0229c90, sa\_mask=[], sa\_flags=SA\_RESTORER|SA\_RESTART|SA\_SIGINFO, sa\_restorer=0x7ff4a02373c0}, NULL, 8) = 0

rt\_sigprocmask(SIG\_UNBLOCK, [RTMIN RT\_1], NULL, 8) = 0

prlimit64(0, RLIMIT\_STACK, NULL, {rlim\_cur=8192\*1024, rlim\_max=RLIM64\_INFINITY}) = 0

brk(NULL) = 0x560f7c10a000

brk(0x560f7c12b000) = 0x560f7c12b000

futex(0x7ff4a08816bc, FUTEX\_WAKE\_PRIVATE, 2147483647) = 0

futex(0x7ff4a08816c8, FUTEX\_WAKE\_PRIVATE, 2147483647) = 0

openat(AT\_FDCWD, "/sys/devices/system/cpu/online", O\_RDONLY|O\_CLOEXEC) = 3

read(3, "0-11\n", 8192) = 5

close(3) = 0

openat(AT\_FDCWD, "/sys/devices/system/cpu", O\_RDONLY|O\_NONBLOCK|O\_CLOEXEC|O\_DIRECTORY) = 3

fstat(3, {st\_mode=S\_IFDIR|0755, st\_size=0, ...}) = 0

getdents64(3, /\* 25 entries \*/, 32768) = 704

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

close(3) = 0

getpid() = 212

sched\_getaffinity(212, 128, [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]) = 32

openat(AT\_FDCWD, "/etc/nsswitch.conf", O\_RDONLY|O\_CLOEXEC) = 3

fstat(3, {st\_mode=S\_IFREG|0644, st\_size=510, ...}) = 0

read(3, "# /etc/nsswitch.conf\n#\n# Example"..., 4096) = 510

read(3, "", 4096) = 0

close(3) = 0

openat(AT\_FDCWD, "/etc/ld.so.cache", O\_RDONLY|O\_CLOEXEC) = 3

fstat(3, {st\_mode=S\_IFREG|0644, st\_size=33170, ...}) = 0

mmap(NULL, 33170, PROT\_READ, MAP\_PRIVATE, 3, 0) = 0x7ff4a092c000

close(3) = 0

openat(AT\_FDCWD, "/lib/x86\_64-linux-gnu/tls/haswell/x86\_64/libnss\_db.so.2", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (No such file or directory)

stat("/lib/x86\_64-linux-gnu/tls/haswell/x86\_64", 0x7ffeaf80bed0) = -1 ENOENT (No such file or directory)

openat(AT\_FDCWD, "/lib/x86\_64-linux-gnu/tls/haswell/libnss\_db.so.2", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (No such file or directory)

stat("/lib/x86\_64-linux-gnu/tls/haswell", 0x7ffeaf80bed0) = -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)

stat("/lib/x86\_64-linux-gnu/tls/x86\_64", 0x7ffeaf80bed0) = -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)

stat("/lib/x86\_64-linux-gnu/tls", 0x7ffeaf80bed0) = -1 ENOENT (No such file or directory)

openat(AT\_FDCWD, "/lib/x86\_64-linux-gnu/haswell/x86\_64/libnss\_db.so.2", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (No such file or directory)

stat("/lib/x86\_64-linux-gnu/haswell/x86\_64", 0x7ffeaf80bed0) = -1 ENOENT (No such file or directory)

openat(AT\_FDCWD, "/lib/x86\_64-linux-gnu/haswell/libnss\_db.so.2", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (No such file or directory)

stat("/lib/x86\_64-linux-gnu/haswell", 0x7ffeaf80bed0) = -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)

stat("/lib/x86\_64-linux-gnu/x86\_64", 0x7ffeaf80bed0) = -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)

stat("/lib/x86\_64-linux-gnu", {st\_mode=S\_IFDIR|0755, st\_size=36864, ...}) = 0

openat(AT\_FDCWD, "/usr/lib/x86\_64-linux-gnu/tls/haswell/x86\_64/libnss\_db.so.2", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (No such file or directory)

stat("/usr/lib/x86\_64-linux-gnu/tls/haswell/x86\_64", 0x7ffeaf80bed0) = -1 ENOENT (No such file or directory)

openat(AT\_FDCWD, "/usr/lib/x86\_64-linux-gnu/tls/haswell/libnss\_db.so.2", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (No such file or directory)

stat("/usr/lib/x86\_64-linux-gnu/tls/haswell", 0x7ffeaf80bed0) = -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)

stat("/usr/lib/x86\_64-linux-gnu/tls/x86\_64", 0x7ffeaf80bed0) = -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)

stat("/usr/lib/x86\_64-linux-gnu/tls", 0x7ffeaf80bed0) = -1 ENOENT (No such file or directory)

openat(AT\_FDCWD, "/usr/lib/x86\_64-linux-gnu/haswell/x86\_64/libnss\_db.so.2", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (No such file or directory)

stat("/usr/lib/x86\_64-linux-gnu/haswell/x86\_64", 0x7ffeaf80bed0) = -1 ENOENT (No such file or directory)

openat(AT\_FDCWD, "/usr/lib/x86\_64-linux-gnu/haswell/libnss\_db.so.2", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (No such file or directory)

stat("/usr/lib/x86\_64-linux-gnu/haswell", 0x7ffeaf80bed0) = -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)

stat("/usr/lib/x86\_64-linux-gnu/x86\_64", 0x7ffeaf80bed0) = -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)

stat("/usr/lib/x86\_64-linux-gnu", {st\_mode=S\_IFDIR|0755, st\_size=36864, ...}) = 0

openat(AT\_FDCWD, "/lib/tls/haswell/x86\_64/libnss\_db.so.2", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (No such file or directory)

stat("/lib/tls/haswell/x86\_64", 0x7ffeaf80bed0) = -1 ENOENT (No such file or directory)

openat(AT\_FDCWD, "/lib/tls/haswell/libnss\_db.so.2", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (No such file or directory)

stat("/lib/tls/haswell", 0x7ffeaf80bed0) = -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)

stat("/lib/tls/x86\_64", 0x7ffeaf80bed0) = -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)

stat("/lib/tls", 0x7ffeaf80bed0) = -1 ENOENT (No such file or directory)

openat(AT\_FDCWD, "/lib/haswell/x86\_64/libnss\_db.so.2", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (No such file or directory)

stat("/lib/haswell/x86\_64", 0x7ffeaf80bed0) = -1 ENOENT (No such file or directory)

openat(AT\_FDCWD, "/lib/haswell/libnss\_db.so.2", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (No such file or directory)

stat("/lib/haswell", 0x7ffeaf80bed0) = -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)

stat("/lib/x86\_64", 0x7ffeaf80bed0) = -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)

stat("/lib", {st\_mode=S\_IFDIR|0755, st\_size=4096, ...}) = 0

openat(AT\_FDCWD, "/usr/lib/tls/haswell/x86\_64/libnss\_db.so.2", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (No such file or directory)

stat("/usr/lib/tls/haswell/x86\_64", 0x7ffeaf80bed0) = -1 ENOENT (No such file or directory)

openat(AT\_FDCWD, "/usr/lib/tls/haswell/libnss\_db.so.2", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (No such file or directory)

stat("/usr/lib/tls/haswell", 0x7ffeaf80bed0) = -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)

stat("/usr/lib/tls/x86\_64", 0x7ffeaf80bed0) = -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)

stat("/usr/lib/tls", 0x7ffeaf80bed0) = -1 ENOENT (No such file or directory)

openat(AT\_FDCWD, "/usr/lib/haswell/x86\_64/libnss\_db.so.2", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (No such file or directory)

stat("/usr/lib/haswell/x86\_64", 0x7ffeaf80bed0) = -1 ENOENT (No such file or directory)

openat(AT\_FDCWD, "/usr/lib/haswell/libnss\_db.so.2", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (No such file or directory)

stat("/usr/lib/haswell", 0x7ffeaf80bed0) = -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)

stat("/usr/lib/x86\_64", 0x7ffeaf80bed0) = -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)

stat("/usr/lib", {st\_mode=S\_IFDIR|0755, st\_size=4096, ...}) = 0

munmap(0x7ff4a092c000, 33170) = 0

openat(AT\_FDCWD, "/etc/ld.so.cache", O\_RDONLY|O\_CLOEXEC) = 3

fstat(3, {st\_mode=S\_IFREG|0644, st\_size=33170, ...}) = 0

mmap(NULL, 33170, PROT\_READ, MAP\_PRIVATE, 3, 0) = 0x7ff4a092c000

close(3) = 0

openat(AT\_FDCWD, "/lib/x86\_64-linux-gnu/libnss\_files.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\3005\0\0\0\0\0\0"..., 832) = 832

fstat(3, {st\_mode=S\_IFREG|0644, st\_size=51832, ...}) = 0

mmap(NULL, 79672, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7ff49ff6a000

mmap(0x7ff49ff6d000, 28672, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x3000) = 0x7ff49ff6d000

mmap(0x7ff49ff74000, 8192, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xa000) = 0x7ff49ff74000

mmap(0x7ff49ff76000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xb000) = 0x7ff49ff76000

mmap(0x7ff49ff78000, 22328, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7ff49ff78000

close(3) = 0

mprotect(0x7ff49ff76000, 4096, PROT\_READ) = 0

munmap(0x7ff4a092c000, 33170) = 0

openat(AT\_FDCWD, "/etc/protocols", O\_RDONLY|O\_CLOEXEC) = 3

lseek(3, 0, SEEK\_CUR) = 0

fstat(3, {st\_mode=S\_IFREG|0644, st\_size=2932, ...}) = 0

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

lseek(3, 0, SEEK\_CUR) = 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

getrandom("\x9b\xd6\x09\xae\x19\xea\x43\xaa\x38\xa1\xef\x4b\xad\x7e\xce\xb8", 16, 0) = 16

getrandom("\x63\xfd\xdc\x8f\x05\xe4\xbd\xcb\x85\xe8\xe1\xaf\x55\xf3\xd8\xd2", 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

epoll\_create1(EPOLL\_CLOEXEC) = 5

epoll\_ctl(5, EPOLL\_CTL\_ADD, 4, {0, {u32=2081541472, u64=94624506039648}}) = 0

epoll\_ctl(5, EPOLL\_CTL\_MOD, 4, {EPOLLIN, {u32=2081541472, u64=94624506039648}}) = 0

mmap(NULL, 8392704, PROT\_NONE, MAP\_PRIVATE|MAP\_ANONYMOUS|MAP\_STACK, -1, 0) = 0x7ff49f769000

mprotect(0x7ff49f76a000, 8388608, PROT\_READ|PROT\_WRITE) = 0

clone(child\_stack=0x7ff49ff68d30, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, parent\_tid=[213], tls=0x7ff49ff69700, child\_tidptr=0x7ff49ff699d0) = 213

eventfd2(0, EFD\_CLOEXEC) = 6

fcntl(6, F\_GETFL) = 0x2 (flags O\_RDWR)

fcntl(6, F\_SETFL, O\_RDWR|O\_NONBLOCK) = 0

fcntl(6, F\_GETFL) = 0x802 (flags O\_RDWR|O\_NONBLOCK)

fcntl(6, F\_SETFL, O\_RDWR|O\_NONBLOCK) = 0

epoll\_create1(EPOLL\_CLOEXEC) = 7

epoll\_ctl(7, EPOLL\_CTL\_ADD, 6, {0, {u32=2081540800, u64=94624506038976}}) = 0

epoll\_ctl(7, EPOLL\_CTL\_MOD, 6, {EPOLLIN, {u32=2081540800, u64=94624506038976}}) = 0

