Московский Авиационный Институт

(Национальный Исследовательский Университет)

Факультет информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

**Лабораторные работы №6-8 по курсу**

**«Операционные системы»**

**Управление серверами сообщений, применение отложенных вычислений, интеграция программных систем друг с другом.**

Студент: Ивенкова Л.В.

Группа: М80 – 208Б-19

Вариант: 1

Преподаватель: Миронов Е. С.

Дата: \_\_\_\_\_\_\_\_\_\_\_

Оценка: \_\_\_\_\_\_\_\_\_\_\_

Подпись: \_\_\_\_\_\_\_\_\_\_\_

Москва, 2021

1. **Постановка задачи**

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

Управляющий узел отвечает за ввод команд от пользователя и отправку этих команд на вычислительные узлы.

1. **Общие сведения о программе**

Программа написана на языке С++ на операционной системе Ubuntu. В программе используется очередь сообщений ZeroMQ.

Программа поддерживает следующие команды:

* create [id] [parent\_id] – создать новый узел [id], родителем которого является узел [parent\_id]. Если [parent\_id] = -1, то родительский узел – управляющий.
* remove [id] – удалить узел [id]. Все дочерние узлы будут также удалены.
* exec [id] n [k1..kn] – сложить n чисел
* pingall – проверить доступность узлов. Будет выведен список всех недоступных на данный момент узлов.
* exit – выйти из программы.

1. **Общий метод и алгоритм решения**

В программе используется тип соединения Request-Response. Узлы передают информацию друг другу при помощи очереди сообщений. Все сообщения имеют следующий вид:

[id узла, которому предназначено сообщение] [команда] [аргументы]

Управляющий узел хранит структуру «список списков», в которую записывает id существующих узлов. При помощи этой структуры он определяет, в какой список нужно направить сообщение.

Вычислительный узел, получив сообщение, сравнивает свой id и id из сообщения. Если они совпадают, то узел начинает обрабатывать запрос, в противном случае узел направляет это же сообщение своему ребенку и ждет от него ответа.

Для удобства функции отправки и получения сообщений, а также функции для подключения к сокетам вынесены в отдельный заголовочный файл, который подключается к программам узлов.

Для проверки доступности узлов используется контейнер std::list<std::list<int>>, который хранит все существующие id узлов .Управляющий узел отправляет запрос всем спискам узлов и получает в ответ строку с id всех доступных узлов списка. С помощью erase они исключаются из общего списка. Оставшиеся выводятся на экран, или же, если их нет, -1.

1. **Основные файлы программы**

***topology.h***

#include <list>

#include <set>

#include <stdexcept>

// Тип топологии: 1

class Topology {

private:

std::list<std::list<int>> container;

public:

// Добавление нового узла

void Insert(int id, int parent\_id) {

if (parent\_id == -1) {

std::list<int> new\_list;

new\_list.push\_back(id);

container.push\_back(new\_list);

}

else {

int list\_id = Find(parent\_id);

if (list\_id == -1) {

throw std::runtime\_error("Wrong parent id");

}

auto it1 = container.begin();

std::advance(it1, list\_id);

for (auto it2 = it1->begin(); it2 != it1->end(); ++it2) {

if (\*it2 == parent\_id) {

it1->insert(++it2, id);

return;

}

}

}

}

// Поиск узла с заданным id в списке списков

int Find(int id) {

int cur\_list\_id = 0;

for (auto it1 = container.begin(); it1 != container.end(); ++it1) {

for (auto it2 = it1->begin(); it2 != it1->end(); ++it2) {

if (\*it2 == id) {

return cur\_list\_id;

}

}

++cur\_list\_id;

}

return -1;

}

// Удаление узла с указанным id

void Erase(int id) {

int list\_id = Find(id);

if (list\_id == -1) {

throw std::runtime\_error("Wrong id");

}

auto it1 = container.begin();

std::advance(it1, list\_id); // Изменяет переданный итератор

for (auto it2 = it1->begin(); it2 != it1->end(); ++it2) {

if (\*it2 == id) {

it1->erase(it2, it1->end());

if (it1->empty()) {

container.erase(it1);

