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: Чекменев В.А.

Группа: М80-207Б-20

Преподаватель: Миронов Е.С.

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

Дата:

**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:** 27. Топология — связный список. Тип вычислительной команды — поиск подстроки в строке. Тип проверки узлов на доступность — пинг всех узлов (заменено на пинг определенного узла).

# **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 следующий узел, из которого возвращается некоторое сообщение(об успехе или об ошибке), которое потом пересылается обратно по списку.
* Для корректной проверки на доступность узлов, используется список, эмулирующий поведение узлов в данной топологии.

## **Код программы**

**client.cpp:**

#include <iostream>

#include <zmq.h>

#include <unistd.h>

#include <fcntl.h>

#include <zmq.hpp>

#include <unistd.h>

#include <string>

#include "server.hpp"

using namespace std;

int TopVertex = 0;

void create(int port, zmq::socket\_t& socket)

{

int NodeId;

cin >> NodeId;

if (TopVertex == 0) {

int NodePid = fork();

if (NodePid == -1) {

cout << "Error: fail fork()\n";

return;

} else if (NodePid == 0) { // in child, executing the server program

execl("./server", "server", to\_string(NodeId).c\_str(), to\_string(port).c\_str(), NULL);

} else { // in parent, print OK message

cout << "Ok: " << NodePid << "\n";

}

TopVertex = 1;

} else {

sendMessage(socket, "create " + to\_string(NodeId));

string Ans = receiveMessage(socket);

cout << Ans;

}

}

void exec(zmq::socket\_t& socket)

{

int NodeId;

string text, pattern;

cin >> NodeId;

if (TopVertex == 0) { // we need to create this node

cout << "Error: " + std::to\_string(NodeId) + ": Not found\n";

return;

}

cin >> text >> pattern;

sendMessage(socket, "exec " + to\_string(NodeId) + " " + text + " " + pattern);

string Ans = receiveMessage(socket);

cout << Ans;

}

void pingid(zmq::socket\_t& socket)

{

int NodeId;

cin >> NodeId;

if (TopVertex == 0) {

cout << "There is no any node here yet\n";

return;

}

sendMessage(socket, "pingid " + to\_string(NodeId));

string ans = receiveMessage(socket);

cout << ans;

}

int main()

{

zmq::context\_t context(1);

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

int port = bindSocket(Rule\_socket);

string s, text, pattern;

cout << "> ";

while (true) {

cin >> s;

if (s == "create") {

create(port, Rule\_socket);

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

exec(Rule\_socket);

} else if (s == "pingid") {

pingid(Rule\_socket);

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

return 0;

} else {

cout << "Error Command\n";

}

cout << "> ";

}

return 0;

}

**server.cpp:**

#include <iostream>

#include <string>

#include <unistd.h>

#include <fcntl.h>

#include <ctime>

#include "server.hpp"

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

{

int CurNodeId = atoi(argv[1]);

int ParentPort = atoi(argv[2]);

zmq::context\_t context(2);

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

ChildSocket.connect(getPortName(ParentPort));

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

int ChildPort = bindSocket(req\_socket);

int ChildTopVertex = 0;

std::string cmd;

while(true) {

std::string request = receiveMessage(ChildSocket); // receive message from parent (client)

std::istringstream cmdStream(request); // thread to read data from accepted string

cmdStream >> cmd;

if (cmd == "create") {

int NodeId;

cmdStream >> NodeId;

if (CurNodeId == NodeId) { // if node has already been created

sendMessage(ChildSocket, "Error: Already exists\n");

} else {

if (ChildTopVertex == 0) {

int NodePid = fork();

if (NodePid == -1) {

sendMessage(ChildSocket, "Error: fork() fail\n");

} else if (NodePid == 0) {

execl("./server", "server", std::to\_string(NodeId).c\_str(), std::to\_string(ChildPort).c\_str(), NULL);

} else {

sendMessage(ChildSocket, "Ok: " + std::to\_string(NodePid) + "\n");

}

ChildTopVertex = 1;

} else {

sendMessage(req\_socket, "create " + std::to\_string(NodeId));

std::string AnsFromChild = receiveMessage(req\_socket);

sendMessage(ChildSocket, AnsFromChild);

}

}

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

int NodeId;

std::string Text, Pattern;

cmdStream >> NodeId;

cmdStream >> Text >> Pattern;

if (NodeId == CurNodeId) {

int pos = 0;

std::vector<size\_t> entries = KMP(Pattern, Text);

std::string Ans;

if (entries.size() != 0) {

for (int i = 0; i < entries.size(); i++) {

if (i == 0) Ans += std::to\_string(entries[i]);

else Ans += " " + std::to\_string(entries[i]);

}

} else {

Ans += "-1";

}

sendMessage(ChildSocket, "Ok: " + std::to\_string(NodeId) + ": [" + Ans + "]\n"); // send result up

} else {

if (ChildTopVertex == 0) { // there is no node with this nodeid

sendMessage(ChildSocket, "Error: " + std::to\_string(NodeId) + ": Not found\n");

} else { // send message to next node

sendMessage(req\_socket, "exec " + std::to\_string(NodeId) + " " + Text + " " + Pattern);

std::string Ans = receiveMessage(req\_socket); // receive and send message to next node

sendMessage(ChildSocket, Ans);