mmap(NULL, 8392704, PROT\_NONE, MAP\_PRIVATE|MAP\_ANONYMOUS|MAP\_STACK, -1, 0) = 0x7ff49ef68000

mprotect(0x7ff49ef69000, 8388608, PROT\_READ|PROT\_WRITE) = 0

clone(child\_stack=0x7ff49f767d30, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, parent\_tid=[214], tls=0x7ff49f768700, child\_tidptr=0x7ff49f7689d0) = 214

eventfd2(0, EFD\_CLOEXEC) = 8

fcntl(8, F\_GETFL) = 0x2 (flags O\_RDWR)

fcntl(8, F\_SETFL, O\_RDWR|O\_NONBLOCK) = 0

fcntl(8, F\_GETFL) = 0x802 (flags O\_RDWR|O\_NONBLOCK)

fcntl(8, F\_SETFL, O\_RDWR|O\_NONBLOCK) = 0

poll([{fd=8, events=POLLIN}], 1, 0) = 0 (Timeout)

socket(AF\_NETLINK, SOCK\_RAW|SOCK\_CLOEXEC, NETLINK\_ROUTE) = 9

bind(9, {sa\_family=AF\_NETLINK, nl\_pid=0, nl\_groups=00000000}, 12) = 0

getsockname(9, {sa\_family=AF\_NETLINK, nl\_pid=212, nl\_groups=00000000}, [12]) = 0

sendto(9, {{len=20, type=RTM\_GETLINK, flags=NLM\_F\_REQUEST|NLM\_F\_DUMP, seq=1618838898, pid=0}, {ifi\_family=AF\_UNSPEC, ...}}, 20, 0, {sa\_family=AF\_NETLINK, nl\_pid=0, nl\_groups=00000000}, 12) = 20