}

return;

}

}

}

// Получение первого id узла в контейнере

int GetFirstId(int list\_id) {

auto it1 = container.begin();

std::advance(it1, list\_id);

if (it1->begin() == it1->end()) {

return -1;

}

return \*(it1->begin());

}

std::set<int> SetAllNodes() {

std::set<int> all\_id;

for (auto it1 = container.begin(); it1 != container.end(); ++it1) {

for (auto it2 = it1->begin(); it2 != it1->end(); ++it2) {

all\_id.insert(\*it2);

}

}

return all\_id;

}

};

***zmq\_functions.h***

#include <zmq.hpp>

#include <iostream>

#include <string>

const int MAIN\_PORT = 4040;

void send\_message(zmq::socket\_t &socket, const std::string &msg) {

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

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

socket.send(message);

}

std::string receive\_message(zmq::socket\_t &socket) {

zmq::message\_t message;

int chars\_read;

try {

chars\_read = (int)socket.recv(&message);

}

catch (...) {

chars\_read = 0;

}

if (chars\_read == 0) {

return "Error ......";

}

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

return received\_msg;

}

void connect(zmq::socket\_t &socket, int id) {

std::string adress = "tcp://127.0.0.1:" + std::to\_string(MAIN\_PORT + id);

socket.connect(adress);

}

// Отключение клиента

void disconnect(zmq::socket\_t &socket, int id) {

std::string adress = "tcp://127.0.0.1:" + std::to\_string(MAIN\_PORT + id);

socket.disconnect(adress);

}

// Связка узла с сокетом

void bind(zmq::socket\_t &socket, int id) {

std::string adress = "tcp://127.0.0.1:" + std::to\_string(MAIN\_PORT + id);

socket.bind(adress);

}

// Освобождение узла от сокета

void unbind(zmq::socket\_t &socket, int id) {

std::string adress = "tcp://127.0.0.1:" + std::to\_string(MAIN\_PORT + id);

socket.unbind(adress);

}

***control.cpp***

#include <iostream>

#include <unistd.h>

#include <sstream>

#include <set>

#include <zmq.hpp>

#include <chrono>

#include <vector>

#include "topology.h"

#include "zmq\_functions.h"

int main() {

Topology network;

std::vector<zmq::socket\_t> branches;

zmq::context\_t context;

std::string comand;

std::string message;

std::set<int> all\_nodes;

while (std::cin >> comand) {

if (comand == "create") {

int node\_id, parent\_id;

std::cin >> node\_id >> parent\_id;

if (network.Find(node\_id) != -1) { // Поиск id выч. узла среди существующих

std::cout << "Error: already exists!\n";

} else if (parent\_id == -1) {

pid\_t pid = fork(); // Создание дочернего узла

if (pid < 0) {

perror("Can't create new process!\n");

exit(EXIT\_FAILURE);

}

if (pid == 0) {

execl("./count", "./count", std::to\_string(node\_id).c\_str(), NULL);

perror("Can't execute new process!\n");

exit(EXIT\_FAILURE);

}

branches.emplace\_back(context, ZMQ\_REQ);

branches[branches.size() - 1].setsockopt(ZMQ\_SNDTIMEO, 5000);

bind(branches[branches.size() - 1], node\_id);

send\_message(branches[branches.size() - 1], std::to\_string(node\_id) + "pid");

std::string reply = receive\_message(branches[branches.size() - 1]);

std::cout << reply << "\n";

network.Insert(node\_id, parent\_id);

} else if (network.Find(parent\_id) == -1) {

std::cout << "Error: parent not found!\n";

} else {

int branch = network.Find(parent\_id);

send\_message(branches[branch], std::to\_string(parent\_id) + "create " + std::to\_string(node\_id));

std::string reply = receive\_message(branches[branch]);

std::cout << reply << "\n";

network.Insert(node\_id, parent\_id);

}

} else if (comand == "remove") {

int id;

std::cin >> id;

int branch = network.Find(id); // Проверка, существует ли узел

if (branch == -1) {

std::cout << "Error: incorrect node id!\n";

