Московский Авиационный Институт

(Национальный Исследовательский Университет)

Факультет информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

**Лабораторная работа №6 по курсу**

**«Операционные системы»**

Студент: Пономарев Н.В.

Группа: М80-207Б-20

Преподаватель: Миронов Е.С.

Оценка: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата: 19.12.21

**Содержание**

1. Постановка задачи.
2. Общие сведения о программе.
3. Общий метод и алгоритм решения.
4. Код программы.
5. Демонстрация работы программы.
6. Вывод.

## Постановка задачи

Реализовать распределенную систему по обработке запросов. В данной 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 сервера сообщений zmq. Также в данной системе необходимо предусмотреть проверку доступности узлов в соответствии с вариантом.

**Вариант задания:** 41. Топология — бинарное дерево. Тип вычислительной команды — сумма n чисел. Тип проверки узлов на доступность — пинг всех узлов.

# **Общие сведения о программе**

Программа состоит из двух файлов, которые компилируются в исполнительные файлы(которые представляют управляющий и вычислительные узлы). Общение между процессами происходит с помощью библиотеки zmq.

# **Общий метод и алгоритм решения**

* Управляющий узел принимает команды, обрабатывает их и пересылает дочерним узлам(или выводит сообщение об ошибке).
* Дочерние узлы проверяют, может ли быть команда выполнена в данном узле, если нет, то команда пересылается в один из дочерних узлов, из которого возвращается некоторое сообщение(об успехе или об ошибке), которое потом пересылается обратно по дереву.
* Для корректной проверки на доступность узлов, используется дерево, эмулирующее поведение узлов в данной топологии(например, при удалении узла, удаляются все его потомки).
* Если узел недоступен, то по истечении таймаута будет сгенерировано сообщение о недоступности узла и оно будет передано вверх по дереву, к управляющему узлу.
* При удалении узла, все его потомки рекурсивно уничтожаются.

## **Код программы**

**main\_prog.cpp:**

#include "zmq.hpp"

#include <sstream>

#include <string>

#include <iostream>

#include <zconf.h>

#include <vector>

#include <signal.h>

#include <sstream>

#include <set>

#include <algorithm>

// g++ main\_prog.cpp -lzmq -o main\_prog -w

using namespace std;

int main(){

zmq::context\_t context(1);

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

string adr = "tcp://127.0.0.1:300";

string command;

int child\_id = 0;

while(1){

cout << "Please, enter command**\n**";

cin >> command;

if(command == "create"){

if(child\_id == 0){

int id;

cin >> id;

int id\_tmp = id - 1;

while(1){

try {

main\_socket.bind(adr + to\_string(++id\_tmp));

break;

} catch(...) {

}

}

string new\_adr = adr + to\_string(id\_tmp);

char\* adr\_ = new char[new\_adr.size() + 1];

memcpy(adr\_, new\_adr.c\_str(), new\_adr.size() + 1);

char\* id\_ = new char[to\_string(id).size() + 1];

memcpy(id\_, to\_string(id).c\_str(), to\_string(id).size() + 1);

char\* args[] = {"./child\_node", adr\_, id\_, NULL};

int id2 = fork();

if (id2 == -1) {

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

id = 0;

exit(1);

} else if(id2 == 0){

execv("./child\_node", args);

} else {

child\_id = id;

}

zmq::message\_t message;

main\_socket.recv(&message);

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

cout << recieved\_message << "**\n**";

delete [] adr\_;

delete [] id\_;

} else {

int id;

cin >> id;

string message\_string = command + " " + to\_string(id);

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

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

main\_socket.send(message);

// catch message from new node

main\_socket.recv(&message);

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

cout << recieved\_message << "**\n**";

}

} else if(command == "exec"){

int id, value;

string name;

cin >> id >> name;

string s;

char q;

while((q = getchar()) != '**\n**'){

s += q;

}

if(s == ""){

string message\_string = command + " " + to\_string(id) + " " + name;

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

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

main\_socket.send(message);

// return value from map

main\_socket.recv(&message);

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

cout << recieved\_message << "**\n**";

} else {

value = stoi(s);

string message\_string = command + " " + to\_string(id) + " " + name + " " + to\_string(value);

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

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

main\_socket.send(message);

// add new element to map

main\_socket.recv(&message);

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

cout << recieved\_message << "**\n**";

}

} else if(command == "ping"){

int id;

cin >> id;

string message\_string = command + " " + to\_string(id);

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

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

main\_socket.send(message);

// receive answer from child

main\_socket.recv(&message);

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

cout << recieved\_message << "**\n**";

} else if(command == "kill"){

int id;

cin >> id;

if(child\_id == 0){

cout << "Error: there isn't nodes**\n**";

} else if(child\_id == id){

string kill\_message = command + " " + to\_string(id);

zmq::message\_t message(kill\_message.size());

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

main\_socket.send(message);

main\_socket.recv(message);

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

cout << received\_message << "**\n**";

cout << "Tree deleted successfully**\n**";

*/\*main\_socket.close();*

*context.close();\*/*

return 0;

} else {

string kill\_message = command + " " + to\_string(id);

zmq::message\_t message(kill\_message.size());

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

main\_socket.send(message);

main\_socket.recv(&message);

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

cout << received\_message << "**\n**";

}

} else if(command == "exit"){

main\_socket.close();

context.close();

break;

} else {

cout << "Error: incorrect command**\n**";

}

}

}

**child\_node.cpp:**

#include "zmq.hpp"

#include <sstream>

#include <string>

#include <iostream>

#include <zconf.h>

#include <vector>

#include <signal.h>

#include <fstream>

#include <algorithm>

#include <map>

// g++ child\_node\_2.cpp -lzmq -o child\_node -w

void send\_message(string message\_string, zmq::socket\_t& socket){

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

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

if(!socket.send(message\_back)){

cout << "Error: can't send message from node with pid " << getpid() << "**\n**";

}

}

using namespace std;

int main(int argc, char \* argv[]){

string adr = argv[1];

zmq::context\_t context(1);

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

main\_socket.connect(argv[1]);

send\_message("OK: " + to\_string(getpid()), main\_socket);

int id = stoi(argv[2]); // id of this node

map<string, int> m;

int left\_id = 0;

int right\_id = 0;

zmq::context\_t context\_l(1);

zmq::context\_t context\_r(1);

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

string adr\_left = "tcp://127.0.0.1:300";

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

string adr\_right = "tcp://127.0.0.1:300";

while(1){

zmq::message\_t message\_main;

main\_socket.recv(&message\_main);