}

}

} else if (cmd == "pingid") {

int NodeId;

cmdStream >> NodeId;

if (NodeId == CurNodeId) {

sendMessage(ChildSocket, "Ok: 1\n");

} else {

if (ChildTopVertex == 0) {

sendMessage(ChildSocket, "Not found\n");

} else {

sendMessage(req\_socket, "pingid " + std::to\_string(NodeId));

std::string ans = receiveMessage(req\_socket);

if (ans == "Ok: 1\n") {

sendMessage(ChildSocket, ans);

} else {

sendMessage(ChildSocket, "Ok: 0\n");

}

}

}

}

}

return 0;

}

**server.hpp:**

#pragma once

#include <cstdlib>

#include <string>

#include <vector>

#include <unistd.h>

#include <zmq.hpp>

std::vector<size\_t> prefix\_function(std::string s)

{

size\_t n = s.length();

// в i-м элементе (его индекс i-1) количество

// совпавших символов в начале и конце для подстроки длины i.

std::vector<size\_t> pi(n);

// p[0]=0 всегда, p[1]=1, если начинается с двух одинаковых

for (size\_t i=1; i<n; ++i)

{

// ищем, какой префикс-суффикс можно расширить

size\_t j = pi[i-1]; // длина предыдущего префикса-суффикса, возможно нулевая

while ((j > 0) && (s[i] != s[j])) // этот нельзя расширить,

j = pi[j-1]; // берем длину меньшего префикса-суффикса

if (s[i] == s[j])

++j; // расширяем найденный (возможно пустой) префикс-суффикс

pi[i] = j;

}

return pi;

}

std::vector<size\_t> KMP(std::string pattern, std::string text)

{

int n = text.length();

int m = pattern.length();

std::vector<size\_t> Lps = prefix\_function(pattern); // применяем префиекс функцию

std::vector<size\_t> out; // вектор с индексами вхождений

int i = 0, j = 0;

while (i < n) {

if (pattern[j] == text[i]) {

i++; j++;

} // если совпало, продолжаем

if (j == m) { // если j==m это подтверждает то, что мы нашли образец в тексте

out.push\_back(i - m); // добавляем этот индекс минус длина образца в вектор out

j = Lps[j - 1]; // обновляем j как префикс последнего совпавшего символа

} else if (i < n && pattern[j] != text[i]) { // если не совпало

if (j == 0) {

i++; // если j становится равным нулю, делаем инкремент индекса i

} else {

j = Lps[j - 1]; // обновляем j как префикс последнего совпавшего символа

}

}

}

return out;

}

// send message to the particular socket

bool sendMessage(zmq::socket\_t &socket, const std::string &message\_string)

{

// message size init

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

// message content init

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

return socket.send(message);

}

std::string receiveMessage(zmq::socket\_t &socket)

{

zmq::message\_t message;

int recResult;

// receiving message from socket

try {

recResult = (int)socket.recv(&message);

if (recResult < 0) {

perror("socket.recv()");

exit(1);

}

}

catch (...) {

recResult = 0;

}

// transform to string

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

if (recieved\_message.empty() || !recResult) {

return "Error: Node is not available";

}

return recieved\_message;

}

std::string getPortName(int port)

{

return "tcp://127.0.0.1:" + std::to\_string(port);

}

int bindSocket(zmq::socket\_t &socket)

{

int port = 8080;

// create endpoint and bind it to the socket

while (true) {

try {

socket.bind(getPortName(port)); // создаёт сокет

break;

}

catch (...) { // в случае неудачи используем другой сокет

port++;

}

}

return port;

}

## **Использование утилиты strace**

[suraba04@asusx512fl third]$ strace ./client

execve("./client", ["./client"], 0x7ffc04e97fc0 /\* 72 vars \*/) = 0

brk(NULL) = 0x55723e2c0000

arch\_prctl(0x3001 /\* ARCH\_??? \*/, 0x7ffc536b3470) = -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

newfstatat(3, "", {st\_mode=S\_IFREG|0644, st\_size=230043, ...}, AT\_EMPTY\_PATH) = 0

mmap(NULL, 230043, PROT\_READ, MAP\_PRIVATE, 3, 0) = 0x7f741cb87000

close(3) = 0

openat(AT\_FDCWD, "/usr/lib/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 \200\1\0\0\0\0\0"..., 832) = 832

newfstatat(3, "", {st\_mode=S\_IFREG|0755, st\_size=743320, ...}, AT\_EMPTY\_PATH) = 0

mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7f741cb85000

mmap(NULL, 745560, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f741cace000

mmap(0x7f741cae6000, 471040, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x18000) = 0x7f741cae6000

mmap(0x7f741cb59000, 143360, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x8b000) = 0x7f741cb59000

mmap(0x7f741cb7c000, 36864, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xad000) = 0x7f741cb7c000

close(3) = 0

openat(AT\_FDCWD, "/usr/lib/libstdc++.so.6", O\_RDONLY|O\_CLOEXEC) = 3

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

pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\210\255N\377\201\240\fhJ\277\340\370c\350t4"..., 36, 800) = 36

newfstatat(3, "", {st\_mode=S\_IFREG|0755, st\_size=18844016, ...}, AT\_EMPTY\_PATH) = 0

mmap(NULL, 2250816, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f741c8a8000

mprotect(0x7f741c941000, 1556480, PROT\_NONE) = 0

mmap(0x7f741c941000, 1085440, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x99000) = 0x7f741c941000