} else {

bool is\_first = (network.GetFirstId(branch) == id);

send\_message(branches[branch], std::to\_string(id) + " remove");

std::string reply = receive\_message(branches[branch]);

std::cout << reply << "\n";

network.Erase(id);

if (is\_first) {

unbind(branches[branch], id);

branches.erase(branches.begin() + branch);

}

}

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

int dest\_id;

int n;

std::cin >> dest\_id >> n;

int integer;

std::string chisla = "";

for(int i = 0; i<n; i++){

std::cin >> integer;

chisla += " " + std::to\_string(integer);

}

int branch = network.Find(dest\_id);

if (branch == -1) {

std::cout << "Error: incorrect node id!\n";

} else {

send\_message(branches[branch], std::to\_string(dest\_id) + " exec " + chisla);

std::string reply = receive\_message(branches[branch]);

std::cout << reply << "\n";

}

} else if (comand == "pingall") {

std::set<int> unavailable\_nodes = network.SetAllNodes();;

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

int first\_node\_id = network.GetFirstId(i);

send\_message(branches[i], std::to\_string(first\_node\_id) + " pingall");

std::string received\_message = receive\_message(branches[i]);

std::istringstream reply(received\_message);

int node;

while (reply >> node) {

//std::cout << node << " ";

unavailable\_nodes.erase(node);

}

}

std::cout << "OK: ";

if (!unavailable\_nodes.empty()){

for(int x: unavailable\_nodes)

std::cout << x << ", ";

}

else std::cout << "-1";

std::cout << std::endl;

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

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

int first\_node\_id = network.GetFirstId(i);

send\_message(branches[i], std::to\_string(first\_node\_id) + " remove");

std::string reply = receive\_message(branches[i]);

if (reply != "OK") {

std::cout << reply << "\n";

} else {

unbind(branches[i], first\_node\_id);

}

}

exit(0);

} else {

std::cout << "Incorrect comand!\n";

}

}

}

***count.cpp***

#include <iostream>

#include <unistd.h>

#include <sstream>

#include <set>

#include "zmq\_functions.h"

#include "topology.h"

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

if (argc != 2 && argc != 3) {

std::cout << "Wrong arguments Not enough parameters!\n";

exit(1);

}

int current\_id = std::atoi(argv[1]);

int child\_id = -1;

if (argc == 3) {

child\_id = std::atoi(argv[2]);

}

zmq::context\_t context;

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

connect(parent\_socket, current\_id);

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

child\_socket.setsockopt(ZMQ\_SNDTIMEO, 5000);

if (argc == 3) {

bind(child\_socket, child\_id);

}

std::string message;

while (1) {

message = receive\_message(parent\_socket);

std::istringstream request(message);

int dest\_id;

request >> dest\_id;

std::string comand;

request >> comand;

if (dest\_id == current\_id) {

if (comand == "pid") {

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

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

int new\_child\_id;

request >> new\_child\_id;

if (child\_id != -1) {

unbind(child\_socket, child\_id);

}

bind(child\_socket, new\_child\_id);

pid\_t pid = fork();

if (pid < 0) {

perror("Can't create new process!\n");

exit(1);

}

if (pid == 0) {

execl("./count", "./count", std::to\_string(new\_child\_id).c\_str(), std::to\_string(child\_id).c\_str(), NULL);

perror("Can't create new process!\n");

exit(1);

}

send\_message(child\_socket, std::to\_string(new\_child\_id) + "pid");

child\_id = new\_child\_id;

send\_message(parent\_socket, receive\_message(child\_socket));

} else if (comand == "remove") {

if (child\_id != -1) {

send\_message(child\_socket, std::to\_string(child\_id) + " remove");

std::string msg = receive\_message(child\_socket);

if (msg == "OK") {

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

}

unbind(child\_socket, child\_id);

disconnect(parent\_socket, current\_id);

break;

}

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

disconnect(parent\_socket, current\_id);

break;

} else if (comand == "pingall") {

std::string reply;

if (child\_id != -1) {

send\_message(child\_socket, std::to\_string(child\_id) + " pingall");

std::string msg = receive\_message(child\_socket);

reply += " " + msg;

}

send\_message(parent\_socket, std::to\_string(current\_id) + reply);