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

string command;

for(int i = 0; i < recieved\_message.size(); ++i){

if(recieved\_message[i] != ' '){

command += recieved\_message[i];

} else {

break;

}

}

if(command == "exec"){

int id\_proc; // id of node for adding

string id\_proc\_, value\_;

string key;

int value;

for(int i = 5; i < recieved\_message.size(); ++i){

if(recieved\_message[i] != ' '){

id\_proc\_ += recieved\_message[i];

} else {

break;

}

}

id\_proc = stoi(id\_proc\_);

if(id\_proc == id){ // id == proc\_id

for(int i = 6 + id\_proc\_.size(); i < recieved\_message.size(); ++i){

if(recieved\_message[i] != ' '){

key += recieved\_message[i];

} else {

break;

}

}

for(int i = 7 + id\_proc\_.size() + key.size(); i < recieved\_message.size(); ++i){

if(recieved\_message[i] != ' ' || recieved\_message[i] != '**\n**'){

value\_ += recieved\_message[i];

} else {

break;

}

}

if(value\_ == ""){

if(m.count(key)){

int value\_map = m[key];

send\_message("OK:" + id\_proc\_ + ":" + to\_string(m[key]), main\_socket);

} else {

cout << key;

send\_message("OK:" + id\_proc\_ + ": **\'**" + key + "**\'** not found", main\_socket);

}

} else {

m[key] = stoi(value\_);

send\_message("OK:" + id\_proc\_, main\_socket);

}

} else {

if(id > id\_proc){

if(left\_id == 0){ // if node not exists

string message\_string = "Error:id: Not found";

send\_message("Error:id: Not found", main\_socket);

} else {

zmq::message\_t message(recieved\_message.size());

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

if(!left\_socket.send(message)){

cout << "Error: can't send message to left node from node with pid: " << getpid() << "**\n**";

}

// catch and send to parent

if(!left\_socket.recv(&message)){

cout << "Error: can't receive message from left node in node with pid: " << getpid() << "**\n**";

}

if(!main\_socket.send(message)){

cout << "Error: can't send message to main node from node with pid: " << getpid() << "**\n**";

}

}

} else {

if(right\_id == 0){ // if node not exists

string message\_string = "Error:id: Not found";

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

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

if(!main\_socket.send(message)){

cout << "Error: can't send message to main node from node with pid: " << getpid() << "**\n**";

}

} else {

zmq::message\_t message(recieved\_message.size());

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

if(!right\_socket.send(message)){

cout << "Error: can't send message to right node from node with pid: " << getpid() << "**\n**";

}

// catch and send to parent

if(!right\_socket.recv(&message)){

cout << "Error: can't receive message from left node in node with pid: " << getpid() << "**\n**";

}

if(!main\_socket.send(message)){

cout << "Error: can't send message to main node from node with pid: " << getpid() << "**\n**";

}

}

}

}

} else if(command == "create"){

ofstream o;

o.open("text.txt", std::ios::app);

o << "Create:" << left\_id << " " << id << " " << right\_id << "**\n**";

o.flush();

int id\_proc; // id of node for creating

string id\_proc\_;

for(int i = 7; i < recieved\_message.size(); ++i){

if(recieved\_message[i] != ' '){

id\_proc\_ += recieved\_message[i];

} else {

break;

}

}

id\_proc = stoi(id\_proc\_);

if(id\_proc == id){

send\_message("Error: Already exists", main\_socket);

} else if(id\_proc > id){

if(right\_id == 0){ // there is not right node

right\_id = id\_proc;

int right\_id\_tmp = right\_id - 1;

while(1){

try {

right\_socket.bind(adr\_right + to\_string(++right\_id\_tmp));

break;

} catch(...) {

}

}

adr\_right += to\_string(right\_id\_tmp);

char\* adr\_right\_ = new char[adr\_right.size() + 1];

memcpy(adr\_right\_, adr\_right.c\_str(), adr\_right.size() + 1);

char\* right\_id\_ = new char[to\_string(right\_id).size() + 1];

memcpy(right\_id\_, to\_string(right\_id).c\_str(), to\_string(right\_id).size() + 1);

char\* args[] = {"./child\_node", adr\_right\_, right\_id\_, NULL};

int f = fork();

if(f == 0){

execv("./child\_node", args);

} else if (f == -1){

cout << "Error in forking in node with pid: " << getpid() << "**\n**";

} else {

// catch message from new node

zmq::message\_t message\_from\_node;

if(!right\_socket.recv(&message\_from\_node)){

cout << "Error: can't receive message from right node in node with pid:" << getpid() << "**\n**";

}

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

*/\*ofstream op;*

*op.open("text.txt", std::ios::app);*

*op << "Received:" << recieved\_message\_from\_node << "\n";*

*op.flush();\*/*

// send message to main node

if(!main\_socket.send(message\_from\_node)){

cout << "Error: can't send message to main node from node with pid:" << getpid() << "**\n**";

}

}

delete [] adr\_right\_;

delete [] right\_id\_;

} else { // send task to right node

send\_message(recieved\_message, right\_socket);

// catch and send to parent

zmq::message\_t message;

if(!right\_socket.recv(&message)){

cout << "Error: can't receive message from left node in node with pid: " << getpid() << "**\n**";

}

if(!main\_socket.send(message)){

cout << "Error: can't send message to main node from node with pid: " << getpid() << "**\n**";

}

}

} else {

if(left\_id == 0){ // there is not left node

left\_id = id\_proc;

int left\_id\_tmp = left\_id - 1;

while(1){

try {

left\_socket.bind(adr\_left + to\_string(++left\_id\_tmp));

break;

} catch(...) {

}

}

adr\_left += to\_string(left\_id\_tmp);

char\* adr\_left\_ = new char[adr\_left.size() + 1];

memcpy(adr\_left\_, adr\_left.c\_str(), adr\_left.size() + 1);

char\* left\_id\_ = new char[to\_string(left\_id).size() + 1];

memcpy(left\_id\_, to\_string(left\_id).c\_str(), to\_string(left\_id).size() + 1);

char\* args[] = {"./child\_node", adr\_left\_, left\_id\_, NULL};

int f = fork();

if(f == 0){

execv("./child\_node", args);

} else if(f == -1){

cout << "Error in forking in node with pid: " << getpid() << "**\n**";

} else {

// catch message from new node

zmq::message\_t message\_from\_node;

if(!left\_socket.recv(&message\_from\_node)){

cout << "Error: can't receive message from left node in node with pid:" << getpid() << "**\n**";