mmap(0x7f741ca4a000, 466944, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1a2000) = 0x7f741ca4a000

mmap(0x7f741cabd000, 57344, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x214000) = 0x7f741cabd000

mmap(0x7f741cacb000, 10304, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7f741cacb000

close(3) = 0

openat(AT\_FDCWD, "/usr/lib/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\0\0\0\0\0\0\0\0"..., 832) = 832

newfstatat(3, "", {st\_mode=S\_IFREG|0755, st\_size=1061880, ...}, AT\_EMPTY\_PATH) = 0

mmap(NULL, 946752, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f741c7c0000

mmap(0x7f741c7d0000, 499712, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x10000) = 0x7f741c7d0000

mmap(0x7f741c84a000, 376832, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x8a000) = 0x7f741c84a000

mmap(0x7f741c8a6000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xe5000) = 0x7f741c8a6000

close(3) = 0

openat(AT\_FDCWD, "/usr/lib/libgcc\_s.so.1", O\_RDONLY|O\_CLOEXEC) = 3

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

newfstatat(3, "", {st\_mode=S\_IFREG|0644, st\_size=478272, ...}, AT\_EMPTY\_PATH) = 0

mmap(NULL, 107240, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f741c7a5000

mprotect(0x7f741c7a8000, 90112, PROT\_NONE) = 0

mmap(0x7f741c7a8000, 73728, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x3000) = 0x7f741c7a8000

mmap(0x7f741c7ba000, 12288, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x15000) = 0x7f741c7ba000

mmap(0x7f741c7be000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x18000) = 0x7f741c7be000

close(3) = 0

openat(AT\_FDCWD, "/usr/lib/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\320\324\2\0\0\0\0\0"..., 832) = 832

pread64(3, "\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0"..., 784, 64) = 784

pread64(3, "\4\0\0\0@\0\0\0\5\0\0\0GNU\0\2\0\0\300\4\0\0\0\3\0\0\0\0\0\0\0"..., 80, 848) = 80

pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\205vn\235\204X\261n\234|\346\340|q,\2"..., 68, 928) = 68

newfstatat(3, "", {st\_mode=S\_IFREG|0755, st\_size=2463384, ...}, AT\_EMPTY\_PATH) = 0

pread64(3, "\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0"..., 784, 64) = 784

mmap(NULL, 2136752, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f741c59b000

mprotect(0x7f741c5c7000, 1880064, PROT\_NONE) = 0

mmap(0x7f741c5c7000, 1531904, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x2c000) = 0x7f741c5c7000

mmap(0x7f741c73d000, 344064, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1a2000) = 0x7f741c73d000

mmap(0x7f741c792000, 24576, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1f6000) = 0x7f741c792000

mmap(0x7f741c798000, 51888, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7f741c798000

close(3) = 0

openat(AT\_FDCWD, "/usr/lib/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 \320\0\0\0\0\0\0"..., 832) = 832

newfstatat(3, "", {st\_mode=S\_IFREG|0755, st\_size=362968, ...}, AT\_EMPTY\_PATH) = 0

mmap(NULL, 365576, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f741c541000

mmap(0x7f741c54e000, 233472, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xd000) = 0x7f741c54e000

mmap(0x7f741c587000, 73728, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x46000) = 0x7f741c587000

mmap(0x7f741c599000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x57000) = 0x7f741c599000

close(3) = 0

openat(AT\_FDCWD, "/usr/lib/libpgm-5.3.so.0", O\_RDONLY|O\_CLOEXEC) = 3

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

newfstatat(3, "", {st\_mode=S\_IFREG|0755, st\_size=340376, ...}, AT\_EMPTY\_PATH) = 0

mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7f741c53f000

mmap(NULL, 328976, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f741c4ee000

mprotect(0x7f741c4f2000, 290816, PROT\_NONE) = 0

mmap(0x7f741c4f2000, 167936, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x4000) = 0x7f741c4f2000

mmap(0x7f741c51b000, 118784, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x2d000) = 0x7f741c51b000

mmap(0x7f741c539000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x4a000) = 0x7f741c539000

mmap(0x7f741c53b000, 13584, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7f741c53b000

close(3) = 0

openat(AT\_FDCWD, "/usr/lib/librt.so.1", O\_RDONLY|O\_CLOEXEC) = 3

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

newfstatat(3, "", {st\_mode=S\_IFREG|0755, st\_size=16200, ...}, AT\_EMPTY\_PATH) = 0

mmap(NULL, 16400, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f741c4e9000

mmap(0x7f741c4ea000, 4096, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1000) = 0x7f741c4ea000

mmap(0x7f741c4eb000, 4096, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x2000) = 0x7f741c4eb000

mmap(0x7f741c4ec000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x2000) = 0x7f741c4ec000

close(3) = 0

openat(AT\_FDCWD, "/usr/lib/libpthread.so.0", O\_RDONLY|O\_CLOEXEC) = 3

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

newfstatat(3, "", {st\_mode=S\_IFREG|0755, st\_size=16488, ...}, AT\_EMPTY\_PATH) = 0

mmap(NULL, 16400, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f741c4e4000

mmap(0x7f741c4e5000, 4096, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1000) = 0x7f741c4e5000

mmap(0x7f741c4e6000, 4096, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x2000) = 0x7f741c4e6000