recvmsg(9, {msg\_name={sa\_family=AF\_NETLINK, nl\_pid=0, nl\_groups=00000000}, msg\_namelen=12, msg\_iov=[{iov\_base=[{{len=1316, type=RTM\_NEWLINK, flags=NLM\_F\_MULTI, seq=1618838898, pid=212}, {ifi\_family=AF\_UNSPEC, ifi\_type=ARPHRD\_LOOPBACK, ifi\_index=if\_nametoindex("lo"), ifi\_flags=IFF\_UP|IFF\_LOOPBACK|IFF\_RUNNING|IFF\_LOWER\_UP, ifi\_change=0}, [{{nla\_len=7, nla\_type=IFLA\_IFNAME}, "lo"}, {{nla\_len=8, nla\_type=IFLA\_TXQLEN}, 1000}, {{nla\_len=5, nla\_type=IFLA\_OPERSTATE}, 0}, {{nla\_len=5, nla\_type=IFLA\_LINKMODE}, 0}, {{nla\_len=8, nla\_type=IFLA\_MTU}, 65536}, {{nla\_len=8, nla\_type=IFLA\_MIN\_MTU}, 0}, {{nla\_len=8, nla\_type=IFLA\_MAX\_MTU}, 0}, {{nla\_len=8, nla\_type=IFLA\_GROUP}, 0}, {{nla\_len=8, nla\_type=IFLA\_PROMISCUITY}, 0}, {{nla\_len=8, nla\_type=IFLA\_NUM\_TX\_QUEUES}, 1}, {{nla\_len=8, nla\_type=IFLA\_GSO\_MAX\_SEGS}, 65535}, {{nla\_len=8, nla\_type=IFLA\_GSO\_MAX\_SIZE}, 65536}, {{nla\_len=8, nla\_type=IFLA\_NUM\_RX\_QUEUES}, 1}, {{nla\_len=5, nla\_type=IFLA\_CARRIER}, 1}, {{nla\_len=12, nla\_type=IFLA\_QDISC}, "noqueue"}, {{nla\_len=8, nla\_type=IFLA\_CARRIER\_CHANGES}, 0}, {{nla\_len=5, nla\_type=IFLA\_PROTO\_DOWN}, 0}, {{nla\_len=8, nla\_type=IFLA\_CARRIER\_UP\_COUNT}, 0}, {{nla\_len=8, nla\_type=IFLA\_CARRIER\_DOWN\_COUNT}, 0}, {{nla\_len=36, nla\_type=IFLA\_MAP}, {mem\_start=0, mem\_end=0, base\_addr=0, irq=0, dma=0, port=0}}, {{nla\_len=10, nla\_type=IFLA\_ADDRESS}, "\x00\x00\x00\x00\x00\x00"}, {{nla\_len=10, nla\_type=IFLA\_BROADCAST}, "\x00\x00\x00\x00\x00\x00"}, {{nla\_len=196, nla\_type=IFLA\_STATS64}, {rx\_packets=242, tx\_packets=242, rx\_bytes=14817, tx\_bytes=14817, rx\_errors=0, tx\_errors=0, rx\_dropped=0, tx\_dropped=0, multicast=0, collisions=0, rx\_length\_errors=0, rx\_over\_errors=0, rx\_crc\_errors=0, rx\_frame\_errors=0, rx\_fifo\_errors=0, rx\_missed\_errors=0, tx\_aborted\_errors=0, tx\_carrier\_errors=0, tx\_fifo\_errors=0, tx\_heartbeat\_errors=0, tx\_window\_errors=0, rx\_compressed=0, tx\_compressed=0, rx\_nohandler=0}}, {{nla\_len=100, nla\_type=IFLA\_STATS}, {rx\_packets=242, tx\_packets=242, rx\_bytes=14817, tx\_bytes=14817, rx\_errors=0, tx\_errors=0, rx\_dropped=0, tx\_dropped=0, multicast=0, collisions=0, rx\_length\_errors=0, rx\_over\_errors=0, rx\_crc\_errors=0, rx\_frame\_errors=0, rx\_fifo\_errors=0, rx\_missed\_errors=0, tx\_aborted\_errors=0, tx\_carrier\_errors=0, tx\_fifo\_errors=0, tx\_heartbeat\_errors=0, tx\_window\_errors=0, rx\_compressed=0, tx\_compressed=0, rx\_nohandler=0}}, {{nla\_len=12, nla\_type=IFLA\_XDP}, {{nla\_len=5, nla\_type=IFLA\_XDP\_ATTACHED}, XDP\_ATTACHED\_NONE}}, {{nla\_len=760, nla\_type=IFLA\_AF\_SPEC}, [{{nla\_len=136, nla\_type=AF\_INET}, {{nla\_len=132, nla\_type=IFLA\_INET\_CONF}, [[IPV4\_DEVCONF\_FORWARDING-1] = 0, [IPV4\_DEVCONF\_MC\_FORWARDING-1] = 0, [IPV4\_DEVCONF\_PROXY\_ARP-1] = 0, [IPV4\_DEVCONF\_ACCEPT\_REDIRECTS-1] = 1, [IPV4\_DEVCONF\_SECURE\_REDIRECTS-1] = 1, [IPV4\_DEVCONF\_SEND\_REDIRECTS-1] = 1, [IPV4\_DEVCONF\_SHARED\_MEDIA-1] = 1, [IPV4\_DEVCONF\_RP\_FILTER-1] = 0, [IPV4\_DEVCONF\_ACCEPT\_SOURCE\_ROUTE-1] = 1, [IPV4\_DEVCONF\_BOOTP\_RELAY-1] = 0, [IPV4\_DEVCONF\_LOG\_MARTIANS-1] = 0, [IPV4\_DEVCONF\_TAG-1] = 0, [IPV4\_DEVCONF\_ARPFILTER-1] = 0, [IPV4\_DEVCONF\_MEDIUM\_ID-1] = 0, [IPV4\_DEVCONF\_NOXFRM-1] = 1, [IPV4\_DEVCONF\_NOPOLICY-1] = 1, [IPV4\_DEVCONF\_FORCE\_IGMP\_VERSION-1] = 0, [IPV4\_DEVCONF\_ARP\_ANNOUNCE-1] = 0, [IPV4\_DEVCONF\_ARP\_IGNORE-1] = 0, [IPV4\_DEVCONF\_PROMOTE\_SECONDARIES-1] = 0, [IPV4\_DEVCONF\_ARP\_ACCEPT-1] = 0, [IPV4\_DEVCONF\_ARP\_NOTIFY-1] = 0, [IPV4\_DEVCONF\_ACCEPT\_LOCAL-1] = 0, [IPV4\_DEVCONF\_SRC\_VMARK-1] = 0, [IPV4\_DEVCONF\_PROXY\_ARP\_PVLAN-1] = 0, [IPV4\_DEVCONF\_ROUTE\_LOCALNET-1] = 0, [IPV4\_DEVCONF\_IGMPV2\_UNSOLICITED\_REPORT\_INTERVAL-1] = 10000, [IPV4\_DEVCONF\_IGMPV3\_UNSOLICITED\_REPORT\_INTERVAL-1] = 1000, [IPV4\_DEVCONF\_IGNORE\_ROUTES\_WITH\_LINKDOWN-1] = 0, [IPV4\_DEVCONF\_DROP\_UNICAST\_IN\_L2\_MULTICAST-1] = 0, [IPV4\_DEVCONF\_DROP\_GRATUITOUS\_ARP-1] = 0, [IPV4\_DEVCONF\_BC\_FORWARDING-1] = 0]}}, {{nla\_len=620, nla\_type=AF\_INET6}, [{{nla\_len=8, nla\_type=IFLA\_INET6\_FLAGS}, IF\_READY}, {{nla\_len=20, nla\_type=IFLA\_INET6\_CACHEINFO}, {max\_reasm\_len=65535, tstamp=62, reachable\_time=41240, retrans\_time=1000}}, {{nla\_len=208, nla\_type=IFLA\_INET6\_CONF}, [[DEVCONF\_FORWARDING] = 0, [DEVCONF\_HOPLIMIT] = 64, [DEVCONF\_MTU6] = 65536, [DEVCONF\_ACCEPT\_RA] = 1, [DEVCONF\_ACCEPT\_REDIRECTS] = 1, [DEVCONF\_AUTOCONF] = 1, [DEVCONF\_DAD\_TRANSMITS] = 1, [DEVCONF\_RTR\_SOLICITS] = -1, [DEVCONF\_RTR\_SOLICIT\_INTERVAL] = 4000, [DEVCONF\_RTR\_SOLICIT\_DELAY] = 1000, [DEVCONF\_USE\_TEMPADDR] = -1, [DEVCONF\_TEMP\_VALID\_LFT] = 604800, [DEVCONF\_TEMP\_PREFERED\_LFT] = 86400, [DEVCONF\_REGEN\_MAX\_RETRY] = 3, [DEVCONF\_MAX\_DESYNC\_FACTOR] = 600, [DEVCONF\_MAX\_ADDRESSES] = 16, [DEVCONF\_FORCE\_MLD\_VERSION] = 0, [DEVCONF\_ACCEPT\_RA\_DEFRTR] = 1, [DEVCONF\_ACCEPT\_RA\_PINFO] = 1, [DEVCONF\_ACCEPT\_RA\_RTR\_PREF] = 0, [DEVCONF\_RTR\_PROBE\_INTERVAL] = 0, [DEVCONF\_ACCEPT\_RA\_RT\_INFO\_MAX\_PLEN] = 0, [DEVCONF\_PROXY\_NDP] = 0, [DEVCONF\_OPTIMISTIC\_DAD] = 0, [DEVCONF\_ACCEPT\_SOURCE\_ROUTE] = 0, [DEVCONF\_MC\_FORWARDING] = 0, [DEVCONF\_DISABLE\_IPV6] = 0, [DEVCONF\_ACCEPT\_DAD] = -1, [DEVCONF\_FORCE\_TLLAO] = 0, [DEVCONF\_NDISC\_NOTIFY] = 0, [DEVCONF\_MLDV1\_UNSOLICITED\_REPORT\_INTERVAL] = 10000, [DEVCONF\_MLDV2\_UNSOLICITED\_REPORT\_INTERVAL] = 1000, ...]}, {{nla\_len=300, nla\_type=IFLA\_INET6\_STATS}, [[IPSTATS\_MIB\_NUM] = 37, [IPSTATS\_MIB\_INPKTS] = 0, [IPSTATS\_MIB\_INOCTETS] = 0, [IPSTATS\_MIB\_INDELIVERS] = 0, [IPSTATS\_MIB\_OUTFORWDATAGRAMS] = 0, [IPSTATS\_MIB\_OUTPKTS] = 0, [IPSTATS\_MIB\_OUTOCTETS] = 0, [IPSTATS\_MIB\_INHDRERRORS] = 0, [IPSTATS\_MIB\_INTOOBIGERRORS] = 0, [IPSTATS\_MIB\_INNOROUTES] = 0, [IPSTATS\_MIB\_INADDRERRORS] = 0, [IPSTATS\_MIB\_INUNKNOWNPROTOS] = 0, [IPSTATS\_MIB\_INTRUNCATEDPKTS] = 0, [IPSTATS\_MIB\_INDISCARDS] = 0, [IPSTATS\_MIB\_OUTDISCARDS] = 0, [IPSTATS\_MIB\_OUTNOROUTES] = 0, [IPSTATS\_MIB\_REASMTIMEOUT] = 0, [IPSTATS\_MIB\_REASMREQDS] = 0, [IPSTATS\_MIB\_REASMOKS] = 0, [IPSTATS\_MIB\_REASMFAILS] = 0, [IPSTATS\_MIB\_FRAGOKS] = 0, [IPSTATS\_MIB\_FRAGFAILS] = 0, [IPSTATS\_MIB\_FRAGCREATES] = 0, [IPSTATS\_MIB\_INMCASTPKTS] = 0, [IPSTATS\_MIB\_OUTMCASTPKTS] = 0, [IPSTATS\_MIB\_INBCASTPKTS] = 0, [IPSTATS\_MIB\_OUTBCASTPKTS] = 0, [IPSTATS\_MIB\_INMCASTOCTETS] = 0, [IPSTATS\_MIB\_OUTMCASTOCTETS] = 0, [IPSTATS\_MIB\_INBCASTOCTETS] = 0, [IPSTATS\_MIB\_OUTBCASTOCTETS] = 0, [IPSTATS\_MIB\_CSUMERRORS] = 0, ...]}, {{nla\_len=52, nla\_type=IFLA\_INET6\_ICMP6STATS}, [[ICMP6\_MIB\_NUM] = 6, [ICMP6\_MIB\_INMSGS] = 0, [ICMP6\_MIB\_INERRORS] = 0, [ICMP6\_MIB\_OUTMSGS] = 0, [ICMP6\_MIB\_OUTERRORS] = 0, [ICMP6\_MIB\_CSUMERRORS] = 0]}, {{nla\_len=20, nla\_type=IFLA\_INET6\_TOKEN}, inet\_pton(AF\_INET6, "::")}, {{nla\_len=5, nla\_type=IFLA\_INET6\_ADDR\_GEN\_MODE}, IN6\_ADDR\_GEN\_MODE\_EUI64}]}]}]}, {{len=1508, type=RTM\_NEWLINK, flags=NLM\_F\_MULTI, seq=1618838898, pid=212}, {ifi\_family=AF\_UNSPEC, ifi\_type=ARPHRD\_ETHER, ifi\_index=if\_nametoindex("bond0"), ifi\_flags=IFF\_BROADCAST|IFF\_MASTER|IFF\_MULTICAST, ifi\_change=0}, [{{nla\_len=10, nla\_type=IFLA\_IFNAME}, "bond0"}, {{nla\_len=8, nla\_type=IFLA\_TXQLEN}, 1000}, {{nla\_len=5, nla\_type=IFLA\_OPERSTATE}, 2}, {{nla\_len=5, nla\_type=IFLA\_LINKMODE}, 0}, {{nla\_len=8, nla\_type=IFLA\_MTU}, 1500}, {{nla\_len=8, nla\_type=IFLA\_MIN\_MTU}, 68}, {{nla\_len=8, nla\_type=IFLA\_MAX\_MTU}, 65535}, {{nla\_len=8, nla\_type=IFLA\_GROUP}, 0}, {{nla\_len=8, nla\_type=IFLA\_PROMISCUITY}, 0}, {{nla\_len=8, nla\_type=IFLA\_NUM\_TX\_QUEUES}, 16}, {{nla\_len=8, nla\_type=IFLA\_GSO\_MAX\_SEGS}, 65535}, {{nla\_len=8, nla\_type=IFLA\_GSO\_MAX\_SIZE}, 65536}, {{nla\_len=8, nla\_type=IFLA\_NUM\_RX\_QUEUES}, 16}, {{nla\_len=5, nla\_type=IFLA\_CARRIER}, 0}, {{nla\_len=9, nla\_type=IFLA\_QDISC}, "noop"}, {{nla\_len=8, nla\_type=IFLA\_CARRIER\_CHANGES}, 1}, {{nla\_len=5, nla\_type=IFLA\_PROTO\_DOWN}, 0}, {{nla\_len=8, nla\_type=IFLA\_CARRIER\_UP\_COUNT}, 0}, {{nla\_len=8, nla\_type=IFLA\_CARRIER\_DOWN\_COUNT}, 1}, {{nla\_len=36, nla\_type=IFLA\_MAP}, {mem\_start=0, mem\_end=0, base\_addr=0, irq=0, dma=0, port=0}}, {{nla\_len=10, nla\_type=IFLA\_ADDRESS}, "\xca\xdd\x08\x64\x9f\xd5"}, {{nla\_len=10, nla\_type=IFLA\_BROADCAST}, "\xff\xff\xff\xff\xff\xff"}, {{nla\_len=196, nla\_type=IFLA\_STATS64}, {rx\_packets=0, tx\_packets=0, rx\_bytes=0, tx\_bytes=0, rx\_errors=0, tx\_errors=0, rx\_dropped=0, tx\_dropped=0, multicast=0, collisions=0, rx\_length\_errors=0, rx\_over\_errors=0, rx\_crc\_errors=0, rx\_frame\_errors=0, rx\_fifo\_errors=0, rx\_missed\_errors=0, tx\_aborted\_errors=0, tx\_carrier\_errors=0, tx\_fifo\_errors=0, tx\_heartbeat\_errors=0, tx\_window\_errors=0, rx\_compressed=0, tx\_compressed=0, rx\_nohandler=0}}, {{nla\_len=100, nla\_type=IFLA\_STATS}, {rx\_packets=0, tx\_packets=0, rx\_bytes=0, tx\_bytes=0, rx\_errors=0, tx\_errors=0, rx\_dropped=0, tx\_dropped=0, multicast=0, collisions=0, rx\_length\_errors=0, rx\_over\_errors=0, rx\_crc\_errors=0, rx\_frame\_errors=0, rx\_fifo\_errors=0, rx\_missed\_errors=0, tx\_aborted\_errors=0, tx\_carrier\_errors=0, tx\_fifo\_errors=0, tx\_heartbeat\_errors=0, tx\_window\_errors=0, rx\_compressed=0, tx\_compressed=0, rx\_nohandler=0}}, {{nla\_len=12, nla\_type=IFLA\_XDP}, {{nla\_len=5, nla\_type=IFLA\_XDP\_ATTACHED}, XDP\_ATTACHED\_NONE}}, {{nla\_len=188, nla\_type=IFLA\_LINKINFO}, [{{nla\_len=9, nla\_type=IFLA\_INFO\_KIND}, "bond"}, {{nla\_len=172, nla\_type=IFLA\_INFO\_DATA}, "\x05\x00\x01\x00\x00\x00\x00\x00\x08\x00\x03\x00\x00\x00\x00\x00\x08\x00\x04\x00\x00\x00\x00\x00\x08\x00\x05\x00\x00\x00\x00\x00"...}]}, {{nla\_len=760, nla\_type=IFLA\_AF\_SPEC}, [{{nla\_len=136, nla\_type=AF\_INET}, {{nla\_len=132, nla\_type=IFLA\_INET\_CONF}, [[IPV4\_DEVCONF\_FORWARDING-1] = 0, [IPV4\_DEVCONF\_MC\_FORWARDING-1] = 0, [IPV4\_DEVCONF\_PROXY\_ARP-1] = 0, [IPV4\_DEVCONF\_ACCEPT\_REDIRECTS-1] = 1, [IPV4\_DEVCONF\_SECURE\_REDIRECTS-1] = 1, [IPV4\_DEVCONF\_SEND\_REDIRECTS-1] = 1, [IPV4\_DEVCONF\_SHARED\_MEDIA-1] = 1, [IPV4\_DEVCONF\_RP\_FILTER-1] = 0, [IPV4\_DEVCONF\_ACCEPT\_SOURCE\_ROUTE-1] = 1, [IPV4\_DEVCONF\_BOOTP\_RELAY-1] = 0, [IPV4\_DEVCONF\_LOG\_MARTIANS-1] = 0, [IPV4\_DEVCONF\_TAG-1] = 0, [IPV4\_DEVCONF\_ARPFILTER-1] = 0, [IPV4\_DEVCONF\_MEDIUM\_ID-1] = 0, [IPV4\_DEVCONF\_NOXFRM-1] = 0, [IPV4\_DEVCONF\_NOPOLICY-1] = 0, [IPV4\_DEVCONF\_FORCE\_IGMP\_VERSION-1] = 0, [IPV4\_DEVCONF\_ARP\_ANNOUNCE-1] = 0, [IPV4\_DEVCONF\_ARP\_IGNORE-1] = 0, [IPV4\_DEVCONF\_PROMOTE\_SECONDARIES-1] = 0, [IPV4\_DEVCONF\_ARP\_ACCEPT-1] = 0, [IPV4\_DEVCONF\_ARP\_NOTIFY-1] = 0, [IPV4\_DEVCONF\_ACCEPT\_LOCAL-1] = 0, [IPV4\_DEVCONF\_SRC\_VMARK-1] = 0, [IPV4\_DEVCONF\_PROXY\_ARP\_PVLAN-1] = 0, [IPV4\_DEVCONF\_ROUTE\_LOCALNET-1] = 0, [IPV4\_DEVCONF\_IGMPV2\_UNSOLICITED\_REPORT\_INTERVAL-1] = 10000, [IPV4\_DEVCONF\_IGMPV3\_UNSOLICITED\_REPORT\_INTERVAL-1] = 1000, [IPV4\_DEVCONF\_IGNORE\_ROUTES\_WITH\_LINKDOWN-1] = 0, [IPV4\_DEVCONF\_DROP\_UNICAST\_IN\_L2\_MULTICAST-1] = 0, [IPV4\_DEVCONF\_DROP\_GRATUITOUS\_ARP-1] = 0, [IPV4\_DEVCONF\_BC\_FORWARDING-1] = 0]}}, {{nla\_len=620, nla\_type=AF\_INET6}, [{{nla\_len=8, nla\_type=IFLA\_INET6\_FLAGS}, 0}, {{nla\_len=20, nla\_type=IFLA\_INET6\_CACHEINFO}, {max\_reasm\_len=65535, tstamp=29, reachable\_time=43210, retrans\_time=1000}}, {{nla\_len=208, nla\_type=IFLA\_INET6\_CONF}, [[DEVCONF\_FORWARDING] = 0, [DEVCONF\_HOPLIMIT] = 64, [DEVCONF\_MTU6] = 1500, [DEVCONF\_ACCEPT\_RA] = 1, [DEVCONF\_ACCEPT\_REDIRECTS] = 1, [DEVCONF\_AUTOCONF] = 1, [DEVCONF\_DAD\_TRANSMITS] = 1, [DEVCONF\_RTR\_SOLICITS] = -1, [DEVCONF\_RTR\_SOLICIT\_INTERVAL] = 4000, [DEVCONF\_RTR\_SOLICIT\_DELAY] = 1000, [DEVCONF\_USE\_TEMPADDR] = 0, [DEVCONF\_TEMP\_VALID\_LFT] = 604800, [DEVCONF\_TEMP\_PREFERED\_LFT] = 86400, [DEVCONF\_REGEN\_MAX\_RETRY] = 3, [DEVCONF\_MAX\_DESYNC\_FACTOR] = 600, [DEVCONF\_MAX\_ADDRESSES] = 16, [DEVCONF\_FORCE\_MLD\_VERSION] = 0, [DEVCONF\_ACCEPT\_RA\_DEFRTR] = 1, [DEVCONF\_ACCEPT\_RA\_PINFO] = 1, [DEVCONF\_ACCEPT\_RA\_RTR\_PREF] = 0, [DEVCONF\_RTR\_PROBE\_INTERVAL] = 0, [DEVCONF\_ACCEPT\_RA\_RT\_INFO\_MAX\_PLEN] = 0, [DEVCONF\_PROXY\_NDP] = 0, [DEVCONF\_OPTIMISTIC\_DAD] = 0, [DEVCONF\_ACCEPT\_SOURCE\_ROUTE] = 0, [DEVCONF\_MC\_FORWARDING] = 0, [DEVCONF\_DISABLE\_IPV6] = 0, [DEVCONF\_ACCEPT\_DAD] = 1, [DEVCONF\_FORCE\_TLLAO] = 0, [DEVCONF\_NDISC\_NOTIFY] = 0, [DEVCONF\_MLDV1\_UNSOLICITED\_REPORT\_INTERVAL] = 10000, [DEVCONF\_MLDV2\_UNSOLICITED\_REPORT\_INTERVAL] = 1000, ...]}, {{nla\_len=300, nla\_type=IFLA\_INET6\_STATS}, [[IPSTATS\_MIB\_NUM] = 37, [IPSTATS\_MIB\_INPKTS] = 0, [IPSTATS\_MIB\_INOCTETS] = 0, [IPSTATS\_MIB\_INDELIVERS] = 0, [IPSTATS\_MIB\_OUTFORWDATAGRAMS] = 0, [IPSTATS\_MIB\_OUTPKTS] = 0, [IPSTATS\_MIB\_OUTOCTETS] = 0, [IPSTATS\_MIB\_INHDRERRORS] = 0, [IPSTATS\_MIB\_INTOOBIGERRORS] = 0, [IPSTATS\_MIB\_INNOROUTES] = 0, [IPSTATS\_MIB\_INADDRERRORS] = 0, [IPSTATS\_MIB\_INUNKNOWNPROTOS] = 0, [IPSTATS\_MIB\_INTRUNCATEDPKTS] = 0, [IPSTATS\_MIB\_INDISCARDS] = 0, [IPSTATS\_MIB\_OUTDISCARDS] = 0, [IPSTATS\_MIB\_OUTNOROUTES] = 0, [IPSTATS\_MIB\_REASMTIMEOUT] = 0, [IPSTATS\_MIB\_REASMREQDS] = 0, [IPSTATS\_MIB\_REASMOKS] = 0, [IPSTATS\_MIB\_REASMFAILS] = 0, [IPSTATS\_MIB\_FRAGOKS] = 0, [IPSTATS\_MIB\_FRAGFAILS] = 0, [IPSTATS\_MIB\_FRAGCREATES] = 0, [IPSTATS\_MIB\_INMCASTPKTS] = 0, [IPSTATS\_MIB\_OUTMCASTPKTS] = 0, [IPSTATS\_MIB\_INBCASTPKTS] = 0, [IPSTATS\_MIB\_OUTBCASTPKTS] = 0, [IPSTATS\_MIB\_INMCASTOCTETS] = 0, [IPSTATS\_MIB\_OUTMCASTOCTETS] = 0, [IPSTATS\_MIB\_INBCASTOCTETS] = 0, [IPSTATS\_MIB\_OUTBCASTOCTETS] = 0, [IPSTATS\_MIB\_CSUMERRORS] = 0, ...]}, {{nla\_len=52, nla\_type=IFLA\_INET6\_ICMP6STATS}, [[ICMP6\_MIB\_NUM] = 6, [ICMP6\_MIB\_INMSGS] = 0, [ICMP6\_MIB\_INERRORS] = 0, [ICMP6\_MIB\_OUTMSGS] = 0, [ICMP6\_MIB\_OUTERRORS] = 0, [ICMP6\_MIB\_CSUMERRORS] = 0]}, {{nla\_len=20, nla\_type=IFLA\_INET6\_TOKEN}, inet\_pton(AF\_INET6, "::")}, {{nla\_len=5, nla\_type=IFLA\_INET6\_ADDR\_GEN\_MODE}, IN6\_ADDR\_GEN\_MODE\_EUI64}]}]}]}], iov\_len=4096}], msg\_iovlen=1, msg\_controllen=0, msg\_flags=0}, 0) = 2824