}

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

// send message to main node

if(!main\_socket.send(message\_from\_node)){

cout << "Error: can't send message to main node from node with pid:" << getpid() << "**\n**";

}

}

delete [] adr\_left\_;

delete [] left\_id\_;

} else { // send task to left node

send\_message(recieved\_message, left\_socket);

// catch and send to parent

zmq::message\_t message;

if(!left\_socket.recv(&message)){

cout << "Error: can't receive message from left node in node with pid: " << getpid() << "**\n**";

}

if(!main\_socket.send(message)){

cout << "Error: can't send message to main node from node with pid: " << getpid() << "**\n**";

}

}

}

} else if(command == "ping") {

*/\*ofstream o;*

*o.open("text.txt", std::ios::app);*

*o << "Ping: " << left\_id << " " << id << " " << right\_id << "\n";*

*o.flush();\*/*

int id\_proc; // id of node for creating

string id\_proc\_;

for(int i = 5; i < recieved\_message.size(); ++i){

if(recieved\_message[i] != ' '){

id\_proc\_ += recieved\_message[i];

} else {

break;

}

}

id\_proc = stoi(id\_proc\_);

if(id\_proc == id){

send\_message("OK: 1", main\_socket);

} else if(id\_proc < id) {

if(left\_id == 0){

send\_message("OK: 0", main\_socket);

} else {

left\_socket.send(message\_main);

zmq::message\_t answ;

left\_socket.recv(&answ);

main\_socket.send(answ);

}

} else if(id\_proc > id) {

if(right\_id == 0){

send\_message("OK: 0", main\_socket);

} else {

right\_socket.send(message\_main);

zmq::message\_t answ;

right\_socket.recv(&answ);

main\_socket.send(answ);

}

}

} else if(command == "kill") {

*/\*ofstream o;*

*o.open("text.txt", std::ios::app);*

*o << "Kill:" << left\_id << " " << id << " " << right\_id << "\n";*

*o.flush();\*/*

int id\_proc; // id of node for killing

string id\_proc\_;

for(int i = 5; i < recieved\_message.size(); ++i){

if(recieved\_message[i] != ' '){

id\_proc\_ += recieved\_message[i];

} else {

break;

}

}

id\_proc = stoi(id\_proc\_);

if(id\_proc > id){

if(right\_id == 0){

send\_message("Error: there isn`t node with this id", main\_socket);

} else {

if(right\_id == id\_proc){

send\_message("Ok: " + to\_string(right\_id), main\_socket);

send\_message("DIE", right\_socket);

right\_socket.unbind(adr\_right);

adr\_right = "tcp://127.0.0.1:300";

right\_id = 0;

} else {

right\_socket.send(message\_main);

zmq::message\_t message;

right\_socket.recv(&message);

main\_socket.send(message);

}

}

} else if(id\_proc < id){

if(left\_id == 0){

send\_message("Error: there isn`t node with this id", main\_socket);

} else {

if(left\_id == id\_proc){

send\_message("Ok: " + to\_string(left\_id), main\_socket);

send\_message("DIE", left\_socket);

left\_socket.unbind(adr\_left);

adr\_left = "tcp://127.0.0.1:300";

left\_id = 0;

} else {

left\_socket.send(message\_main);

zmq::message\_t message;

left\_socket.recv(&message);

main\_socket.send(message);

}

}

}

} else if (command == "DIE") {

main\_socket.unbind(adr\_right);

main\_socket.close();

return 0;

}

}

}

## **Использование утилиты strace**

execve("./main\_prog", ["./main\_prog"], 0x7ffdaa21bc10 /\* 49 vars \*/) = 0

brk(NULL) = 0x55864355c000

arch\_prctl(0x3001 /\* ARCH\_??? \*/, 0x7ffe397c15d0) = -1 EINVAL (Недопустимый аргумент)

access("/etc/ld.so.preload", R\_OK) = -1 ENOENT (Нет такого файла или каталога)

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

fstat(3, {st\_mode=S\_IFREG|0644, st\_size=73408, ...}) = 0

mmap(NULL, 73408, PROT\_READ, MAP\_PRIVATE, 3, 0) = 0x7f85babc5000

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) = 0x7f85babc3000

mmap(NULL, 678128, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f85bab1d000

mmap(0x7f85bab33000, 430080, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x16000) = 0x7f85bab33000

mmap(0x7f85bab9c000, 126976, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x7f000) = 0x7f85bab9c000

mmap(0x7f85babbb000, 32768, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x9d000) = 0x7f85babbb000

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

fstat(3, {st\_mode=S\_IFREG|0644, st\_size=1956992, ...}) = 0

mmap(NULL, 1972224, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f85ba93b000

mprotect(0x7f85ba9d1000, 1290240, PROT\_NONE) = 0

mmap(0x7f85ba9d1000, 987136, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x96000) = 0x7f85ba9d1000

mmap(0x7f85baac2000, 299008, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x187000) = 0x7f85baac2000

mmap(0x7f85bab0c000, 57344, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1d0000) = 0x7f85bab0c000

mmap(0x7f85bab1a000, 10240, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7f85bab1a000

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) = 0x7f85ba920000

mmap(0x7f85ba923000, 73728, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x3000) = 0x7f85ba923000

mmap(0x7f85ba935000, 16384, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x15000) = 0x7f85ba935000

mmap(0x7f85ba939000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x18000) = 0x7f85ba939000

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) = 0x7f85ba72e000

mprotect(0x7f85ba753000, 1847296, PROT\_NONE) = 0

mmap(0x7f85ba753000, 1540096, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x25000) = 0x7f85ba753000

mmap(0x7f85ba8cb000, 303104, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x19d000) = 0x7f85ba8cb000

mmap(0x7f85ba916000, 24576, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1e7000) = 0x7f85ba916000

mmap(0x7f85ba91c000, 13528, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7f85ba91c000

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) = 0x7f85ba6d6000

mmap(0x7f85ba6e2000, 229376, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xc000) = 0x7f85ba6e2000

mmap(0x7f85ba71a000, 73728, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x44000) = 0x7f85ba71a000

mmap(0x7f85ba72c000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x55000) = 0x7f85ba72c000

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) = 0x7f85ba687000

mmap(0x7f85ba68b000, 163840, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x4000) = 0x7f85ba68b000

mmap(0x7f85ba6b3000, 118784, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x2c000) = 0x7f85ba6b3000

mmap(0x7f85ba6d0000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x48000) = 0x7f85ba6d0000

mmap(0x7f85ba6d2000, 14384, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7f85ba6d2000