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

int integer, sum = 0;

while(request >> integer)

sum += integer;

std::string msg = "OK: " + std::to\_string(sum);

send\_message(parent\_socket, msg);

}

} else if (child\_id != -1) {

send\_message(child\_socket, message);

send\_message(parent\_socket, receive\_message(child\_socket));

if (child\_id == dest\_id && comand == "remove") {

child\_id = -1;

}

} else {

send\_message(parent\_socket, "Error: node is unavailable!\n");

}

}

}

1. **Демонстрация работы программы**

**parsifal@DESKTOP-3G70RV4:~/OS/Lab6-8$ ./control**

**create 2 -1**

**OK: 4163**

**create 3 -1**

**OK: 4168**

**create 2 -1**

**Error: already exists!**

**cerate 5 4**

**Incorrect comand!**

**Incorrect comand!**

**Incorrect comand!**

**create 5 4**

**Error: parent not found!**

**remove 3**

**OK**

**pingall**

**OK: -1**

**exec 2 5 1 2 3 4 5**

**OK: 15**

**exit**

**parsifal@DESKTOP-3G70RV4:~/OS/Lab6-8$ strace -o log.txt ./control**

**create 2 -1**

**OK: 4154**

**exit**

**parsifal@DESKTOP-3G70RV4:~/OS/Lab6-8$ cat log.txt**

**execve("./control", ["./control"], 0x7ffff9a3d9e0 /\* 27 vars \*/) = 0**

**brk(NULL) = 0x7fffe0bc4000**

**arch\_prctl(0x3001 /\* ARCH\_??? \*/, 0x7fffe8368b50) = -1 EINVAL (Invalid argument)**

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

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

**fstat(3, {st\_mode=S\_IFREG|0644, st\_size=32310, ...}) = 0**

**mmap(NULL, 32310, PROT\_READ, MAP\_PRIVATE, 3, 0) = 0x7f513ab98000**

**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) = 0x7f513abd0000**

**mmap(NULL, 678128, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f513aaf0000**

**mmap(0x7f513ab06000, 430080, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x16000) = 0x7f513ab06000**

**mmap(0x7f513ab6f000, 126976, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x7f000) = 0x7f513ab6f000**

**mmap(0x7f513ab8e000, 32768, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x9d000) = 0x7f513ab8e000**

**close(3) = 0**

**openat(AT\_FDCWD, "/lib/x86\_64-linux-gnu/libstdc++.so.6", O\_RDONLY|O\_CLOEXEC) = 3**

**read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\240\341\t\0\0\0\0\0"..., 832) = 832**

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

**mmap(NULL, 1968128, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f513a900000**

**mprotect(0x7f513a996000, 1286144, PROT\_NONE) = 0**

**mmap(0x7f513a996000, 983040, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x96000) = 0x7f513a996000**

**mmap(0x7f513aa86000, 299008, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x186000) = 0x7f513aa86000**

**mmap(0x7f513aad0000, 57344, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1cf000) = 0x7f513aad0000**

**mmap(0x7f513aade000, 10240, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7f513aade000**

**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) = 0x7f513a8e0000**

**mmap(0x7f513a8e3000, 73728, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x3000) = 0x7f513a8e3000**

**mmap(0x7f513a8f5000, 16384, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x15000) = 0x7f513a8f5000**

**mmap(0x7f513a8f9000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x18000) = 0x7f513a8f9000**

**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\363\377?\332\200\270\27\304d\245n\355Y\377\t\334"..., 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\363\377?\332\200\270\27\304d\245n\355Y\377\t\334"..., 68, 880) = 68**