recvmsg(9, {msg\_name={sa\_family=AF\_NETLINK, nl\_pid=0, nl\_groups=00000000}, msg\_namelen=12, msg\_iov=[{iov\_base=[{{len=1336, type=RTM\_NEWLINK, flags=NLM\_F\_MULTI, seq=1618838898, pid=212}, {ifi\_family=AF\_UNSPEC, ifi\_type=ARPHRD\_ETHER, ifi\_index=if\_nametoindex("dummy0"), ifi\_flags=IFF\_BROADCAST|IFF\_NOARP, ifi\_change=0}, [{{nla\_len=11, nla\_type=IFLA\_IFNAME}, "dummy0"}, {{nla\_len=8, nla\_type=IFLA\_TXQLEN}, 1000}, {{nla\_len=5, nla\_type=IFLA\_OPERSTATE}, 2}, {{nla\_len=5, nla\_type=IFLA\_LINKMODE}, 0}, {{nla\_len=8, nla\_type=IFLA\_MTU}, 1500}, {{nla\_len=8, nla\_type=IFLA\_MIN\_MTU}, 0}, {{nla\_len=8, nla\_type=IFLA\_MAX\_MTU}, 0}, {{nla\_len=8, nla\_type=IFLA\_GROUP}, 0}, {{nla\_len=8, nla\_type=IFLA\_PROMISCUITY}, 0}, {{nla\_len=8, nla\_type=IFLA\_NUM\_TX\_QUEUES}, 1}, {{nla\_len=8, nla\_type=IFLA\_GSO\_MAX\_SEGS}, 65535}, {{nla\_len=8, nla\_type=IFLA\_GSO\_MAX\_SIZE}, 65536}, {{nla\_len=8, nla\_type=IFLA\_NUM\_RX\_QUEUES}, 1}, {{nla\_len=5, nla\_type=IFLA\_CARRIER}, 1}, {{nla\_len=9, nla\_type=IFLA\_QDISC}, "noop"}, {{nla\_len=8, nla\_type=IFLA\_CARRIER\_CHANGES}, 0}, {{nla\_len=5, nla\_type=IFLA\_PROTO\_DOWN}, 0}, {{nla\_len=8, nla\_type=IFLA\_CARRIER\_UP\_COUNT}, 0}, {{nla\_len=8, nla\_type=IFLA\_CARRIER\_DOWN\_COUNT}, 0}, {{nla\_len=36, nla\_type=IFLA\_MAP}, {mem\_start=0, mem\_end=0, base\_addr=0, irq=0, dma=0, port=0}}, {{nla\_len=10, nla\_type=IFLA\_ADDRESS}, "\xae\x09\x51\x72\x73\xbf"}, {{nla\_len=10, nla\_type=IFLA\_BROADCAST}, "\xff\xff\xff\xff\xff\xff"}, {{nla\_len=196, nla\_type=IFLA\_STATS64}, {rx\_packets=0, tx\_packets=0, rx\_bytes=0, tx\_bytes=0, rx\_errors=0, tx\_errors=0, rx\_dropped=0, tx\_dropped=0, multicast=0, collisions=0, rx\_length\_errors=0, rx\_over\_errors=0, rx\_crc\_errors=0, rx\_frame\_errors=0, rx\_fifo\_errors=0, rx\_missed\_errors=0, tx\_aborted\_errors=0, tx\_carrier\_errors=0, tx\_fifo\_errors=0, tx\_heartbeat\_errors=0, tx\_window\_errors=0, rx\_compressed=0, tx\_compressed=0, rx\_nohandler=0}}, {{nla\_len=100, nla\_type=IFLA\_STATS}, {rx\_packets=0, tx\_packets=0, rx\_bytes=0, tx\_bytes=0, rx\_errors=0, tx\_errors=0, rx\_dropped=0, tx\_dropped=0, multicast=0, collisions=0, rx\_length\_errors=0, rx\_over\_errors=0, rx\_crc\_errors=0, rx\_frame\_errors=0, rx\_fifo\_errors=0, rx\_missed\_errors=0, tx\_aborted\_errors=0, tx\_carrier\_errors=0, tx\_fifo\_errors=0, tx\_heartbeat\_errors=0, tx\_window\_errors=0, rx\_compressed=0, tx\_compressed=0, rx\_nohandler=0}}, {{nla\_len=12, nla\_type=IFLA\_XDP}, {{nla\_len=5, nla\_type=IFLA\_XDP\_ATTACHED}, XDP\_ATTACHED\_NONE}}, {{nla\_len=16, nla\_type=IFLA\_LINKINFO}, {{nla\_len=10, nla\_type=IFLA\_INFO\_KIND}, "dummy"}}, {{nla\_len=760, nla\_type=IFLA\_AF\_SPEC}, [{{nla\_len=136, nla\_type=AF\_INET}, {{nla\_len=132, nla\_type=IFLA\_INET\_CONF}, [[IPV4\_DEVCONF\_FORWARDING-1] = 0, [IPV4\_DEVCONF\_MC\_FORWARDING-1] = 0, [IPV4\_DEVCONF\_PROXY\_ARP-1] = 0, [IPV4\_DEVCONF\_ACCEPT\_REDIRECTS-1] = 1, [IPV4\_DEVCONF\_SECURE\_REDIRECTS-1] = 1, [IPV4\_DEVCONF\_SEND\_REDIRECTS-1] = 1, [IPV4\_DEVCONF\_SHARED\_MEDIA-1] = 1, [IPV4\_DEVCONF\_RP\_FILTER-1] = 0, [IPV4\_DEVCONF\_ACCEPT\_SOURCE\_ROUTE-1] = 1, [IPV4\_DEVCONF\_BOOTP\_RELAY-1] = 0, [IPV4\_DEVCONF\_LOG\_MARTIANS-1] = 0, [IPV4\_DEVCONF\_TAG-1] = 0, [IPV4\_DEVCONF\_ARPFILTER-1] = 0, [IPV4\_DEVCONF\_MEDIUM\_ID-1] = 0, [IPV4\_DEVCONF\_NOXFRM-1] = 0, [IPV4\_DEVCONF\_NOPOLICY-1] = 0, [IPV4\_DEVCONF\_FORCE\_IGMP\_VERSION-1] = 0, [IPV4\_DEVCONF\_ARP\_ANNOUNCE-1] = 0, [IPV4\_DEVCONF\_ARP\_IGNORE-1] = 0, [IPV4\_DEVCONF\_PROMOTE\_SECONDARIES-1] = 0, [IPV4\_DEVCONF\_ARP\_ACCEPT-1] = 0, [IPV4\_DEVCONF\_ARP\_NOTIFY-1] = 0, [IPV4\_DEVCONF\_ACCEPT\_LOCAL-1] = 0, [IPV4\_DEVCONF\_SRC\_VMARK-1] = 0, [IPV4\_DEVCONF\_PROXY\_ARP\_PVLAN-1] = 0, [IPV4\_DEVCONF\_ROUTE\_LOCALNET-1] = 0, [IPV4\_DEVCONF\_IGMPV2\_UNSOLICITED\_REPORT\_INTERVAL-1] = 10000, [IPV4\_DEVCONF\_IGMPV3\_UNSOLICITED\_REPORT\_INTERVAL-1] = 1000, [IPV4\_DEVCONF\_IGNORE\_ROUTES\_WITH\_LINKDOWN-1] = 0, [IPV4\_DEVCONF\_DROP\_UNICAST\_IN\_L2\_MULTICAST-1] = 0, [IPV4\_DEVCONF\_DROP\_GRATUITOUS\_ARP-1] = 0, [IPV4\_DEVCONF\_BC\_FORWARDING-1] = 0]}}, {{nla\_len=620, nla\_type=AF\_INET6}, [{{nla\_len=8, nla\_type=IFLA\_INET6\_FLAGS}, 0}, {{nla\_len=20, nla\_type=IFLA\_INET6\_CACHEINFO}, {max\_reasm\_len=65535, tstamp=29, reachable\_time=42740, retrans\_time=1000}}, {{nla\_len=208, nla\_type=IFLA\_INET6\_CONF}, [[DEVCONF\_FORWARDING] = 0, [DEVCONF\_HOPLIMIT] = 64, [DEVCONF\_MTU6] = 1500, [DEVCONF\_ACCEPT\_RA] = 1, [DEVCONF\_ACCEPT\_REDIRECTS] = 1, [DEVCONF\_AUTOCONF] = 1, [DEVCONF\_DAD\_TRANSMITS] = 1, [DEVCONF\_RTR\_SOLICITS] = -1, [DEVCONF\_RTR\_SOLICIT\_INTERVAL] = 4000, [DEVCONF\_RTR\_SOLICIT\_DELAY] = 1000, [DEVCONF\_USE\_TEMPADDR] = 0, [DEVCONF\_TEMP\_VALID\_LFT] = 604800, [DEVCONF\_TEMP\_PREFERED\_LFT] = 86400, [DEVCONF\_REGEN\_MAX\_RETRY] = 3, [DEVCONF\_MAX\_DESYNC\_FACTOR] = 600, [DEVCONF\_MAX\_ADDRESSES] = 16, [DEVCONF\_FORCE\_MLD\_VERSION] = 0, [DEVCONF\_ACCEPT\_RA\_DEFRTR] = 1, [DEVCONF\_ACCEPT\_RA\_PINFO] = 1, [DEVCONF\_ACCEPT\_RA\_RTR\_PREF] = 0, [DEVCONF\_RTR\_PROBE\_INTERVAL] = 0, [DEVCONF\_ACCEPT\_RA\_RT\_INFO\_MAX\_PLEN] = 0, [DEVCONF\_PROXY\_NDP] = 0, [DEVCONF\_OPTIMISTIC\_DAD] = 0, [DEVCONF\_ACCEPT\_SOURCE\_ROUTE] = 0, [DEVCONF\_MC\_FORWARDING] = 0, [DEVCONF\_DISABLE\_IPV6] = 0, [DEVCONF\_ACCEPT\_DAD] = -1, [DEVCONF\_FORCE\_TLLAO] = 0, [DEVCONF\_NDISC\_NOTIFY] = 0, [DEVCONF\_MLDV1\_UNSOLICITED\_REPORT\_INTERVAL] = 10000, [DEVCONF\_MLDV2\_UNSOLICITED\_REPORT\_INTERVAL] = 1000, ...]}, {{nla\_len=300, nla\_type=IFLA\_INET6\_STATS}, [[IPSTATS\_MIB\_NUM] = 37, [IPSTATS\_MIB\_INPKTS] = 0, [IPSTATS\_MIB\_INOCTETS] = 0, [IPSTATS\_MIB\_INDELIVERS] = 0, [IPSTATS\_MIB\_OUTFORWDATAGRAMS] = 0, [IPSTATS\_MIB\_OUTPKTS] = 0, [IPSTATS\_MIB\_OUTOCTETS] = 0, [IPSTATS\_MIB\_INHDRERRORS] = 0, [IPSTATS\_MIB\_INTOOBIGERRORS] = 0, [IPSTATS\_MIB\_INNOROUTES] = 0, [IPSTATS\_MIB\_INADDRERRORS] = 0, [IPSTATS\_MIB\_INUNKNOWNPROTOS] = 0, [IPSTATS\_MIB\_INTRUNCATEDPKTS] = 0, [IPSTATS\_MIB\_INDISCARDS] = 0, [IPSTATS\_MIB\_OUTDISCARDS] = 0, [IPSTATS\_MIB\_OUTNOROUTES] = 0, [IPSTATS\_MIB\_REASMTIMEOUT] = 0, [IPSTATS\_MIB\_REASMREQDS] = 0, [IPSTATS\_MIB\_REASMOKS] = 0, [IPSTATS\_MIB\_REASMFAILS] = 0, [IPSTATS\_MIB\_FRAGOKS] = 0, [IPSTATS\_MIB\_FRAGFAILS] = 0, [IPSTATS\_MIB\_FRAGCREATES] = 0, [IPSTATS\_MIB\_INMCASTPKTS] = 0, [IPSTATS\_MIB\_OUTMCASTPKTS] = 0, [IPSTATS\_MIB\_INBCASTPKTS] = 0, [IPSTATS\_MIB\_OUTBCASTPKTS] = 0, [IPSTATS\_MIB\_INMCASTOCTETS] = 0, [IPSTATS\_MIB\_OUTMCASTOCTETS] = 0, [IPSTATS\_MIB\_INBCASTOCTETS] = 0, [IPSTATS\_MIB\_OUTBCASTOCTETS] = 0, [IPSTATS\_MIB\_CSUMERRORS] = 0, ...]