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) = 0x7f85ba685000

mmap(NULL, 1420000, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f85ba52a000

mmap(0x7f85ba534000, 421888, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xa000) = 0x7f85ba534000

mmap(0x7f85ba59b000, 217088, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x71000) = 0x7f85ba59b000

mmap(0x7f85ba5d0000, 16384, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xa5000) = 0x7f85ba5d0000

mmap(0x7f85ba5d4000, 723680, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7f85ba5d4000

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) = 0x7f85ba4dd000

mmap(0x7f85ba4e8000, 204800, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xb000) = 0x7f85ba4e8000

mmap(0x7f85ba51a000, 49152, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x3d000) = 0x7f85ba51a000

mmap(0x7f85ba526000, 16384, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x48000) = 0x7f85ba526000

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) = 0x7f85ba4ba000

mmap(0x7f85ba4c1000, 69632, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x7000) = 0x7f85ba4c1000

mmap(0x7f85ba4d2000, 20480, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x18000) = 0x7f85ba4d2000

mmap(0x7f85ba4d7000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1c000) = 0x7f85ba4d7000

mmap(0x7f85ba4d9000, 13432, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7f85ba4d9000

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) = 0x7f85ba36b000

mmap(0x7f85ba37a000, 684032, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xf000) = 0x7f85ba37a000

mmap(0x7f85ba421000, 618496, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xb6000) = 0x7f85ba421000

mmap(0x7f85ba4b8000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x14c000) = 0x7f85ba4b8000

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) = 0x7f85ba28e000

mprotect(0x7f85ba2b0000, 700416, PROT\_NONE) = 0

mmap(0x7f85ba2b0000, 397312, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x22000) = 0x7f85ba2b0000

mmap(0x7f85ba311000, 299008, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x83000) = 0x7f85ba311000

mmap(0x7f85ba35b000, 65536, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xcc000) = 0x7f85ba35b000

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) = 0x7f85ba28c000

mmap(NULL, 196696, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f85ba25b000

mprotect(0x7f85ba25f000, 172032, PROT\_NONE) = 0

mmap(0x7f85ba25f000, 114688, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x4000) = 0x7f85ba25f000

mmap(0x7f85ba27b000, 53248, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x20000) = 0x7f85ba27b000

mmap(0x7f85ba289000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x2d000) = 0x7f85ba289000

mmap(0x7f85ba28b000, 88, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7f85ba28b000

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) = 0x7f85ba254000

mmap(0x7f85ba256000, 8192, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x2000) = 0x7f85ba256000

mmap(0x7f85ba258000, 4096, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x4000) = 0x7f85ba258000

mmap(0x7f85ba259000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x4000) = 0x7f85ba259000

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) = 0x7f85ba245000

mmap(0x7f85ba248000, 28672, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x3000) = 0x7f85ba248000

mmap(0x7f85ba24f000, 12288, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xa000) = 0x7f85ba24f000

mmap(0x7f85ba252000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xc000) = 0x7f85ba252000

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) = 0x7f85ba23e000

mmap(0x7f85ba240000, 8192, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x2000) = 0x7f85ba240000

mmap(0x7f85ba242000, 4096, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x4000) = 0x7f85ba242000

mmap(0x7f85ba243000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x4000) = 0x7f85ba243000

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) = 0x7f85ba222000

mprotect(0x7f85ba226000, 81920, PROT\_NONE) = 0

mmap(0x7f85ba226000, 65536, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x4000) = 0x7f85ba226000

mmap(0x7f85ba236000, 12288, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x14000) = 0x7f85ba236000

mmap(0x7f85ba23a000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x17000) = 0x7f85ba23a000

mmap(0x7f85ba23c000, 6784, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7f85ba23c000

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) = 0x7f85ba220000

mmap(NULL, 20752, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f85ba21a000

mmap(0x7f85ba21b000, 8192, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1000) = 0x7f85ba21b000

mmap(0x7f85ba21d000, 4096, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x3000) = 0x7f85ba21d000

mmap(0x7f85ba21e000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x3000) = 0x7f85ba21e000

close(3) = 0

mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7f85ba218000

mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7f85ba216000

arch\_prctl(ARCH\_SET\_FS, 0x7f85ba219600) = 0

mprotect(0x7f85ba916000, 12288, PROT\_READ) = 0

mprotect(0x7f85ba21e000, 4096, PROT\_READ) = 0

mprotect(0x7f85ba23a000, 4096, PROT\_READ) = 0

mprotect(0x7f85ba243000, 4096, PROT\_READ) = 0

mprotect(0x7f85ba252000, 4096, PROT\_READ) = 0

mprotect(0x7f85ba4d7000, 4096, PROT\_READ) = 0

mprotect(0x7f85ba259000, 4096, PROT\_READ) = 0

mprotect(0x7f85ba289000, 4096, PROT\_READ) = 0

mprotect(0x7f85ba35b000, 57344, PROT\_READ) = 0

mprotect(0x7f85ba4b8000, 4096, PROT\_READ) = 0

mprotect(0x7f85ba526000, 8192, PROT\_READ) = 0

mprotect(0x7f85ba939000, 4096, PROT\_READ) = 0

mprotect(0x7f85bab0c000, 45056, PROT\_READ) = 0

mprotect(0x7f85ba5d0000, 12288, PROT\_READ) = 0

mprotect(0x7f85ba6d0000, 4096, PROT\_READ) = 0

mprotect(0x7f85ba72c000, 4096, PROT\_READ) = 0

mprotect(0x7f85babbb000, 28672, PROT\_READ) = 0

mprotect(0x55864295f000, 4096, PROT\_READ) = 0

mprotect(0x7f85bac04000, 4096, PROT\_READ) = 0

munmap(0x7f85babc5000, 73408) = 0

set\_tid\_address(0x7f85ba2198d0) = 36901

set\_robust\_list(0x7f85ba2198e0, 24) = 0

rt\_sigaction(SIGRTMIN, {sa\_handler=0x7f85ba4c1bf0, sa\_mask=[], sa\_flags=SA\_RESTORER|SA\_SIGINFO, sa\_restorer=0x7f85ba4cf3c0}, NULL, 8) = 0

rt\_sigaction(SIGRT\_1, {sa\_handler=0x7f85ba4c1c90, sa\_mask=[], sa\_flags=SA\_RESTORER|SA\_RESTART|SA\_SIGINFO, sa\_restorer=0x7f85ba4cf3c0}, 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) = 0x55864355c000

brk(0x55864357d000) = 0x55864357d000

futex(0x7f85bab1a6bc, FUTEX\_WAKE\_PRIVATE, 2147483647) = 0

futex(0x7f85bab1a6c8, FUTEX\_WAKE\_PRIVATE, 2147483647) = 0

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

read(3, "0-3\n", 8192) = 4

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, /\* 22 entries \*/, 32768) = 656