mmap(0x7f741c4e7000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x2000) = 0x7f741c4e7000

close(3) = 0

mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7f741c4e2000

mmap(NULL, 12288, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7f741c4df000

arch\_prctl(ARCH\_SET\_FS, 0x7f741c4df980) = 0

set\_tid\_address(0x7f741c4dfc50) = 9677

set\_robust\_list(0x7f741c4dfc60, 24) = 0

rseq(0x7f741c4e0320, 0x20, 0, 0x53053053) = 0

mprotect(0x7f741c792000, 12288, PROT\_READ) = 0

mprotect(0x7f741c4e7000, 4096, PROT\_READ) = 0

mprotect(0x7f741c4ec000, 4096, PROT\_READ) = 0

mprotect(0x7f741c8a6000, 4096, PROT\_READ) = 0

mprotect(0x7f741c539000, 4096, PROT\_READ) = 0

mprotect(0x7f741c599000, 4096, PROT\_READ) = 0

mprotect(0x7f741c7be000, 4096, PROT\_READ) = 0

mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7f741c4dd000

mprotect(0x7f741cabd000, 53248, PROT\_READ) = 0

mprotect(0x7f741cb7c000, 32768, PROT\_READ) = 0

mprotect(0x55723dd2c000, 4096, PROT\_READ) = 0

mprotect(0x7f741cbf5000, 8192, PROT\_READ) = 0

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

munmap(0x7f741cb87000, 230043) = 0

getrandom("\xbe\xea\xb3\xc4\x2f\x3e\x79\x2a", 8, GRND\_NONBLOCK) = 8

brk(NULL) = 0x55723e2c0000

brk(0x55723e2e1000) = 0x55723e2e1000

futex(0x7f741cacb6bc, FUTEX\_WAKE\_PRIVATE, 2147483647) = 0

futex(0x7f741cacb6c8, FUTEX\_WAKE\_PRIVATE, 2147483647) = 0

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

read(3, "0-7\n", 1024) = 4

close(3) = 0

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

newfstatat(3, "", {st\_mode=S\_IFDIR|0755, st\_size=0, ...}, AT\_EMPTY\_PATH) = 0

getdents64(3, 0x55723e2d1ee0 /\* 26 entries \*/, 32768) = 752

getdents64(3, 0x55723e2d1ee0 /\* 0 entries \*/, 32768) = 0

close(3) = 0

getpid() = 9677

sched\_getaffinity(9677, 128, [0, 1, 2, 3, 4, 5, 6, 7]) = 40

newfstatat(AT\_FDCWD, "/etc/nsswitch.conf", {st\_mode=S\_IFREG|0644, st\_size=391, ...}, 0) = 0

newfstatat(AT\_FDCWD, "/", {st\_mode=S\_IFDIR|0755, st\_size=4096, ...}, 0) = 0

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

newfstatat(3, "", {st\_mode=S\_IFREG|0644, st\_size=391, ...}, AT\_EMPTY\_PATH) = 0

read(3, "# Name Service Switch configurat"..., 4096) = 391

read(3, "", 4096) = 0

newfstatat(3, "", {st\_mode=S\_IFREG|0644, st\_size=391, ...}, AT\_EMPTY\_PATH) = 0

close(3) = 0

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

newfstatat(3, "", {st\_mode=S\_IFREG|0644, st\_size=3171, ...}, AT\_EMPTY\_PATH) = 0

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

read(3, "# Full data: /usr/share/iana-etc"..., 4096) = 3171

close(3) = 0

eventfd2(0, EFD\_CLOEXEC) = 3

fcntl(3, F\_GETFL) = 0x2 (flags O\_RDWR)

fcntl(3, F\_SETFL, O\_RDWR|O\_NONBLOCK) = 0

fcntl(3, F\_GETFL) = 0x802 (flags O\_RDWR|O\_NONBLOCK)

fcntl(3, F\_SETFL, O\_RDWR|O\_NONBLOCK) = 0

getpid() = 9677

getpid() = 9677

getrandom("\x8d\xad\xf5\x6d\x7d\x05\x76\xed\x86\x38\x8f\xa5\xc2\x3b\xc3\x34", 16, 0) = 16

getrandom("\x05\x2f\x49\xb5\xe8\x90\x1a\x49\xb9\xe0\xb8\x4a\xb7\x2f\x5b\x64", 16, 0) = 16

eventfd2(0, EFD\_CLOEXEC) = 4

fcntl(4, F\_GETFL) = 0x2 (flags O\_RDWR)

fcntl(4, F\_SETFL, O\_RDWR|O\_NONBLOCK) = 0

fcntl(4, F\_GETFL) = 0x802 (flags O\_RDWR|O\_NONBLOCK)

fcntl(4, F\_SETFL, O\_RDWR|O\_NONBLOCK) = 0

getpid() = 9677

epoll\_create1(EPOLL\_CLOEXEC) = 5

epoll\_ctl(5, EPOLL\_CTL\_ADD, 4, {events=0, data={u32=1043147360, u64=93949157780064}}) = 0

epoll\_ctl(5, EPOLL\_CTL\_MOD, 4, {events=EPOLLIN, data={u32=1043147360, u64=93949157780064}}) = 0

getpid() = 9677

rt\_sigaction(SIGRT\_1, {sa\_handler=0x7f741c625940, sa\_mask=[], sa\_flags=SA\_RESTORER|SA\_ONSTACK|SA\_RESTART|SA\_SIGINFO, sa\_restorer=0x7f741c5dd560}, NULL, 8) = 0