}, {{nla\_len=52, nla\_type=IFLA\_INET6\_ICMP6STATS}, [[ICMP6\_MIB\_NUM] = 6, [ICMP6\_MIB\_INMSGS] = 0, [ICMP6\_MIB\_INERRORS] = 0, [ICMP6\_MIB\_OUTMSGS] = 0, [ICMP6\_MIB\_OUTERRORS] = 0, [ICMP6\_MIB\_CSUMERRORS] = 0]}, {{nla\_len=20, nla\_type=IFLA\_INET6\_TOKEN}, inet\_pton(AF\_INET6, "::")}, {{nla\_len=5, nla\_type=IFLA\_INET6\_ADDR\_GEN\_MODE}, IN6\_ADDR\_GEN\_MODE\_EUI64}]}]}]}, {{len=1316, type=RTM\_NEWLINK, flags=NLM\_F\_MULTI, seq=1618838898, pid=212}, {ifi\_family=AF\_UNSPEC, ifi\_type=ARPHRD\_ETHER, ifi\_index=if\_nametoindex("eth0"), ifi\_flags=IFF\_UP|IFF\_BROADCAST|IFF\_RUNNING|IFF\_MULTICAST|IFF\_LOWER\_UP, ifi\_change=0}, [{{nla\_len=9, nla\_type=IFLA\_IFNAME}, "eth0"}, {{nla\_len=8, nla\_type=IFLA\_TXQLEN}, 1000}, {{nla\_len=5, nla\_type=IFLA\_OPERSTATE}, 6}, {{nla\_len=5, nla\_type=IFLA\_LINKMODE}, 0}, {{nla\_len=8, nla\_type=IFLA\_MTU}, 1500}, {{nla\_len=8, nla\_type=IFLA\_MIN\_MTU}, 68}, {{nla\_len=8, nla\_type=IFLA\_MAX\_MTU}, 65521}, {{nla\_len=8, nla\_type=IFLA\_GROUP}, 0}, {{nla\_len=8, nla\_type=IFLA\_PROMISCUITY}, 0}, {{nla\_len=8, nla\_type=IFLA\_NUM\_TX\_QUEUES}, 64}, {{nla\_len=8, nla\_type=IFLA\_GSO\_MAX\_SEGS}, 65535}, {{nla\_len=8, nla\_type=IFLA\_GSO\_MAX\_SIZE}, 62780}, {{nla\_len=8, nla\_type=IFLA\_NUM\_RX\_QUEUES}, 64}, {{nla\_len=5, nla\_type=IFLA\_CARRIER}, 1}, {{nla\_len=7, nla\_type=IFLA\_QDISC}, "mq"}, {{nla\_len=8, nla\_type=IFLA\_CARRIER\_CHANGES}, 1}, {{nla\_len=5, nla\_type=IFLA\_PROTO\_DOWN}, 0}, {{nla\_len=8, nla\_type=IFLA\_CARRIER\_UP\_COUNT}, 1}, {{nla\_len=8, nla\_type=IFLA\_CARRIER\_DOWN\_COUNT}, 0}, {{nla\_len=36, nla\_type=IFLA\_MAP}, {mem\_start=0, mem\_end=0, base\_addr=0, irq=0, dma=0, port=0}}, {{nla\_len=10, nla\_type=IFLA\_ADDRESS}, "\x00\x15\x5d\x4f\xb6\x5c"}, {{nla\_len=10, nla\_type=IFLA\_BROADCAST}, "\xff\xff\xff\xff\xff\xff"}, {{nla\_len=196, nla\_type=IFLA\_STATS64}, {rx\_packets=1271, tx\_packets=1061, rx\_bytes=266359, tx\_bytes=253301, rx\_errors=0, tx\_errors=0, rx\_dropped=0, tx\_dropped=0, multicast=21, collisions=0, rx\_length\_errors=0, rx\_over\_errors=0, rx\_crc\_errors=0, rx\_frame\_errors=0, rx\_fifo\_errors=0, rx\_missed\_errors=0, tx\_aborted\_errors=0, tx\_carrier\_errors=0, tx\_fifo\_errors=0, tx\_heartbeat\_errors=0, tx\_window\_errors=0, rx\_compressed=0, tx\_compressed=0, rx\_nohandler=0}}, {{nla\_len=100, nla\_type=IFLA\_STATS}, {rx\_packets=1271, tx\_packets=1061, rx\_bytes=266359, tx\_bytes=253301, rx\_errors=0, tx\_errors=0, rx\_dropped=0, tx\_dropped=0, multicast=21, collisions=0, rx\_length\_errors=0, rx\_over\_errors=0, rx\_crc\_errors=0, rx\_frame\_errors=0, rx\_fifo\_errors=0, rx\_missed\_errors=0, tx\_aborted\_errors=0, tx\_carrier\_errors=0, tx\_fifo\_errors=0, tx\_heartbeat\_errors=0, tx\_window\_errors=0, rx\_compressed=0, tx\_compressed=0, rx\_nohandler=0}}, {{nla\_len=12, nla\_type=IFLA\_XDP}, {{nla\_len=5, nla\_type=IFLA\_XDP\_ATTACHED}, XDP\_ATTACHED\_NONE}}, {{nla\_len=760, nla\_type=IFLA\_AF\_SPEC}, [{{nla\_len=136, nla\_type=AF\_INET}, {{nla\_len=132, nla\_type=IFLA\_INET\_CONF}, [[IPV4\_DEVCONF\_FORWARDING-1] = 0, [IPV4\_DEVCONF\_MC\_FORWARDING-1] = 0, [IPV4\_DEVCONF\_PROXY\_ARP-1] = 0, [IPV4\_DEVCONF\_ACCEPT\_REDIRECTS-1] = 1, [IPV4\_DEVCONF\_SECURE\_REDIRECTS-1] = 1, [IPV4\_DEVCONF\_SEND\_REDIRECTS-1] = 1, [IPV4\_DEVCONF\_SHARED\_MEDIA-1] = 1, [IPV4\_DEVCONF\_RP\_FILTER-1] = 0, [IPV4\_DEVCONF\_ACCEPT\_SOURCE\_ROUTE-1] = 1, [IPV4\_DEVCONF\_BOOTP\_RELAY-1] = 0, [IPV4\_DEVCONF\_LOG\_MARTIANS-1] = 0, [IPV4\_DEVCONF\_TAG-1] = 0, [IPV4\_DEVCONF\_ARPFILTER-1] = 0, [IPV4\_DEVCONF\_MEDIUM\_ID-1] = 0, [IPV4\_DEVCONF\_NOXFRM-1] = 0, [IPV4\_DEVCONF\_NOPOLICY-1] = 0, [IPV4\_DEVCONF\_FORCE\_IGMP\_VERSION-1] = 0, [IPV4\_DEVCONF\_ARP\_ANNOUNCE-1] = 0, [IPV4\_DEVCONF\_ARP\_IGNORE-1] = 0, [IPV4\_DEVCONF\_PROMOTE\_SECONDARIES-1] = 0, [IPV4\_DEVCONF\_ARP\_ACCEPT-1] = 0, [IPV4\_DEVCONF\_ARP\_NOTIFY-1] = 0, [IPV4\_DEVCONF\_ACCEPT\_LOCAL-1] = 0, [IPV4\_DEVCONF\_SRC\_VMARK-1] = 0, [IPV4\_DEVCONF\_PROXY\_ARP\_PVLAN-1] = 0, [IPV4\_DEVCONF\_ROUTE\_LOCALNET-1] = 0, [IPV4\_DEVCONF\_IGMPV2\_UNSOLICITED\_REPORT\_INTERVAL-1] = 10000, [IPV4\_DEVCONF\_IGMPV3\_UNSOLICITED\_REPORT\_INTERVAL-1] = 1000, [IPV4\_DEVCONF\_IGNORE\_ROUTES\_WITH\_LINKDOWN-1] = 0, [IPV4\_DEVCONF\_DROP\_UNICAST\_IN\_L2\_MULTICAST-1] = 0, [IPV4\_DEVCONF\_DROP\_GRATUITOUS\_ARP-1] = 0, [IPV4\_DEVCONF\_BC\_FORWARDING-1] = 0]}}, {{nla\_len=620, nla\_type=AF\_INET6}, [{{nla\_len=8, nla\_type=IFLA\_INET6\_FLAGS}, IF\_RS\_SENT|IF\_READY}, {{nla\_len=20, nla\_type=IFLA\_INET6\_CACHEINFO}, {max\_reasm\_len=65535, tstamp=126, reachable\_time=39500, retrans\_time=1000}}, {{nla\_len=208, nla\_type=IFLA\_INET6\_CONF}, [[DEVCONF\_FORWARDING] = 0, [DEVCONF\_HOPLIMIT] = 64, [DEVCONF\_MTU6] = 1500, [DEVCONF\_ACCEPT\_RA] = 1, [DEVCONF\_ACCEPT\_REDIRECTS] = 1, [DEVCONF\_AUTOCONF] = 1, [DEVCONF\_DAD\_TRANSMITS] = 1, [DEVCONF\_RTR\_SOLICITS] = -1, [DEVCONF\_RTR\_SOLICIT\_INTERVAL] = 4000, [DEVCONF\_RTR\_SOLICIT\_DELAY] = 1000, [DEVCONF\_USE\_TEMPADDR] = 0, [DEVCONF\_TEMP\_VALID\_LFT] = 604800, [DEVCONF\_TEMP\_PREFERED\_LFT] = 86400, [DEVCONF\_REGEN\_MAX\_RETRY] = 3, [DEVCONF\_MAX\_DESYNC\_FACTOR] = 600, [DEVCONF\_MAX\_ADDRESSES] = 16, [DEVCONF\_FORCE\_MLD\_VERSION] = 0, [DEVCONF\_ACCEPT\_RA\_DEFRTR] = 1, [DEVCONF\_ACCEPT\_RA\_PINFO] = 1, [DEVCONF\_ACCEPT\_RA\_RTR\_PREF] = 0, [DEVCONF\_RTR\_PROBE\_INTERVAL] = 0, [DEVCONF\_ACCEPT\_RA\_RT\_INFO\_MAX\_PLEN] = 0, [DEVCONF\_PROXY\_NDP] = 0, [DEVCONF\_OPTIMISTIC\_DAD] = 0, [DEVCONF\_ACCEPT\_SOURCE\_ROUTE] = 0, [DEVCONF\_MC\_FORWARDING] = 0, [DEVCONF\_DISABLE\_IPV6] = 0, [DEVCONF\_ACCEPT\_DAD] = 1, [DEVCONF\_FORCE\_TLLAO] = 0, [DEVCONF\_NDISC\_NOTIFY] = 0, [DEVCONF\_MLDV1\_UNSOLICITED\_REPORT\_INTERVAL] = 10000, [DEVCONF\_MLDV2\_UNSOLICITED\_REPORT\_INTERVAL] = 1000, ...]}, {{nla\_len=300, nla\_type=IFLA\_INET6\_STATS}, [[IPSTATS\_MIB\_NUM] = 37, [IPSTATS\_MIB\_INPKTS] = 4, [IPSTATS\_MIB\_INOCTETS] = 364, [IPSTATS\_MIB\_INDELIVERS] = 0, [IPSTATS\_MIB\_OUTFORWDATAGRAMS] = 0, [IPSTATS\_MIB\_OUTPKTS] = 11, [IPSTATS\_MIB\_OUTOCTETS] = 712, [IPSTATS\_MIB\_INHDRERRORS] = 0, [IPSTATS\_MIB\_INTOOBIGERRORS] = 0, [IPSTATS\_MIB\_INNOROUTES] = 0, [IPSTATS\_MIB\_INADDRERRORS] = 0, [IPSTATS\_MIB\_INUNKNOWNPROTOS] = 0, [IPSTATS\_MIB\_INTRUNCATEDPKTS] = 0, [IPSTATS\_MIB\_INDISCARDS] = 0, [IPSTATS\_MIB\_OUTDISCARDS] = 0, [IPSTATS\_MIB\_OUTNOROUTES] = 0, [IPSTATS\_MIB\_REASMTIMEOUT] = 0, [IPSTATS\_MIB\_REASMREQDS] = 0, [IPSTATS\_MIB\_REASMOKS] = 0, [IPSTATS\_MIB\_REASMFAILS] = 0, [IPSTATS\_MIB\_FRAGOKS] = 0, [IPSTATS\_MIB\_FRAGFAILS] = 0, [IPSTATS\_MIB\_FRAGCREATES] = 0, [IPSTATS\_MIB\_INMCASTPKTS] = 4, [IPSTATS\_MIB\_OUTMCASTPKTS] = 11, [IPSTATS\_MIB\_INBCASTPKTS] = 0, [IPSTATS\_MIB\_OUTBCASTPKTS] = 0, [IPSTATS\_MIB\_INMCASTOCTETS] = 364, [IPSTATS\_MIB\_OUTMCASTOCTETS] = 712, [IPSTATS\_MIB\_INBCASTOCTETS] = 0, [IPSTATS\_MIB\_OUTBCASTOCTETS] = 0, [IPSTATS\_MIB\_CSUMERRORS] = 0, ...]