**mmap(NULL, 2036952, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f513a6e0000**

**mprotect(0x7f513a705000, 1847296, PROT\_NONE) = 0**

**mmap(0x7f513a705000, 1540096, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x25000) = 0x7f513a705000**

**mmap(0x7f513a87d000, 303104, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x19d000) = 0x7f513a87d000**

**mmap(0x7f513a8c8000, 24576, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1e7000) = 0x7f513a8c8000**

**mmap(0x7f513a8ce000, 13528, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7f513a8ce000**

**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) = 0x7f513a680000**

**mmap(0x7f513a68c000, 229376, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xc000) = 0x7f513a68c000**

**mmap(0x7f513a6c4000, 73728, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x44000) = 0x7f513a6c4000**

**mmap(0x7f513a6d6000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x55000) = 0x7f513a6d6000**

**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) = 0x7f513a630000**

**mmap(0x7f513a634000, 163840, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x4000) = 0x7f513a634000**

**mmap(0x7f513a65c000, 118784, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x2c000) = 0x7f513a65c000**

**mmap(0x7f513a679000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x48000) = 0x7f513a679000**

**mmap(0x7f513a67b000, 14384, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7f513a67b000**

**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) = 0x7f513a620000**

**mmap(NULL, 1420000, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f513a4c0000**

**mmap(0x7f513a4ca000, 421888, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xa000) = 0x7f513a4ca000**

**mmap(0x7f513a531000, 217088, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x71000) = 0x7f513a531000**

**mmap(0x7f513a566000, 16384, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xa5000) = 0x7f513a566000**

**mmap(0x7f513a56a000, 723680, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7f513a56a000**

**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) = 0x7f513a470000**

**mmap(0x7f513a47b000, 204800, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xb000) = 0x7f513a47b000**

**mmap(0x7f513a4ad000, 49152, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x3d000) = 0x7f513a4ad000**

**mmap(0x7f513a4b9000, 16384, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x48000) = 0x7f513a4b9000**

**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\0O\305\3743\364B\2216\244\224\306@\261\23\327o"..., 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\0O\305\3743\364B\2216\244\224\306@\261\23\327o"..., 68, 824) = 68**