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

mmap(NULL, 8392704, PROT\_NONE, MAP\_PRIVATE|MAP\_ANONYMOUS|MAP\_STACK, -1, 0) = 0x7f741bcdc000

mprotect(0x7f741bcdd000, 8388608, PROT\_READ|PROT\_WRITE) = 0

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

clone(child\_stack=0x7f741c4dbcb0, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, parent\_tid=[9678], tls=0x7f741c4dc640, child\_tidptr=0x7f741c4dc910) = 9678

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

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

getpid() = 9677

epoll\_create1(EPOLL\_CLOEXEC) = 7

epoll\_ctl(7, EPOLL\_CTL\_ADD, 6, {events=0, data={u32=1043145888, u64=93949157778592}}) = 0

epoll\_ctl(7, EPOLL\_CTL\_MOD, 6, {events=EPOLLIN, data={u32=1043145888, u64=93949157778592}}) = 0

mmap(NULL, 8392704, PROT\_NONE, MAP\_PRIVATE|MAP\_ANONYMOUS|MAP\_STACK, -1, 0) = 0x7f741b4db000

mprotect(0x7f741b4dc000, 8388608, PROT\_READ|PROT\_WRITE) = 0

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

clone(child\_stack=0x7f741bcdacb0, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, parent\_tid=[9679], tls=0x7f741bcdb640, child\_tidptr=0x7f741bcdb910) = 9679

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

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

getpid() = 9677

getpid() = 9677

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=9677, nl\_groups=00000000}, [12]) = 0

sendto(9, [{nlmsg\_len=20, nlmsg\_type=RTM\_GETLINK, nlmsg\_flags=NLM\_F\_REQUEST|NLM\_F\_DUMP, nlmsg\_seq=1648280464, nlmsg\_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=[[{nlmsg\_len=1320, nlmsg\_type=RTM\_NEWLINK, nlmsg\_flags=NLM\_F\_MULTI, nlmsg\_seq=1648280464, nlmsg\_pid=9677}, {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}, 00:00:00:00:00:00], [{nla\_len=10, nla\_type=IFLA\_BROADCAST}, 00:00:00:00:00:00], [{nla\_len=196, nla\_type=IFLA\_STATS64}, {rx\_packets=3536, tx\_packets=3536, rx\_bytes=232023, tx\_bytes=232023, 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=3536, tx\_packets=3536, rx\_bytes=232023, tx\_bytes=232023, 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] = 1, [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] = 0, [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=119, reachable\_time=25594, 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] = 173, [IPSTATS\_MIB\_INOCTETS] = 15413, [IPSTATS\_MIB\_INDELIVERS] = 173, [IPSTATS\_MIB\_OUTFORWDATAGRAMS] = 0, [IPSTATS\_MIB\_OUTPKTS] = 173, [IPSTATS\_MIB\_OUTOCTETS] = 15413, [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]]]]]]], [{nlmsg\_len=1356, nlmsg\_type=RTM\_NEWLINK, nlmsg\_flags=NLM\_F\_MULTI, nlmsg\_seq=1648280464, nlmsg\_pid=9677}, {ifi\_family=AF\_UNSPEC, ifi\_type=ARPHRD\_ETHER, ifi\_index=if\_nametoindex("wlo1"), ifi\_flags=IFF\_UP|IFF\_BROADCAST|IFF\_RUNNING|IFF\_MULTICAST|IFF\_LOWER\_UP, ifi\_change=0}, [[{nla\_len=9, nla\_type=IFLA\_IFNAME}, "wlo1"], [{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}, 2], [{nla\_len=8, nla\_type=IFLA\_CARRIER\_UP\_COUNT}, 1], [{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}, 5c:80:b6:b8:01:f0], [{nla\_len=10, nla\_type=IFLA\_BROADCAST}, ff:ff:ff:ff:ff:ff], [{nla\_len=196, nla\_type=IFLA\_STATS64}, {rx\_packets=200374, tx\_packets=72363, rx\_bytes=270279625, tx\_bytes=10527493, rx\_errors=0, tx\_errors=0, rx\_dropped=100, 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=200374, tx\_packets=72363, rx\_bytes=270279625, tx\_bytes=10527493, rx\_errors=0, tx\_errors=0, rx\_dropped=100, 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=IFLA\_PERM\_ADDRESS}, 5c:80:b6:b8:01:f0], [{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] = 1, [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] = 0, [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=724, reachable\_time=29730, 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] = 13, [IPSTATS\_MIB\_INOCTETS] = 1048, [IPSTATS\_MIB\_INDELIVERS] = 0, [IPSTATS\_MIB\_OUTFORWDATAGRAMS] = 0, [IPSTATS\_MIB\_OUTPKTS] = 23, [IPSTATS\_MIB\_OUTOCTETS] = 1520, [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] = 13, [IPSTATS\_MIB\_OUTMCASTPKTS] = 23, [IPSTATS\_MIB\_INBCASTPKTS] = 0, [IPSTATS\_MIB\_OUTBCASTPKTS] = 0, [IPSTATS\_MIB\_INMCASTOCTETS] = 1048, [IPSTATS\_MIB\_OUTMCASTOCTETS] = 1520, [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] = 19, [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]]]]], [{nla\_len=20, nla\_type=NLA\_F\_NESTED|IFLA\_PROP\_LIST}, [{nla\_len=14, nla\_type=IFLA\_ALT\_IFNAME}, "wlp0s20f3"]]]]], iov\_len=4096}], msg\_iovlen=1, msg\_controllen=0, msg\_flags=0}, 0) = 2676