}, {{nla\_len=52, nla\_type=IFLA\_INET6\_ICMP6STATS}, [[ICMP6\_MIB\_NUM] = 6, [ICMP6\_MIB\_INMSGS] = 0, [ICMP6\_MIB\_INERRORS] = 0, [ICMP6\_MIB\_OUTMSGS] = 11, [ICMP6\_MIB\_OUTERRORS] = 0, [ICMP6\_MIB\_CSUMERRORS] = 0]}, {{nla\_len=20, nla\_type=IFLA\_INET6\_TOKEN}, inet\_pton(AF\_INET6, "::")}, {{nla\_len=5, nla\_type=IFLA\_INET6\_ADDR\_GEN\_MODE}, IN6\_ADDR\_GEN\_MODE\_EUI64}]}]}]}, {{len=1440, type=RTM\_NEWLINK, flags=NLM\_F\_MULTI, seq=1618838898, pid=212}, {ifi\_family=AF\_UNSPEC, ifi\_type=ARPHRD\_SIT, ifi\_index=if\_nametoindex("sit0"), ifi\_flags=IFF\_NOARP, ifi\_change=0}, [{{nla\_len=9, nla\_type=IFLA\_IFNAME}, "sit0"}, {{nla\_len=8, nla\_type=IFLA\_TXQLEN}, 1000}, {{nla\_len=5, nla\_type=IFLA\_OPERSTATE}, 2}, {{nla\_len=5, nla\_type=IFLA\_LINKMODE}, 0}, {{nla\_len=8, nla\_type=IFLA\_MTU}, 1480}, {{nla\_len=8, nla\_type=IFLA\_MIN\_MTU}, 1280}, {{nla\_len=8, nla\_type=IFLA\_MAX\_MTU}, 65555}, {{nla\_len=8, nla\_type=IFLA\_GROUP}, 0}, {{nla\_len=8, nla\_type=IFLA\_PROMISCUITY}, 0}, {{nla\_len=8, nla\_type=IFLA\_NUM\_TX\_QUEUES}, 1}, {{nla\_len=8, nla\_type=IFLA\_GSO\_MAX\_SEGS}, 65535}, {{nla\_len=8, nla\_type=IFLA\_GSO\_MAX\_SIZE}, 65536}, {{nla\_len=8, nla\_type=IFLA\_NUM\_RX\_QUEUES}, 1}, {{nla\_len=5, nla\_type=IFLA\_CARRIER}, 1}, {{nla\_len=9, nla\_type=IFLA\_QDISC}, "noop"}, {{nla\_len=8, nla\_type=IFLA\_CARRIER\_CHANGES}, 0}, {{nla\_len=5, nla\_type=IFLA\_PROTO\_DOWN}, 0}, {{nla\_len=8, nla\_type=IFLA\_CARRIER\_UP\_COUNT}, 0}, {{nla\_len=8, nla\_type=IFLA\_CARRIER\_DOWN\_COUNT}, 0}, {{nla\_len=36, nla\_type=IFLA\_MAP}, {mem\_start=0, mem\_end=0, base\_addr=0, irq=0, dma=0, port=0}}, {{nla\_len=8, nla\_type=IFLA\_ADDRESS}, "\x00\x00\x00\x00"}, {{nla\_len=8, nla\_type=IFLA\_BROADCAST}, "\x00\x00\x00\x00"}, {{nla\_len=196, nla\_type=IFLA\_STATS64}, {rx\_packets=0, tx\_packets=0, rx\_bytes=0, tx\_bytes=0, rx\_errors=0, tx\_errors=0, rx\_dropped=0, tx\_dropped=0, multicast=0, collisions=0, rx\_length\_errors=0, rx\_over\_errors=0, rx\_crc\_errors=0, rx\_frame\_errors=0, rx\_fifo\_errors=0, rx\_missed\_errors=0, tx\_aborted\_errors=0, tx\_carrier\_errors=0, tx\_fifo\_errors=0, tx\_heartbeat\_errors=0, tx\_window\_errors=0, rx\_compressed=0, tx\_compressed=0, rx\_nohandler=0}}, {{nla\_len=100, nla\_type=IFLA\_STATS}, {rx\_packets=0, tx\_packets=0, rx\_bytes=0, tx\_bytes=0, rx\_errors=0, tx\_errors=0, rx\_dropped=0, tx\_dropped=0, multicast=0, collisions=0, rx\_length\_errors=0, rx\_over\_errors=0, rx\_crc\_errors=0, rx\_frame\_errors=0, rx\_fifo\_errors=0, rx\_missed\_errors=0, tx\_aborted\_errors=0, tx\_carrier\_errors=0, tx\_fifo\_errors=0, tx\_heartbeat\_errors=0, tx\_window\_errors=0, rx\_compressed=0, tx\_compressed=0, rx\_nohandler=0}}, {{nla\_len=12, nla\_type=IFLA\_XDP}, {{nla\_len=5, nla\_type=IFLA\_XDP\_ATTACHED}, XDP\_ATTACHED\_NONE}}, {{nla\_len=120, nla\_type=IFLA\_LINKINFO}, [{{nla\_len=8, nla\_type=IFLA\_INFO\_KIND}, "sit"}, {{nla\_len=108, nla\_type=IFLA\_INFO\_DATA}, "\x08\x00\x01\x00\x00\x00\x00\x00\x08\x00\x02\x00\x00\x00\x00\x00\x08\x00\x03\x00\x00\x00\x00\x00\x05\x00\x04\x00\x40\x00\x00\x00"...}]}, {{nla\_len=8, nla\_type=IFLA\_LINK}, 0}, {{nla\_len=760, nla\_type=IFLA\_AF\_SPEC}, [{{nla\_len=136, nla\_type=AF\_INET}, {{nla\_len=132, nla\_type=IFLA\_INET\_CONF}, [[IPV4\_DEVCONF\_FORWARDING-1] = 0, [IPV4\_DEVCONF\_MC\_FORWARDING-1] = 0, [IPV4\_DEVCONF\_PROXY\_ARP-1] = 0, [IPV4\_DEVCONF\_ACCEPT\_REDIRECTS-1] = 1, [IPV4\_DEVCONF\_SECURE\_REDIRECTS-1] = 1, [IPV4\_DEVCONF\_SEND\_REDIRECTS-1] = 1, [IPV4\_DEVCONF\_SHARED\_MEDIA-1] = 1, [IPV4\_DEVCONF\_RP\_FILTER-1] = 0, [IPV4\_DEVCONF\_ACCEPT\_SOURCE\_ROUTE-1] = 1, [IPV4\_DEVCONF\_BOOTP\_RELAY-1] = 0, [IPV4\_DEVCONF\_LOG\_MARTIANS-1] = 0, [IPV4\_DEVCONF\_TAG-1] = 0, [IPV4\_DEVCONF\_ARPFILTER-1] = 0, [IPV4\_DEVCONF\_MEDIUM\_ID-1] = 0, [IPV4\_DEVCONF\_NOXFRM-1] = 0, [IPV4\_DEVCONF\_NOPOLICY-1] = 0, [IPV4\_DEVCONF\_FORCE\_IGMP\_VERSION-1] = 0, [IPV4\_DEVCONF\_ARP\_ANNOUNCE-1] = 0, [IPV4\_DEVCONF\_ARP\_IGNORE-1] = 0, [IPV4\_DEVCONF\_PROMOTE\_SECONDARIES-1] = 0, [IPV4\_DEVCONF\_ARP\_ACCEPT-1] = 0, [IPV4\_DEVCONF\_ARP\_NOTIFY-1] = 0, [IPV4\_DEVCONF\_ACCEPT\_LOCAL-1] = 0, [IPV4\_DEVCONF\_SRC\_VMARK-1] = 0, [IPV4\_DEVCONF\_PROXY\_ARP\_PVLAN-1] = 0, [IPV4\_DEVCONF\_ROUTE\_LOCALNET-1] = 0, [IPV4\_DEVCONF\_IGMPV2\_UNSOLICITED\_REPORT\_INTERVAL-1] = 10000, [IPV4\_DEVCONF\_IGMPV3\_UNSOLICITED\_REPORT\_INTERVAL-1] = 1000, [IPV4\_DEVCONF\_IGNORE\_ROUTES\_WITH\_LINKDOWN-1] = 0, [IPV4\_DEVCONF\_DROP\_UNICAST\_IN\_L2\_MULTICAST-1] = 0, [IPV4\_DEVCONF\_DROP\_GRATUITOUS\_ARP-1] = 0, [IPV4\_DEVCONF\_BC\_FORWARDING-1] = 0]}}, {{nla\_len=620, nla\_type=AF\_INET6}, [{{nla\_len=8, nla\_type=IFLA\_INET6\_FLAGS}, 0}, {{nla\_len=20, nla\_type=IFLA\_INET6\_CACHEINFO}, {max\_reasm\_len=65535, tstamp=29, reachable\_time=20560, retrans\_time=1000}}, {{nla\_len=208, nla\_type=IFLA\_INET6\_CONF}, [[DEVCONF\_FORWARDING] = 0, [DEVCONF\_HOPLIMIT] = 64, [DEVCONF\_MTU6] = 1480, [DEVCONF\_ACCEPT\_RA] = 1, [DEVCONF\_ACCEPT\_REDIRECTS] = 1, [DEVCONF\_AUTOCONF] = 1, [DEVCONF\_DAD\_TRANSMITS] = 1, [DEVCONF\_RTR\_SOLICITS] = -1, [DEVCONF\_RTR\_SOLICIT\_INTERVAL] = 4000, [DEVCONF\_RTR\_SOLICIT\_DELAY] = 1000, [DEVCONF\_USE\_TEMPADDR] = -1, [DEVCONF\_TEMP\_VALID\_LFT] = 604800, [DEVCONF\_TEMP\_PREFERED\_LFT] = 86400, [DEVCONF\_REGEN\_MAX\_RETRY] = 3, [DEVCONF\_MAX\_DESYNC\_FACTOR] = 600, [DEVCONF\_MAX\_ADDRESSES] = 16, [DEVCONF\_FORCE\_MLD\_VERSION] = 0, [DEVCONF\_ACCEPT\_RA\_DEFRTR] = 1, [DEVCONF\_ACCEPT\_RA\_PINFO] = 1, [DEVCONF\_ACCEPT\_RA\_RTR\_PREF] = 0, [DEVCONF\_RTR\_PROBE\_INTERVAL] = 0, [DEVCONF\_ACCEPT\_RA\_RT\_INFO\_MAX\_PLEN] = 0, [DEVCONF\_PROXY\_NDP] = 0, [DEVCONF\_OPTIMISTIC\_DAD] = 0, [DEVCONF\_ACCEPT\_SOURCE\_ROUTE] = 0, [DEVCONF\_MC\_FORWARDING] = 0, [DEVCONF\_DISABLE\_IPV6] = 0, [DEVCONF\_ACCEPT\_DAD] = -1, [DEVCONF\_FORCE\_TLLAO] = 0, [DEVCONF\_NDISC\_NOTIFY] = 0, [DEVCONF\_MLDV1\_UNSOLICITED\_REPORT\_INTERVAL] = 10000, [DEVCONF\_MLDV2\_UNSOLICITED\_REPORT\_INTERVAL] = 1000, ...]}, {{nla\_len=300, nla\_type=IFLA\_INET6\_STATS}, [[IPSTATS\_MIB\_NUM] = 37, [IPSTATS\_MIB\_INPKTS] = 0, [IPSTATS\_MIB\_INOCTETS] = 0, [IPSTATS\_MIB\_INDELIVERS] = 0, [IPSTATS\_MIB\_OUTFORWDATAGRAMS] = 0, [IPSTATS\_MIB\_OUTPKTS] = 0, [IPSTATS\_MIB\_OUTOCTETS] = 0, [IPSTATS\_MIB\_INHDRERRORS] = 0, [IPSTATS\_MIB\_INTOOBIGERRORS] = 0, [IPSTATS\_MIB\_INNOROUTES] = 0, [IPSTATS\_MIB\_INADDRERRORS] = 0, [IPSTATS\_MIB\_INUNKNOWNPROTOS] = 0, [IPSTATS\_MIB\_INTRUNCATEDPKTS] = 0, [IPSTATS\_MIB\_INDISCARDS] = 0, [IPSTATS\_MIB\_OUTDISCARDS] = 0, [IPSTATS\_MIB\_OUTNOROUTES] = 0, [IPSTATS\_MIB\_REASMTIMEOUT] = 0, [IPSTATS\_MIB\_REASMREQDS] = 0, [IPSTATS\_MIB\_REASMOKS] = 0, [IPSTATS\_MIB\_REASMFAILS] = 0, [IPSTATS\_MIB\_FRAGOKS] = 0, [IPSTATS\_MIB\_FRAGFAILS] = 0, [IPSTATS\_MIB\_FRAGCREATES] = 0, [IPSTATS\_MIB\_INMCASTPKTS] = 0, [IPSTATS\_MIB\_OUTMCASTPKTS] = 0, [IPSTATS\_MIB\_INBCASTPKTS] = 0, [IPSTATS\_MIB\_OUTBCASTPKTS] = 0, [IPSTATS\_MIB\_INMCASTOCTETS] = 0, [IPSTATS\_MIB\_OUTMCASTOCTETS] = 0, [IPSTATS\_MIB\_INBCASTOCTETS] = 0, [IPSTATS\_MIB\_OUTBCASTOCTETS] = 0, [IPSTATS\_MIB\_CSUMERRORS] = 0, ...]}, {{nla\_len=52, nla\_type=IFLA\_INET6\_ICMP6STATS}, [[ICMP6\_MIB\_NUM] = 6, [ICMP6\_MIB\_INMSGS] = 0, [ICMP6\_MIB\_INERRORS] = 0, [ICMP6\_MIB\_OUTMSGS] = 0, [ICMP6\_MIB\_OUTERRORS] = 0, [ICMP6\_MIB\_CSUMERRORS] = 0]}, {{nla\_len=20, nla\_type=IFLA\_INET6\_TOKEN}, inet\_pton(AF\_INET6, "::")}, {{nla\_len=5, nla\_type=IFLA\_INET6\_ADDR\_GEN\_MODE}, IN6\_ADDR\_GEN\_MODE\_EUI64}]}]}]}], iov\_len=4096}], msg\_iovlen=1, msg\_controllen=0, msg\_flags=0}, 0) = 4092