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

close(3) = 0

getpid() = 36901

sched\_getaffinity(36901, 128, [0, 1, 2, 3]) = 8

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

fstat(3, {st\_mode=S\_IFREG|0644, st\_size=542, ...}) = 0

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

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=73408, ...}) = 0

mmap(NULL, 73408, PROT\_READ, MAP\_PRIVATE, 3, 0) = 0x7f85babc5000

close(3) = 0

openat(AT\_FDCWD, "/lib/x86\_64-linux-gnu/tls/x86\_64/x86\_64/libnss\_db.so.2", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (Нет такого файла или каталога)

stat("/lib/x86\_64-linux-gnu/tls/x86\_64/x86\_64", 0x7ffe397be740) = -1 ENOENT (Нет такого файла или каталога)

openat(AT\_FDCWD, "/lib/x86\_64-linux-gnu/tls/x86\_64/libnss\_db.so.2", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (Нет такого файла или каталога)

stat("/lib/x86\_64-linux-gnu/tls/x86\_64", 0x7ffe397be740) = -1 ENOENT (Нет такого файла или каталога)

openat(AT\_FDCWD, "/lib/x86\_64-linux-gnu/tls/x86\_64/libnss\_db.so.2", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (Нет такого файла или каталога)

stat("/lib/x86\_64-linux-gnu/tls/x86\_64", 0x7ffe397be740) = -1 ENOENT (Нет такого файла или каталога)

openat(AT\_FDCWD, "/lib/x86\_64-linux-gnu/tls/libnss\_db.so.2", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (Нет такого файла или каталога)

stat("/lib/x86\_64-linux-gnu/tls", 0x7ffe397be740) = -1 ENOENT (Нет такого файла или каталога)

openat(AT\_FDCWD, "/lib/x86\_64-linux-gnu/x86\_64/x86\_64/libnss\_db.so.2", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (Нет такого файла или каталога)

stat("/lib/x86\_64-linux-gnu/x86\_64/x86\_64", 0x7ffe397be740) = -1 ENOENT (Нет такого файла или каталога)

openat(AT\_FDCWD, "/lib/x86\_64-linux-gnu/x86\_64/libnss\_db.so.2", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (Нет такого файла или каталога)

stat("/lib/x86\_64-linux-gnu/x86\_64", 0x7ffe397be740) = -1 ENOENT (Нет такого файла или каталога)

openat(AT\_FDCWD, "/lib/x86\_64-linux-gnu/x86\_64/libnss\_db.so.2", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (Нет такого файла или каталога)

stat("/lib/x86\_64-linux-gnu/x86\_64", 0x7ffe397be740) = -1 ENOENT (Нет такого файла или каталога)

openat(AT\_FDCWD, "/lib/x86\_64-linux-gnu/libnss\_db.so.2", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (Нет такого файла или каталога)

stat("/lib/x86\_64-linux-gnu", {st\_mode=S\_IFDIR|0755, st\_size=77824, ...}) = 0

openat(AT\_FDCWD, "/usr/lib/x86\_64-linux-gnu/tls/x86\_64/x86\_64/libnss\_db.so.2", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (Нет такого файла или каталога)

stat("/usr/lib/x86\_64-linux-gnu/tls/x86\_64/x86\_64", 0x7ffe397be740) = -1 ENOENT (Нет такого файла или каталога)

openat(AT\_FDCWD, "/usr/lib/x86\_64-linux-gnu/tls/x86\_64/libnss\_db.so.2", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (Нет такого файла или каталога)

stat("/usr/lib/x86\_64-linux-gnu/tls/x86\_64", 0x7ffe397be740) = -1 ENOENT (Нет такого файла или каталога)

openat(AT\_FDCWD, "/usr/lib/x86\_64-linux-gnu/tls/x86\_64/libnss\_db.so.2", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (Нет такого файла или каталога)

stat("/usr/lib/x86\_64-linux-gnu/tls/x86\_64", 0x7ffe397be740) = -1 ENOENT (Нет такого файла или каталога)

openat(AT\_FDCWD, "/usr/lib/x86\_64-linux-gnu/tls/libnss\_db.so.2", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (Нет такого файла или каталога)

stat("/usr/lib/x86\_64-linux-gnu/tls", 0x7ffe397be740) = -1 ENOENT (Нет такого файла или каталога)

openat(AT\_FDCWD, "/usr/lib/x86\_64-linux-gnu/x86\_64/x86\_64/libnss\_db.so.2", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (Нет такого файла или каталога)

stat("/usr/lib/x86\_64-linux-gnu/x86\_64/x86\_64", 0x7ffe397be740) = -1 ENOENT (Нет такого файла или каталога)

openat(AT\_FDCWD, "/usr/lib/x86\_64-linux-gnu/x86\_64/libnss\_db.so.2", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (Нет такого файла или каталога)

stat("/usr/lib/x86\_64-linux-gnu/x86\_64", 0x7ffe397be740) = -1 ENOENT (Нет такого файла или каталога)

openat(AT\_FDCWD, "/usr/lib/x86\_64-linux-gnu/x86\_64/libnss\_db.so.2", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (Нет такого файла или каталога)

stat("/usr/lib/x86\_64-linux-gnu/x86\_64", 0x7ffe397be740) = -1 ENOENT (Нет такого файла или каталога)

openat(AT\_FDCWD, "/usr/lib/x86\_64-linux-gnu/libnss\_db.so.2", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (Нет такого файла или каталога)

stat("/usr/lib/x86\_64-linux-gnu", {st\_mode=S\_IFDIR|0755, st\_size=77824, ...}) = 0

openat(AT\_FDCWD, "/lib/tls/x86\_64/x86\_64/libnss\_db.so.2", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (Нет такого файла или каталога)

stat("/lib/tls/x86\_64/x86\_64", 0x7ffe397be740) = -1 ENOENT (Нет такого файла или каталога)

openat(AT\_FDCWD, "/lib/tls/x86\_64/libnss\_db.so.2", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (Нет такого файла или каталога)

stat("/lib/tls/x86\_64", 0x7ffe397be740) = -1 ENOENT (Нет такого файла или каталога)

openat(AT\_FDCWD, "/lib/tls/x86\_64/libnss\_db.so.2", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (Нет такого файла или каталога)