recvmsg(9, {msg\_name={sa\_family=AF\_NETLINK, nl\_pid=0, nl\_groups=00000000}, msg\_namelen=12, msg\_iov=[{iov\_base=[{nlmsg\_len=1752, nlmsg\_type=RTM\_NEWLINK, nlmsg\_flags=NLM\_F\_MULTI, nlmsg\_seq=1648280464, nlmsg\_pid=9677}, {ifi\_family=AF\_UNSPEC, ifi\_type=ARPHRD\_ETHER, ifi\_index=if\_nametoindex("docker0"), ifi\_flags=IFF\_UP|IFF\_BROADCAST|IFF\_MULTICAST, ifi\_change=0}, [[{nla\_len=12, nla\_type=IFLA\_IFNAME}, "docker0"], [{nla\_len=8, nla\_type=IFLA\_TXQLEN}, 0], [{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}, 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}, 0], [{nla\_len=12, nla\_type=IFLA\_QDISC}, "noqueue"], [{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}, 02:42:54:1b:0a:7f], [{nla\_len=10, nla\_type=IFLA\_BROADCAST}, ff:ff:ff:ff:ff:ff], [{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=428, nla\_type=IFLA\_LINKINFO}, [[{nla\_len=11, nla\_type=IFLA\_INFO\_KIND}, "bridge"], [{nla\_len=412, nla\_type=IFLA\_INFO\_DATA}, [[{nla\_len=12, nla\_type=IFLA\_BR\_HELLO\_TIMER}, 0], [{nla\_len=12, nla\_type=IFLA\_BR\_TCN\_TIMER}, 0], [{nla\_len=12, nla\_type=IFLA\_BR\_TOPOLOGY\_CHANGE\_TIMER}, 0], [{nla\_len=12, nla\_type=IFLA\_BR\_GC\_TIMER}, 28937 /\* 289.37 s \*/], [{nla\_len=8, nla\_type=IFLA\_BR\_FORWARD\_DELAY}, 1499 /\* 14.99 s \*/], [{nla\_len=8, nla\_type=IFLA\_BR\_HELLO\_TIME}, 199 /\* 1.99 s \*/], [{nla\_len=8, nla\_type=IFLA\_BR\_MAX\_AGE}, 1999 /\* 19.99 s \*/], [{nla\_len=8, nla\_type=IFLA\_BR\_AGEING\_TIME}, 29999 /\* 299.99 s \*/], [{nla\_len=8, nla\_type=IFLA\_BR\_STP\_STATE}, 0], [{nla\_len=6, nla\_type=IFLA\_BR\_PRIORITY}, 32768], [{nla\_len=5, nla\_type=IFLA\_BR\_VLAN\_FILTERING}, 0], [{nla\_len=6, nla\_type=IFLA\_BR\_GROUP\_FWD\_MASK}, 0], [{nla\_len=12, nla\_type=IFLA\_BR\_BRIDGE\_ID}, {prio=[128, 0], addr=02:42:54:1b:0a:7f}], [{nla\_len=12, nla\_type=IFLA\_BR\_ROOT\_ID}, {prio=[128, 0], addr=02:42:54:1b:0a:7f}], [{nla\_len=6, nla\_type=IFLA\_BR\_ROOT\_PORT}, 0], [{nla\_len=8, nla\_type=IFLA\_BR\_ROOT\_PATH\_COST}, 0], [{nla\_len=5, nla\_type=IFLA\_BR\_TOPOLOGY\_CHANGE}, 0], [{nla\_len=5, nla\_type=IFLA\_BR\_TOPOLOGY\_CHANGE\_DETECTED}, 0], [{nla\_len=10, nla\_type=IFLA\_BR\_GROUP\_ADDR}, 01:80:c2:00:00:00], [{nla\_len=12, nla\_type=IFLA\_BR\_MULTI\_BOOLOPT}, {optval=0, optmask=1<<BR\_BOOLOPT\_NO\_LL\_LEARN}], [{nla\_len=6, nla\_type=IFLA\_BR\_VLAN\_PROTOCOL}, htons(ETH\_P\_8021Q)], [{nla\_len=6, nla\_type=IFLA\_BR\_VLAN\_DEFAULT\_PVID}, 1], [{nla\_len=5, nla\_type=IFLA\_BR\_VLAN\_STATS\_ENABLED}, 0], [{nla\_len=5, nla\_type=IFLA\_BR\_VLAN\_STATS\_PER\_PORT}, 0], [{nla\_len=5, nla\_type=IFLA\_BR\_MCAST\_ROUTER}, 1], [{nla\_len=5, nla\_type=IFLA\_BR\_MCAST\_SNOOPING}, 1], [{nla\_len=5, nla\_type=IFLA\_BR\_MCAST\_QUERY\_USE\_IFADDR}, 0], [{nla\_len=5, nla\_type=IFLA\_BR\_MCAST\_QUERIER}, 0], [{nla\_len=5, nla\_type=IFLA\_BR\_MCAST\_STATS\_ENABLED}, 0], [{nla\_len=8, nla\_type=IFLA\_BR\_MCAST\_HASH\_ELASTICITY}, 16], [{nla\_len=8, nla\_type=IFLA\_BR\_MCAST\_HASH\_MAX}, 4096], [{nla\_len=8, nla\_type=IFLA\_BR\_MCAST\_LAST\_MEMBER\_CNT}, 2], ...]]]], [{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] = 1, [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] = 0, [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=811, reachable\_time=35664, 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\_EUI64]]]]]]], iov\_len=4096}], msg\_iovlen=1, msg\_controllen=0, msg\_flags=0}, 0) = 1752