**mmap(NULL, 140408, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f513a44d000**

**mmap(0x7f513a454000, 69632, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x7000) = 0x7f513a454000**

**mmap(0x7f513a465000, 20480, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x18000) = 0x7f513a465000**

**mmap(0x7f513a46a000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1c000) = 0x7f513a46a000**

**mmap(0x7f513a46c000, 13432, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7f513a46c000**

**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) = 0x7f513a2fe000**

**mmap(0x7f513a30d000, 684032, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xf000) = 0x7f513a30d000**

**mmap(0x7f513a3b4000, 618496, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xb6000) = 0x7f513a3b4000**

**mmap(0x7f513a44b000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x14c000) = 0x7f513a44b000**

**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) = 0x7f513a220000**

**mprotect(0x7f513a242000, 700416, PROT\_NONE) = 0**

**mmap(0x7f513a242000, 397312, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x22000) = 0x7f513a242000**

**mmap(0x7f513a2a3000, 299008, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x83000) = 0x7f513a2a3000**

**mmap(0x7f513a2ed000, 65536, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xcc000) = 0x7f513a2ed000**

**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) = 0x7f513a210000**

**mmap(NULL, 196696, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f513a1d0000**

**mprotect(0x7f513a1d4000, 172032, PROT\_NONE) = 0**

**mmap(0x7f513a1d4000, 114688, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x4000) = 0x7f513a1d4000**

**mmap(0x7f513a1f0000, 53248, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x20000) = 0x7f513a1f0000**

**mmap(0x7f513a1fe000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x2d000) = 0x7f513a1fe000**

**mmap(0x7f513a200000, 88, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7f513a200000**

**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) = 0x7f513a1c0000**

**mmap(0x7f513a1c2000, 8192, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x2000) = 0x7f513a1c2000**

**mmap(0x7f513a1c4000, 4096, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x4000) = 0x7f513a1c4000**

**mmap(0x7f513a1c5000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x4000) = 0x7f513a1c5000**

**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) = 0x7f513a1b0000**

**mmap(0x7f513a1b3000, 28672, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x3000) = 0x7f513a1b3000**

**mmap(0x7f513a1ba000, 12288, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xa000) = 0x7f513a1ba000**

**mmap(0x7f513a1bd000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xc000) = 0x7f513a1bd000**

**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) = 0x7f513a1a0000**

**mmap(0x7f513a1a2000, 8192, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x2000) = 0x7f513a1a2000**

**mmap(0x7f513a1a4000, 4096, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x4000) = 0x7f513a1a4000**

**mmap(0x7f513a1a5000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x4000) = 0x7f513a1a5000**

**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) = 0x7f513a180000**

**mprotect(0x7f513a184000, 81920, PROT\_NONE) = 0**

**mmap(0x7f513a184000, 65536, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x4000) = 0x7f513a184000**

**mmap(0x7f513a194000, 12288, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x14000) = 0x7f513a194000**

**mmap(0x7f513a198000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x17000) = 0x7f513a198000**

**mmap(0x7f513a19a000, 6784, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7f513a19a000**

**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) = 0x7f513a170000**

**mmap(NULL, 20752, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f513a160000**

**mmap(0x7f513a161000, 8192, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1000) = 0x7f513a161000**

**mmap(0x7f513a163000, 4096, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x3000) = 0x7f513a163000**

**mmap(0x7f513a164000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x3000) = 0x7f513a164000**

**close(3) = 0**

**mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7f513a150000**

**mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7f513a140000**

**arch\_prctl(ARCH\_SET\_FS, 0x7f513a151600) = 0**

**mprotect(0x7f513a8c8000, 12288, PROT\_READ) = 0**

**mprotect(0x7f513a164000, 4096, PROT\_READ) = 0**

**mprotect(0x7f513a198000, 4096, PROT\_READ) = 0**

**mprotect(0x7f513a1a5000, 4096, PROT\_READ) = 0**

**mprotect(0x7f513a1bd000, 4096, PROT\_READ) = 0**

**mprotect(0x7f513a46a000, 4096, PROT\_READ) = 0**

**mprotect(0x7f513a1c5000, 4096, PROT\_READ) = 0**

**mprotect(0x7f513a1fe000, 4096, PROT\_READ) = 0**

**mprotect(0x7f513a2ed000, 57344, PROT\_READ) = 0**

**mprotect(0x7f513a44b000, 4096, PROT\_READ) = 0**

**mprotect(0x7f513a4b9000, 8192, PROT\_READ) = 0**

**mprotect(0x7f513a8f9000, 4096, PROT\_READ) = 0**

**mprotect(0x7f513aad0000, 45056, PROT\_READ) = 0**

**mprotect(0x7f513a566000, 12288, PROT\_READ) = 0**

**mprotect(0x7f513a679000, 4096, PROT\_READ) = 0**

**mprotect(0x7f513a6d6000, 4096, PROT\_READ) = 0**

**mprotect(0x7f513ab8e000, 28672, PROT\_READ) = 0**

**mprotect(0x7f513abe5000, 4096, PROT\_READ) = 0**

**mprotect(0x7f513abcd000, 4096, PROT\_READ) = 0**

**munmap(0x7f513ab98000, 32310) = 0**

**set\_tid\_address(0x7f513a1518d0) = 4153**

**set\_robust\_list(0x7f513a1518e0, 24) = 0**

**rt\_sigaction(SIGRTMIN, {sa\_handler=0x7f513a454bf0, sa\_mask=[], sa\_flags=SA\_RESTORER|SA\_SIGINFO, sa\_restorer=0x7f513a4623c0}, NULL, 8) = 0**

**rt\_sigaction(SIGRT\_1, {sa\_handler=0x7f513a454c90, sa\_mask=[], sa\_flags=SA\_RESTORER|SA\_RESTART|SA\_SIGINFO, sa\_restorer=0x7f513a4623c0}, NULL, 8) = 0**

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

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

**brk(NULL) = 0x7fffe0bc4000**

**brk(0x7fffe0be5000) = 0x7fffe0be5000**

**gettimeofday({tv\_sec=1621064014, tv\_usec=102374}, {tz\_minuteswest=0, tz\_dsttime=0}) = 0**

**futex(0x7f513aade6bc, FUTEX\_WAKE\_PRIVATE, 2147483647) = 0**

**futex(0x7f513aade6c8, FUTEX\_WAKE\_PRIVATE, 2147483647) = 0**

**clock\_gettime(CLOCK\_REALTIME\_COARSE, {tv\_sec=1621064014, tv\_nsec=103668300}) = 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, /\* 9 entries \*/, 32768) = 240**