stat("/lib/tls/x86\_64", 0x7ffe397be740) = -1 ENOENT (Нет такого файла или каталога)

openat(AT\_FDCWD, "/lib/tls/libnss\_db.so.2", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (Нет такого файла или каталога)

stat("/lib/tls", 0x7ffe397be740) = -1 ENOENT (Нет такого файла или каталога)

openat(AT\_FDCWD, "/lib/x86\_64/x86\_64/libnss\_db.so.2", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (Нет такого файла или каталога)

stat("/lib/x86\_64/x86\_64", 0x7ffe397be740) = -1 ENOENT (Нет такого файла или каталога)

openat(AT\_FDCWD, "/lib/x86\_64/libnss\_db.so.2", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (Нет такого файла или каталога)

stat("/lib/x86\_64", 0x7ffe397be740) = -1 ENOENT (Нет такого файла или каталога)

openat(AT\_FDCWD, "/lib/x86\_64/libnss\_db.so.2", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (Нет такого файла или каталога)

stat("/lib/x86\_64", 0x7ffe397be740) = -1 ENOENT (Нет такого файла или каталога)

openat(AT\_FDCWD, "/lib/libnss\_db.so.2", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (Нет такого файла или каталога)

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

openat(AT\_FDCWD, "/usr/lib/tls/x86\_64/x86\_64/libnss\_db.so.2", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (Нет такого файла или каталога)

stat("/usr/lib/tls/x86\_64/x86\_64", 0x7ffe397be740) = -1 ENOENT (Нет такого файла или каталога)

openat(AT\_FDCWD, "/usr/lib/tls/x86\_64/libnss\_db.so.2", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (Нет такого файла или каталога)

stat("/usr/lib/tls/x86\_64", 0x7ffe397be740) = -1 ENOENT (Нет такого файла или каталога)

openat(AT\_FDCWD, "/usr/lib/tls/x86\_64/libnss\_db.so.2", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (Нет такого файла или каталога)

stat("/usr/lib/tls/x86\_64", 0x7ffe397be740) = -1 ENOENT (Нет такого файла или каталога)

openat(AT\_FDCWD, "/usr/lib/tls/libnss\_db.so.2", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (Нет такого файла или каталога)

stat("/usr/lib/tls", 0x7ffe397be740) = -1 ENOENT (Нет такого файла или каталога)

openat(AT\_FDCWD, "/usr/lib/x86\_64/x86\_64/libnss\_db.so.2", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (Нет такого файла или каталога)

stat("/usr/lib/x86\_64/x86\_64", 0x7ffe397be740) = -1 ENOENT (Нет такого файла или каталога)

openat(AT\_FDCWD, "/usr/lib/x86\_64/libnss\_db.so.2", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (Нет такого файла или каталога)

stat("/usr/lib/x86\_64", 0x7ffe397be740) = -1 ENOENT (Нет такого файла или каталога)

openat(AT\_FDCWD, "/usr/lib/x86\_64/libnss\_db.so.2", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (Нет такого файла или каталога)

stat("/usr/lib/x86\_64", 0x7ffe397be740) = -1 ENOENT (Нет такого файла или каталога)

openat(AT\_FDCWD, "/usr/lib/libnss\_db.so.2", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (Нет такого файла или каталога)

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

munmap(0x7f85babc5000, 73408) = 0

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

fstat(3, {st\_mode=S\_IFREG|0644, st\_size=73408, ...}) = 0

mmap(NULL, 73408, PROT\_READ, MAP\_PRIVATE, 3, 0) = 0x7f85babc5000

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) = 0x7f85ba202000

mmap(0x7f85ba205000, 28672, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x3000) = 0x7f85ba205000

mmap(0x7f85ba20c000, 8192, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xa000) = 0x7f85ba20c000

mmap(0x7f85ba20e000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xb000) = 0x7f85ba20e000

mmap(0x7f85ba210000, 22328, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7f85ba210000

close(3) = 0

mprotect(0x7f85ba20e000, 4096, PROT\_READ) = 0

munmap(0x7f85babc5000, 73408) = 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("\x00\xee\x95\xb1\x41\xc2\x3c\x57\xd4\x20\x87\x23\xfe\xdc\xba\xd0", 16, 0) = 16

getrandom("\x58\x07\xe2\x87\x66\xfa\x29\x08\x29\xfb\xea\xe7\x25\x66\x17\x7a", 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=1129770336, u64=94035143748960}}) = 0

epoll\_ctl(5, EPOLL\_CTL\_MOD, 4, {EPOLLIN, {u32=1129770336, u64=94035143748960}}) = 0

mmap(NULL, 8392704, PROT\_NONE, MAP\_PRIVATE|MAP\_ANONYMOUS|MAP\_STACK, -1, 0) = 0x7f85b9a01000

mprotect(0x7f85b9a02000, 8388608, PROT\_READ|PROT\_WRITE) = 0

clone(child\_stack=0x7f85ba200d30, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, parent\_tid=[36902], tls=0x7f85ba201700, child\_tidptr=0x7f85ba2019d0) = 36902

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=1129788000, u64=94035143766624}}) = 0

epoll\_ctl(7, EPOLL\_CTL\_MOD, 6, {EPOLLIN, {u32=1129788000, u64=94035143766624}}) = 0

mmap(NULL, 8392704, PROT\_NONE, MAP\_PRIVATE|MAP\_ANONYMOUS|MAP\_STACK, -1, 0) = 0x7f85b9200000

mprotect(0x7f85b9201000, 8388608, PROT\_READ|PROT\_WRITE) = 0

clone(child\_stack=0x7f85b99ffd30, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, parent\_tid=[36903], tls=0x7f85b9a00700, child\_tidptr=0x7f85b9a009d0) = 36903