recvmsg(9, {msg\_name={sa\_family=AF\_NETLINK, nl\_pid=0, nl\_groups=00000000}, msg\_namelen=12, msg\_iov=[{iov\_base={{len=20, type=NLMSG\_DONE, flags=NLM\_F\_MULTI, seq=1618838898, pid=212}, 0}, iov\_len=4096}], msg\_iovlen=1, msg\_controllen=0, msg\_flags=0}, 0) = 20

sendto(9, {{len=20, type=RTM\_GETADDR, flags=NLM\_F\_REQUEST|NLM\_F\_DUMP, seq=1618838899, pid=0}, {ifa\_family=AF\_UNSPEC, ...}}, 20, 0, {sa\_family=AF\_NETLINK, nl\_pid=0, nl\_groups=00000000}, 12) = 20

recvmsg(9, {msg\_name={sa\_family=AF\_NETLINK, nl\_pid=0, nl\_groups=00000000}, msg\_namelen=12, msg\_iov=[{iov\_base=[{{len=76, type=RTM\_NEWADDR, flags=NLM\_F\_MULTI, seq=1618838899, pid=212}, {ifa\_family=AF\_INET, ifa\_prefixlen=8, ifa\_flags=IFA\_F\_PERMANENT, ifa\_scope=RT\_SCOPE\_HOST, ifa\_index=if\_nametoindex("lo")}, [{{nla\_len=8, nla\_type=IFA\_ADDRESS}, inet\_addr("127.0.0.1")}, {{nla\_len=8, nla\_type=IFA\_LOCAL}, inet\_addr("127.0.0.1")}, {{nla\_len=7, nla\_type=IFA\_LABEL}, "lo"}, {{nla\_len=8, nla\_type=IFA\_FLAGS}, IFA\_F\_PERMANENT}, {{nla\_len=20, nla\_type=IFA\_CACHEINFO}, {ifa\_prefered=4294967295, ifa\_valid=4294967295, cstamp=62, tstamp=62}}]}, {{len=88, type=RTM\_NEWADDR, flags=NLM\_F\_MULTI, seq=1618838899, pid=212}, {ifa\_family=AF\_INET, ifa\_prefixlen=20, ifa\_flags=IFA\_F\_PERMANENT, ifa\_scope=RT\_SCOPE\_UNIVERSE, ifa\_index=if\_nametoindex("eth0")}, [{{nla\_len=8, nla\_type=IFA\_ADDRESS}, inet\_addr("172.25.231.165")}, {{nla\_len=8, nla\_type=IFA\_LOCAL}, inet\_addr("172.25.231.165")}, {{nla\_len=8, nla\_type=IFA\_BROADCAST}, inet\_addr("172.25.239.255")}, {{nla\_len=9, nla\_type=IFA\_LABEL}, "eth0"}, {{nla\_len=8, nla\_type=IFA\_FLAGS}, IFA\_F\_PERMANENT}, {{nla\_len=20, nla\_type=IFA\_CACHEINFO}, {ifa\_prefered=4294967295, ifa\_valid=4294967295, cstamp=62, tstamp=62}}]}], iov\_len=4096}], msg\_iovlen=1, msg\_controllen=0, msg\_flags=0}, 0) = 164