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

**close(3) = 0**

**getpid() = 4153**

**sched\_getaffinity(4153, 128, [0, 1, 2, 3]) = 64**

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

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

**read(3, "# /etc/nsswitch.conf\n#\n# Example"..., 512) = 510**

**read(3, "", 512) = 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=32310, ...}) = 0**

**mmap(NULL, 32310, PROT\_READ, MAP\_PRIVATE, 3, 0) = 0x7f513ab98000**

**close(3) = 0**

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

**stat("/lib/x86\_64-linux-gnu/tls/haswell/x86\_64", 0x7fffe8365ba0) = -1 ENOENT (No such file or directory)**

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

**stat("/lib/x86\_64-linux-gnu/tls/haswell", 0x7fffe8365ba0) = -1 ENOENT (No such file or directory)**

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

**stat("/lib/x86\_64-linux-gnu/tls/x86\_64", 0x7fffe8365ba0) = -1 ENOENT (No such file or directory)**

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

**stat("/lib/x86\_64-linux-gnu/tls", 0x7fffe8365ba0) = -1 ENOENT (No such file or directory)**

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

**stat("/lib/x86\_64-linux-gnu/haswell/x86\_64", 0x7fffe8365ba0) = -1 ENOENT (No such file or directory)**

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

**stat("/lib/x86\_64-linux-gnu/haswell", 0x7fffe8365ba0) = -1 ENOENT (No such file or directory)**

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

**stat("/lib/x86\_64-linux-gnu/x86\_64", 0x7fffe8365ba0) = -1 ENOENT (No such file or directory)**

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

**stat("/lib/x86\_64-linux-gnu", {st\_mode=S\_IFDIR|0755, st\_size=512, ...}) = 0**

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

**stat("/usr/lib/x86\_64-linux-gnu/tls/haswell/x86\_64", 0x7fffe8365ba0) = -1 ENOENT (No such file or directory)**

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

**stat("/usr/lib/x86\_64-linux-gnu/tls/haswell", 0x7fffe8365ba0) = -1 ENOENT (No such file or directory)**

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

**stat("/usr/lib/x86\_64-linux-gnu/tls/x86\_64", 0x7fffe8365ba0) = -1 ENOENT (No such file or directory)**

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

**stat("/usr/lib/x86\_64-linux-gnu/tls", 0x7fffe8365ba0) = -1 ENOENT (No such file or directory)**

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

**stat("/usr/lib/x86\_64-linux-gnu/haswell/x86\_64", 0x7fffe8365ba0) = -1 ENOENT (No such file or directory)**

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

**stat("/usr/lib/x86\_64-linux-gnu/haswell", 0x7fffe8365ba0) = -1 ENOENT (No such file or directory)**

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

**stat("/usr/lib/x86\_64-linux-gnu/x86\_64", 0x7fffe8365ba0) = -1 ENOENT (No such file or directory)**

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

**stat("/usr/lib/x86\_64-linux-gnu", {st\_mode=S\_IFDIR|0755, st\_size=512, ...}) = 0**

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

**stat("/lib/tls/haswell/x86\_64", 0x7fffe8365ba0) = -1 ENOENT (No such file or directory)**

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

**stat("/lib/tls/haswell", 0x7fffe8365ba0) = -1 ENOENT (No such file or directory)**

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

**stat("/lib/tls/x86\_64", 0x7fffe8365ba0) = -1 ENOENT (No such file or directory)**

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

**stat("/lib/tls", 0x7fffe8365ba0) = -1 ENOENT (No such file or directory)**

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

**stat("/lib/haswell/x86\_64", 0x7fffe8365ba0) = -1 ENOENT (No such file or directory)**

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