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

fstat(1, {st\_mode=S\_IFCHR|0620, st\_rdev=makedev(0x88, 0x3), ...}) = 0

write(1, "Please, enter command\n", 22Please, enter command

) = 22

fstat(0, {st\_mode=S\_IFCHR|0620, st\_rdev=makedev(0x88, 0x3), ...}) = 0

read(0, creat 12

"creat 12\n", 1024) = 9

write(1, "Error: incorrect command\n", 25Error: incorrect command

) = 25

write(1, "Please, enter command\n", 22Please, enter command

) = 22

write(1, "Error: incorrect command\n", 25Error: incorrect command

) = 25

write(1, "Please, enter command\n", 22Please, enter command

) = 22

read(0, create 14

"create 14\n", 1024) = 10

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=36901, nl\_groups=00000000}, [12]) = 0

sendto(9, {{len=20, type=RTM\_GETLINK, flags=NLM\_F\_REQUEST|NLM\_F\_DUMP, seq=1639914113, 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=1320, type=RTM\_NEWLINK, flags=NLM\_F\_MULTI, seq=1639914113, pid=36901}, {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=8, nla\_type=IFLA\_CARRIER\_UP\_COUNT}, 0}, {{nla\_len=8, nla\_type=IFLA\_CARRIER\_DOWN\_COUNT}, 0}, {{nla\_len=5, nla\_type=IFLA\_PROTO\_DOWN}, 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=504737, tx\_packets=504737, rx\_bytes=29732181, tx\_bytes=29732181, 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=504737, tx\_packets=504737, rx\_bytes=29732181, tx\_bytes=29732181, 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=764, 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] = 1, [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=624, 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=3127, reachable\_time=26264, retrans\_time=1000}}, {{nla\_len=212, 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] = 1, [DEVCONF\_RTR\_PROBE\_INTERVAL] = 60000, [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] = 10, [IPSTATS\_MIB\_INOCTETS] = 712, [IPSTATS\_MIB\_INDELIVERS] = 10, [IPSTATS\_MIB\_OUTFORWDATAGRAMS] = 0, [IPSTATS\_MIB\_OUTPKTS] = 10, [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] = 0, [IPSTATS\_MIB\_OUTMCASTPKTS] = 2, [IPSTATS\_MIB\_INBCASTPKTS] = 0, [IPSTATS\_MIB\_OUTBCASTPKTS] = 0, [IPSTATS\_MIB\_INMCASTOCTETS] = 0, [IPSTATS\_MIB\_OUTMCASTOCTETS] = 152, [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] = 2, [ICMP6\_MIB\_INERRORS] = 0, [ICMP6\_MIB\_OUTMSGS] = 2, [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=1340, type=RTM\_NEWLINK, flags=NLM\_F\_MULTI, seq=1639914113, pid=36901}, {ifi\_family=AF\_UNSPEC, ifi\_type=ARPHRD\_ETHER, ifi\_index=if\_nametoindex("enp3s0"), ifi\_flags=IFF\_UP|IFF\_BROADCAST|IFF\_MULTICAST, ifi\_change=0}, [{{nla\_len=11, nla\_type=IFLA\_IFNAME}, "enp3s0"}, {{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}, 9194}, {{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}, 64}, {{nla\_len=8, nla\_type=IFLA\_GSO\_MAX\_SIZE}, 64000}, {{nla\_len=8, nla\_type=IFLA\_NUM\_RX\_QUEUES}, 1}, {{nla\_len=5, nla\_type=IFLA\_CARRIER}, 0}, {{nla\_len=13, nla\_type=IFLA\_QDISC}, "fq\_codel"}, {{nla\_len=8, nla\_type=IFLA\_CARRIER\_CHANGES}, 1}, {{nla\_len=8, nla\_type=IFLA\_CARRIER\_UP\_COUNT}, 0}, {{nla\_len=8, nla\_type=IFLA\_CARRIER\_DOWN\_COUNT}, 1}, {{nla\_len=5, nla\_type=IFLA\_PROTO\_DOWN}, 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}, "\x54\xb2\x03\x92\x1b\x27"}, {{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=10, nla\_type=0x36 /\* IFLA\_??? \*/}, "\x54\xb2\x03\x92\x1b\x27"}, {{nla\_len=764, 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] = 2, [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] = 1, [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=624, 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=1578628, reachable\_time=33300, retrans\_time=1000}}, {{nla\_len=212, nla\_type=IFLA\_INET6\_CONF}, [[DEVCONF\_FORWARDING] = 0, [DEVCONF\_HOPLIMIT] = 64, [DEVCONF\_MTU6] = 1500, [DEVCONF\_ACCEPT\_RA] = 0, [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] = 1, [DEVCONF\_RTR\_PROBE\_INTERVAL] = 60000, [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\_NONE}]}]}]}], iov\_len=4096}], msg\_iovlen=1, msg\_controllen=0, msg\_flags=0}, 0) = 2660

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=1639914113, pid=36901}, {ifi\_family=AF\_UNSPEC, ifi\_type=ARPHRD\_ETHER, ifi\_index=if\_nametoindex("wlp2s0"), ifi\_flags=IFF\_UP|IFF\_BROADCAST|IFF\_RUNNING|IFF\_MULTICAST|IFF\_LOWER\_UP, ifi\_change=0}, [{{nla\_len=11, nla\_type=IFLA\_IFNAME}, "wlp2s0"}, {{nla\_len=8, nla\_type=IFLA\_TXQLEN}, 1000}, {{nla\_len=5, nla\_type=IFLA\_OPERSTATE}, 6}, {{nla\_len=5, nla\_type=IFLA\_LINKMODE}, 1}, {{nla\_len=8, nla\_type=IFLA\_MTU}, 1500}, {{nla\_len=8, nla\_type=IFLA\_MIN\_MTU}, 256}, {{nla\_len=8, nla\_type=IFLA\_MAX\_MTU}, 2304}, {{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}, 68}, {{nla\_len=8, nla\_type=IFLA\_CARRIER\_UP\_COUNT}, 34}, {{nla\_len=8, nla\_type=IFLA\_CARRIER\_DOWN\_COUNT}, 34}, {{nla\_len=5, nla\_type=IFLA\_PROTO\_DOWN}, 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}, "\x48\x89\xe7\xeb\x69\xb4"}, {{nla\_len=10, nla\_type=IFLA\_BROADCAST}, "\xff\xff\xff\xff\xff\xff"}, {{nla\_len=196, nla\_type=IFLA\_STATS64}, {rx\_packets=2434696, tx\_packets=1023787, rx\_bytes=3116013699, tx\_bytes=160602672, rx\_errors=0, tx\_errors=0, rx\_dropped=2, 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=2434696, tx\_packets=1023787, rx\_bytes=3116013699, tx\_bytes=160602672, rx\_errors=0, tx\_errors=0, rx\_dropped=2, 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=10, nla\_type=0x36 /\* IFLA\_??? \*/}, "\x48\x89\xe7\xeb\x69\xb4"}, {{nla\_len=764, 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] = 2, [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] = 1, [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=624, 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=1976130, reachable\_time=24148, retrans\_time=1000}}, {{nla\_len=212, nla\_type=IFLA\_INET6\_CONF}, [[DEVCONF\_FORWARDING] = 0, [DEVCONF\_HOPLIMIT] = 64, [DEVCONF\_MTU6] = 1500, [DEVCONF\_ACCEPT\_RA] = 0, [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] = 2, [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] = 1, [DEVCONF\_RTR\_PROBE\_INTERVAL] = 60000, [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] = 807, [IPSTATS\_MIB\_INOCTETS] = 125632, [IPSTATS\_MIB\_INDELIVERS] = 807, [IPSTATS\_MIB\_OUTFORWDATAGRAMS] = 0, [IPSTATS\_MIB\_OUTPKTS] = 499, [IPSTATS\_MIB\_OUTOCTETS] = 52324, [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] = 20, [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] = 807, [IPSTATS\_MIB\_OUTMCASTPKTS] = 499, [IPSTATS\_MIB\_INBCASTPKTS] = 0, [IPSTATS\_MIB\_OUTBCASTPKTS] = 0, [IPSTATS\_MIB\_INMCASTOCTETS] = 125632, [IPSTATS\_MIB\_OUTMCASTOCTETS] = 52324, [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] = 5, [ICMP6\_MIB\_INERRORS] = 0, [ICMP6\_MIB\_OUTMSGS] = 254, [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\_NONE}]}]}]}, iov\_len=4096}], msg\_iovlen=1, msg\_controllen=0, msg\_flags=0}, 0) = 1336