recvmsg(9, {msg\_name={sa\_family=AF\_NETLINK, nl\_pid=0, nl\_groups=00000000}, msg\_namelen=12, msg\_iov=[{iov\_base=[{{len=72, type=RTM\_NEWADDR, flags=NLM\_F\_MULTI, seq=1618838899, pid=212}, {ifa\_family=AF\_INET6, ifa\_prefixlen=128, ifa\_flags=IFA\_F\_PERMANENT, ifa\_scope=RT\_SCOPE\_HOST, ifa\_index=if\_nametoindex("lo")}, [{{nla\_len=20, nla\_type=IFA\_ADDRESS}, inet\_pton(AF\_INET6, "::1")}, {{nla\_len=20, nla\_type=IFA\_CACHEINFO}, {ifa\_prefered=4294967295, ifa\_valid=4294967295, cstamp=62, tstamp=62}}, {{nla\_len=8, nla\_type=IFA\_FLAGS}, IFA\_F\_PERMANENT}]}, {{len=72, type=RTM\_NEWADDR, flags=NLM\_F\_MULTI, seq=1618838899, pid=212}, {ifa\_family=AF\_INET6, ifa\_prefixlen=64, ifa\_flags=IFA\_F\_PERMANENT, ifa\_scope=RT\_SCOPE\_LINK, ifa\_index=if\_nametoindex("eth0")}, [{{nla\_len=20, nla\_type=IFA\_ADDRESS}, inet\_pton(AF\_INET6, "fe80::215:5dff:fe4f:b65c")}, {{nla\_len=20, nla\_type=IFA\_CACHEINFO}, {ifa\_prefered=4294967295, ifa\_valid=4294967295, cstamp=126, tstamp=126}}, {{nla\_len=8, nla\_type=IFA\_FLAGS}, IFA\_F\_PERMANENT}]}], iov\_len=4096}], msg\_iovlen=1, msg\_controllen=0, msg\_flags=0}, 0) = 144

recvmsg(9, {msg\_name={sa\_family=AF\_NETLINK, nl\_pid=0, nl\_groups=00000000}, msg\_namelen=12, msg\_iov=[{iov\_base={{len=20, type=NLMSG\_DONE, flags=NLM\_F\_MULTI, seq=1618838899, pid=212}, 0}, iov\_len=4096}], msg\_iovlen=1, msg\_controllen=0, msg\_flags=0}, 0) = 20

close(9) = 0

socket(AF\_INET, SOCK\_STREAM|SOCK\_CLOEXEC, IPPROTO\_TCP) = 9

setsockopt(9, SOL\_SOCKET, SO\_REUSEADDR, [1], 4) = 0

bind(9, {sa\_family=AF\_INET, sin\_port=htons(1234), sin\_addr=inet\_addr("127.0.0.1")}, 16) = 0

listen(9, 100) = 0

getsockname(9, {sa\_family=AF\_INET, sin\_port=htons(1234), sin\_addr=inet\_addr("127.0.0.1")}, [128->16]) = 0

getsockname(9, {sa\_family=AF\_INET, sin\_port=htons(1234), sin\_addr=inet\_addr("127.0.0.1")}, [128->16]) = 0

write(6, "\1\0\0\0\0\0\0\0", 8) = 8

write(8, "\1\0\0\0\0\0\0\0", 8) = 8

fstat(1, {st\_mode=S\_IFCHR|0620, st\_rdev=makedev(0x88, 0x1), ...}) = 0

write(1, "--------------------------------"..., 35) = 35

write(1, "create [id]\n", 12) = 12

write(1, "remove [id]\n", 12) = 12

write(1, "exec [id] [cmd - start/stop/time"..., 34) = 34

write(1, "heartbeat [time (ms)]\n", 22) = 22

write(1, "exit\n", 5) = 5

write(1, "--------------------------------"..., 35) = 35

fstat(0, {st\_mode=S\_IFCHR|0620, st\_rdev=makedev(0x88, 0x1), ...}) = 0

read(0, "create 3\n", 1024) = 9

clone(child\_stack=NULL, flags=CLONE\_CHILD\_CLEARTID|CLONE\_CHILD\_SETTID|SIGCHLD, child\_tidptr=0x7ff49ff818d0) = 215

poll([{fd=8, events=POLLIN}], 1, 0) = 1 ([{fd=8, revents=POLLIN}])

read(8, "\1\0\0\0\0\0\0\0", 8) = 8

poll([{fd=8, events=POLLIN}], 1, 0) = 0 (Timeout)

poll([{fd=8, events=POLLIN}], 1, 2000) = 1 ([{fd=8, revents=POLLIN}])

read(8, "\1\0\0\0\0\0\0\0", 8) = 8

poll([{fd=8, events=POLLIN}], 1, 0) = 0 (Timeout)

write(6, "\1\0\0\0\0\0\0\0", 8) = 8

poll([{fd=8, events=POLLIN}], 1, -1) = 1 ([{fd=8, revents=POLLIN}])

read(8, "\1\0\0\0\0\0\0\0", 8) = 8

poll([{fd=8, events=POLLIN}], 1, 0) = 0 (Timeout)

write(6, "\1\0\0\0\0\0\0\0", 8) = 8

write(1, "Ok:215\n", 7) = 7

read(0, "exit\n", 1024) = 5

rt\_sigaction(SIGINT, {sa\_handler=SIG\_IGN, sa\_mask=[], sa\_flags=SA\_RESTORER, sa\_restorer=0x7ff4a04dc210}, {sa\_handler=SIG\_DFL, sa\_mask=[], sa\_flags=0}, 8) = 0

rt\_sigaction(SIGQUIT, {sa\_handler=SIG\_IGN, sa\_mask=[], sa\_flags=SA\_RESTORER, sa\_restorer=0x7ff4a04dc210}, {sa\_handler=SIG\_DFL, sa\_mask=[], sa\_flags=0}, 8) = 0

rt\_sigprocmask(SIG\_BLOCK, [CHLD], [], 8) = 0

mmap(NULL, 36864, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS|MAP\_STACK, -1, 0) = 0x7ff4a092c000

rt\_sigprocmask(SIG\_BLOCK, ~[], [CHLD], 8) = 0

clone(child\_stack=0x7ff4a0934ff0, flags=CLONE\_VM|CLONE\_VFORK|SIGCHLD) = 220

munmap(0x7ff4a092c000, 36864) = 0

rt\_sigprocmask(SIG\_SETMASK, [CHLD], NULL, 8) = 0

wait4(220, [{WIFEXITED(s) && WEXITSTATUS(s) == 0}], 0, NULL) = 220

rt\_sigaction(SIGINT, {sa\_handler=SIG\_DFL, sa\_mask=[], sa\_flags=SA\_RESTORER, sa\_restorer=0x7ff4a04dc210}, NULL, 8) = 0

rt\_sigaction(SIGQUIT, {sa\_handler=SIG\_DFL, sa\_mask=[], sa\_flags=SA\_RESTORER, sa\_restorer=0x7ff4a04dc210}, NULL, 8) = 0

rt\_sigprocmask(SIG\_SETMASK, [], NULL, 8) = 0

--- SIGCHLD {si\_signo=SIGCHLD, si\_code=CLD\_EXITED, si\_pid=220, si\_uid=1000, si\_status=0, si\_utime=0, si\_stime=0} ---

write(4, "\1\0\0\0\0\0\0\0", 8) = 8

write(4, "\1\0\0\0\0\0\0\0", 8) = 8

poll([{fd=3, events=POLLIN}], 1, -1) = 1 ([{fd=3, revents=POLLIN}])

read(3, "\1\0\0\0\0\0\0\0", 8) = 8

write(6, "\1\0\0\0\0\0\0\0", 8) = 8

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 +++

1. **Вывод**

В данной лабораторной работе мы рассмотрели, как работает технология очереди сообщений.

Сокеты позволяют удобно организовывать построение и использование архитектуры клиент-сервер. Такие структуры, как деревья хорошо подходят для хранения информации о клиентах и сервере. Навыки отработанные в данной лаборатороной работы помогут в проектированнии мультипроцессорных программ.