recvmsg(9, {msg\_name={sa\_family=AF\_NETLINK, nl\_pid=0, nl\_groups=00000000}, msg\_namelen=12, msg\_iov=[{iov\_base=[{nlmsg\_len=20, nlmsg\_type=NLMSG\_DONE, nlmsg\_flags=NLM\_F\_MULTI, nlmsg\_seq=1648280464, nlmsg\_pid=9677}, 0], iov\_len=4096}], msg\_iovlen=1, msg\_controllen=0, msg\_flags=0}, 0) = 20

sendto(9, [{nlmsg\_len=20, nlmsg\_type=RTM\_GETADDR, nlmsg\_flags=NLM\_F\_REQUEST|NLM\_F\_DUMP, nlmsg\_seq=1648280465, nlmsg\_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=[[{nlmsg\_len=76, nlmsg\_type=RTM\_NEWADDR, nlmsg\_flags=NLM\_F\_MULTI, nlmsg\_seq=1648280465, nlmsg\_pid=9677}, {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=119, tstamp=119}]]], [{nlmsg\_len=88, nlmsg\_type=RTM\_NEWADDR, nlmsg\_flags=NLM\_F\_MULTI, nlmsg\_seq=1648280465, nlmsg\_pid=9677}, {ifa\_family=AF\_INET, ifa\_prefixlen=24, ifa\_flags=0, ifa\_scope=RT\_SCOPE\_UNIVERSE, ifa\_index=if\_nametoindex("wlo1")}, [[{nla\_len=8, nla\_type=IFA\_ADDRESS}, inet\_addr("192.168.1.137")], [{nla\_len=8, nla\_type=IFA\_LOCAL}, inet\_addr("192.168.1.137")], [{nla\_len=8, nla\_type=IFA\_BROADCAST}, inet\_addr("192.168.1.255")], [{nla\_len=9, nla\_type=IFA\_LABEL}, "wlo1"], [{nla\_len=8, nla\_type=IFA\_FLAGS}, IFA\_F\_NOPREFIXROUTE], [{nla\_len=20, nla\_type=IFA\_CACHEINFO}, {ifa\_prefered=16774, ifa\_valid=16774, cstamp=726, tstamp=739}]]], [{nlmsg\_len=88, nlmsg\_type=RTM\_NEWADDR, nlmsg\_flags=NLM\_F\_MULTI, nlmsg\_seq=1648280465, nlmsg\_pid=9677}, {ifa\_family=AF\_INET, ifa\_prefixlen=16, ifa\_flags=IFA\_F\_PERMANENT, ifa\_scope=RT\_SCOPE\_UNIVERSE, ifa\_index=if\_nametoindex("docker0")}, [[{nla\_len=8, nla\_type=IFA\_ADDRESS}, inet\_addr("172.17.0.1")], [{nla\_len=8, nla\_type=IFA\_LOCAL}, inet\_addr("172.17.0.1")], [{nla\_len=8, nla\_type=IFA\_BROADCAST}, inet\_addr("172.17.255.255")], [{nla\_len=12, nla\_type=IFA\_LABEL}, "docker0"], [{nla\_len=8, nla\_type=IFA\_FLAGS}, IFA\_F\_PERMANENT], [{nla\_len=20, nla\_type=IFA\_CACHEINFO}, {ifa\_prefered=4294967295, ifa\_valid=4294967295, cstamp=811, tstamp=811}]]]], iov\_len=4096}], msg\_iovlen=1, msg\_controllen=0, msg\_flags=0}, 0) = 252

recvmsg(9, {msg\_name={sa\_family=AF\_NETLINK, nl\_pid=0, nl\_groups=00000000}, msg\_namelen=12, msg\_iov=[{iov\_base=[[{nlmsg\_len=72, nlmsg\_type=RTM\_NEWADDR, nlmsg\_flags=NLM\_F\_MULTI, nlmsg\_seq=1648280465, nlmsg\_pid=9677}, {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=119, tstamp=119}], [{nla\_len=8, nla\_type=IFA\_FLAGS}, IFA\_F\_PERMANENT]]], [{nlmsg\_len=72, nlmsg\_type=RTM\_NEWADDR, nlmsg\_flags=NLM\_F\_MULTI, nlmsg\_seq=1648280465, nlmsg\_pid=9677}, {ifa\_family=AF\_INET6, ifa\_prefixlen=64, ifa\_flags=IFA\_F\_PERMANENT, ifa\_scope=RT\_SCOPE\_LINK, ifa\_index=if\_nametoindex("wlo1")}, [[{nla\_len=20, nla\_type=IFA\_ADDRESS}, inet\_pton(AF\_INET6, "fe80::37d0:fc9b:b6d4:8b78")], [{nla\_len=20, nla\_type=IFA\_CACHEINFO}, {ifa\_prefered=4294967295, ifa\_valid=4294967295, cstamp=724, tstamp=826}], [{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=[{nlmsg\_len=20, nlmsg\_type=NLMSG\_DONE, nlmsg\_flags=NLM\_F\_MULTI, nlmsg\_seq=1648280465, nlmsg\_pid=9677}, 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(8080), sin\_addr=inet\_addr("127.0.0.1")}, 16) = 0