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=1639914113, pid=36901}, 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=1639914114, 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=1639914114, pid=36901}, {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=3127, tstamp=3127}}]}, {{len=88, type=RTM\_NEWADDR, flags=NLM\_F\_MULTI, seq=1639914114, pid=36901}, {ifa\_family=AF\_INET, ifa\_prefixlen=16, ifa\_flags=0, ifa\_scope=RT\_SCOPE\_UNIVERSE, ifa\_index=if\_nametoindex("wlp2s0")}, [{{nla\_len=8, nla\_type=IFA\_ADDRESS}, inet\_addr("172.17.19.45")}, {{nla\_len=8, nla\_type=IFA\_LOCAL}, inet\_addr("172.17.19.45")}, {{nla\_len=8, nla\_type=IFA\_BROADCAST}, inet\_addr("172.17.255.255")}, {{nla\_len=11, nla\_type=IFA\_LABEL}, "wlp2s0"}, {{nla\_len=8, nla\_type=IFA\_FLAGS}, IFA\_F\_NOPREFIXROUTE}, {{nla\_len=20, nla\_type=IFA\_CACHEINFO}, {ifa\_prefered=1584, ifa\_valid=1584, cstamp=1976137, tstamp=1976684}}]}], 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=1639914114, pid=36901}, {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=3127, tstamp=3127}}, {{nla\_len=8, nla\_type=IFA\_FLAGS}, IFA\_F\_PERMANENT}]}, {{len=72, type=RTM\_NEWADDR, flags=NLM\_F\_MULTI, seq=1639914114, pid=36901}, {ifa\_family=AF\_INET6, ifa\_prefixlen=64, ifa\_flags=IFA\_F\_PERMANENT, ifa\_scope=RT\_SCOPE\_LINK, ifa\_index=if\_nametoindex("wlp2s0")}, [{{nla\_len=20, nla\_type=IFA\_ADDRESS}, inet\_pton(AF\_INET6, "fe80::df41:123d:8f15:da36")}, {{nla\_len=20, nla\_type=IFA\_CACHEINFO}, {ifa\_prefered=4294967295, ifa\_valid=4294967295, cstamp=1976130, tstamp=1976320}}, {{nla\_len=8, nla\_type=IFA\_FLAGS}, IFA\_F\_PERMANENT|IFA\_F\_NOPREFIXROUTE}]}], 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=1639914114, pid=36901}, 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(30014), sin\_addr=inet\_addr("127.0.0.1")}, 16) = 0

listen(9, 100) = 0

getsockname(9, {sa\_family=AF\_INET, sin\_port=htons(30014), sin\_addr=inet\_addr("127.0.0.1")}, [128->16]) = 0

getsockname(9, {sa\_family=AF\_INET, sin\_port=htons(30014), 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

clone(child\_stack=NULL, flags=CLONE\_CHILD\_CLEARTID|CLONE\_CHILD\_SETTID|SIGCHLD, child\_tidptr=0x7f85ba2198d0) = 36910

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)

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

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

write(6, "\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: 36910\n", 10OK: 36910

) = 10

write(1, "Please, enter command\n", 22Please, enter command

) = 22

read(0, create 16

"create 16\n", 1024) = 10

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)

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(1, "OK: 36917\n", 10OK: 36917

) = 10

write(1, "Please, enter command\n", 22Please, enter command

) = 22

read(0, kill 16

"kill 16\n", 1024) = 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}])

terminate called after throwing an instance of 'zmq::error\_t'

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

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

write(1, "Ok: 16\n", 7Ok: 16

) = 7

write(1, "Please, enter command\n", 22Please, enter command

) = 22

read(0, what(): No such file or directory

exit

"exit\n", 1024) = 5

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 (Недопустимая операция смещения)

exit\_group(0) = ?

+++ exited with 0 +++

## **Демонстрация работы программы**

Please, enter command

create 14

OK: 41012

Please, enter command

create 11

OK: 41019

Please, enter command

creaet 1

Error: incorrect command

Please, enter command

create 1

OK: 41125

Please, enter command

create 19

OK: 41158

Please, enter command

oing 19

Error: incorrect command

Please, enter command

Error: incorrect command

Please, enter command

ping 19

OK: 1

Please, enter command

kill 19

Ok: 19

Please, enter command

ping 19

OK: 0

Please, enter command

create 19

OK: 41375

Please, enter command

exec 19 tyt 78

OK:19

Please, enter command

exec 19

tyt

OK:19:78

Please, enter command

exit

## **Вывод**

Хотя лабораторная работа была сложной (самая сложная лабораторная работа за все время обучения в МАИ), но очень интересной. Ведь в ней сразу можно применить знания, полученные в ходе выполнения предыдущих лабораторных работ, так как здесь и многопоточность, и межпроцессорное взаимодействие, основанное на очередях сообщений, и синхронизация потоков. А помимо всего этого также необходимо разобраться с дополнительной библиотекой (zmq). Единственное, что мне не очень понравилось – наличие небольшого количества документации по этой иблиотеки и более сложная отладка программ. Для поиска багов и ошибок в вычислительных узлах был организован вывод в файл, что помогло найти и исправить несколько сложноуловимых багов.