listen(9, 100) = 0

getsockname(9, {sa\_family=AF\_INET, sin\_port=htons(8080), sin\_addr=inet\_addr("127.0.0.1")}, [128 => 16]) = 0

getsockname(9, {sa\_family=AF\_INET, sin\_port=htons(8080), sin\_addr=inet\_addr("127.0.0.1")}, [128 => 16]) = 0

getpid() = 9677

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

getpid() = 9677

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

newfstatat(1, "", {st\_mode=S\_IFCHR|0620, st\_rdev=makedev(0x88, 0x5), ...}, AT\_EMPTY\_PATH) = 0

write(1, "> ", 2> ) = 2

newfstatat(0, "", {st\_mode=S\_IFCHR|0620, st\_rdev=makedev(0x88, 0x5), ...}, AT\_EMPTY\_PATH) = 0

read(0, create 10

"create 10\n", 1024) = 10

clone(child\_stack=NULL, flags=CLONE\_CHILD\_CLEARTID|CLONE\_CHILD\_SETTID|SIGCHLD, child\_tidptr=0x7f741c4dfc50) = 9698

write(1, "Ok: 9698\n", 9Ok: 9698

) = 9

write(1, "> ", 2> ) = 2

read(0, create 20

"create 20\n", 1024) = 10

getpid() = 9677

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

getpid() = 9677

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

getpid() = 9677

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

getpid() = 9677

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

getpid() = 9677

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

getpid() = 9677

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

getpid() = 9677

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

getpid() = 9677

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

getpid() = 9677

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

getpid() = 9677

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

getpid() = 9677

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

write(1, "Ok: 9724\n", 9Ok: 9724

) = 9

write(1, "> ", 2> ) = 2

read(0, exec 10 abraasabra abra

"exec 10 abraasabra abra\n", 1024) = 24

getpid() = 9677

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

getpid() = 9677

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

getpid() = 9677

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

getpid() = 9677

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

getpid() = 9677

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

write(1, "Ok: 10: [0 6]\n", 14Ok: 10: [0 6]

) = 14

write(1, "> ", 2> ) = 2

read(0, pingid 10

"pingid 10\n", 1024) = 10

getpid() = 9677

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

getpid() = 9677

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

getpid() = 9677

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

getpid() = 9677

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

getpid() = 9677

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

write(1, "Ok: 1\n", 6Ok: 1

) = 6

write(1, "> ", 2> ) = 2

read(0, pingid 12

"pingid 12\n", 1024) = 10

getpid() = 9677

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

getpid() = 9677

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

getpid() = 9677

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

getpid() = 9677

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

getpid() = 9677

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

write(1, "Ok: 0\n", 6Ok: 0

) = 6

write(1, "> ", 2> ) = 2

read(0, exit

"exit\n", 1024) = 5

getpid() = 9677

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

getpid() = 9677

getpid() = 9677

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

getpid() = 9677

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

getpid() = 9677

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

futex(0x7f741bcdb910, FUTEX\_WAIT\_BITSET|FUTEX\_CLOCK\_REALTIME, 9679, NULL, FUTEX\_BITSET\_MATCH\_ANY) = -1 EAGAIN (Resource temporarily unavailable)

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 +++

[suraba04@asusx512fl third]$

## 

## **Демонстрация работы программы**

[suraba04@asusx512fl third]$ ./client

> create 1

Ok: 3965

> create 2

Ok: 3986

> create 3

Ok: 4000

> create 4

Ok: 4005

> create 5

Ok: 4009

> cl;ak

Error Command

> exec 1 hlasghljh as

Ok: 1: [2]

> exec 2 abracadabra abra

Ok: 2: [0 7]

> exec 3 ComebackpapaFranku a

Ok: 3: [5 9 11 14]

> exec 6 alsjh lhjg

Error: 6: Not found

> exec 5 lasjgh abab

Ok: 5: [-1]

> pingid 1

Ok: 1

> pingid 3

Ok: 1

> exit

## 

## **Вывод**

Данная лабораторная работа оказалась самой сложной частью курса, даже отчасти сложнее моего курсового проекта.

Однако она является и самой полезной. Ее польза состоит в том, что она соединяет все элементы курса в одном проекте. Хотя я и не использовал потоки, но их тоже можно было добавить.

Выполнение ЛР побудило меня разобраться в zmq, она имеет хорошую документации и подробный гайд, поэтому я хочу использовать данную библиотеку и в курсовом проекте.

Больше всего понравилось строить схему общения (архитектуру) между серверами и клиентом. Долго думал как и какие паттерны использовать, чтобы программа работала максимально логично, однако остановился на req-rep. Но, считаю, можно было использовать pub-sub, однако нативно у него нет свойства ответа.

Также для выполнения специального задания (поиск подстроки в строке) я решил использовать какой-нибудь интересный алгоритм, осатновился на алгоритме КМП, так как я его писал еще в первом семестре.