**stat("/lib/haswell", 0x7fffe8365ba0) = -1 ENOENT (No such file or directory)**

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

**stat("/lib/x86\_64", 0x7fffe8365ba0) = -1 ENOENT (No such file or directory)**

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

**stat("/lib", {st\_mode=S\_IFDIR|0755, st\_size=512, ...}) = 0**

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

**stat("/usr/lib/tls/haswell/x86\_64", 0x7fffe8365ba0) = -1 ENOENT (No such file or directory)**

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

**stat("/usr/lib/tls/haswell", 0x7fffe8365ba0) = -1 ENOENT (No such file or directory)**

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

**stat("/usr/lib/tls/x86\_64", 0x7fffe8365ba0) = -1 ENOENT (No such file or directory)**

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

**stat("/usr/lib/tls", 0x7fffe8365ba0) = -1 ENOENT (No such file or directory)**

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

**stat("/usr/lib/haswell/x86\_64", 0x7fffe8365ba0) = -1 ENOENT (No such file or directory)**

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

**stat("/usr/lib/haswell", 0x7fffe8365ba0) = -1 ENOENT (No such file or directory)**

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

**stat("/usr/lib/x86\_64", 0x7fffe8365ba0) = -1 ENOENT (No such file or directory)**

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

**stat("/usr/lib", {st\_mode=S\_IFDIR|0755, st\_size=512, ...}) = 0**

**munmap(0x7f513ab98000, 32310) = 0**

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

**fstat(3, {st\_mode=S\_IFREG|0644, st\_size=32310, ...}) = 0**

**mmap(NULL, 32310, PROT\_READ, MAP\_PRIVATE, 3, 0) = 0x7f513ab98000**

**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) = 0x7f513a120000**

**mmap(0x7f513a123000, 28672, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x3000) = 0x7f513a123000**

**mmap(0x7f513a12a000, 8192, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xa000) = 0x7f513a12a000**

**mmap(0x7f513a12c000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xb000) = 0x7f513a12c000**

**mmap(0x7f513a12e000, 22328, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7f513a12e000**

**close(3) = 0**

**mprotect(0x7f513a12c000, 4096, PROT\_READ) = 0**

**munmap(0x7f513ab98000, 32310) = 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"..., 512) = 512**

**lseek(3, 0, SEEK\_CUR) = 512**

**read(3, "teway-gateway protocol\nipencap\t4"..., 512) = 512**

**lseek(3, 0, SEEK\_CUR) = 1024**

**read(3, "33\tDCCP\t\t# Datagram Congestion C"..., 512) = 512**

**lseek(3, 0, SEEK\_CUR) = 1536**

**read(3, "06]\nah\t51\tIPSEC-AH\t# Authenticat"..., 512) = 512**

**lseek(3, 0, SEEK\_CUR) = 2048**

**read(3, "tocol\netherip\t97\tETHERIP\t\t# Ethe"..., 512) = 512**

**lseek(3, 0, SEEK\_CUR) = 2560**

**read(3, "r 135 Mobility-Header # Mobility"..., 512) = 372**

**lseek(3, 0, SEEK\_CUR) = 2932**

**read(3, "", 512) = 0**

**close(3) = 0**

**gettimeofday({tv\_sec=1621064014, tv\_usec=128448}, NULL) = 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("\x1c\xab\x2e\x1d\x87\x30\x3e\xb9\x37\x02\xb5\xe6\x23\xf8\xdb\xe6", 16, 0) = 16**

**getrandom("\x4e\x83\xcd\xc8\x5f\x93\x02\x6c\x82\x35\x0b\x20\x3e\x27\x3d\x43", 16, 0) = 16**

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

**read(0, "create 2 -1\n", 1024) = 12**

**clone(child\_stack=NULL, flags=CLONE\_CHILD\_CLEARTID|CLONE\_CHILD\_SETTID|SIGCHLD, child\_tidptr=0x7f513a1518d0) = 4154**

**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**

**clock\_gettime(CLOCK\_MONOTONIC, {tv\_sec=17988, tv\_nsec=305114500}) = 0**

**epoll\_create1(EPOLL\_CLOEXEC) = 5**

**epoll\_ctl(5, EPOLL\_CTL\_ADD, 4, {0, {u32=3770529072, u64=140736963917104}}) = 0**

**epoll\_ctl(5, EPOLL\_CTL\_MOD, 4, {EPOLLIN, {u32=3770529072, u64=140736963917104}}) = 0**

**mmap(NULL, 8392704, PROT\_NONE, MAP\_PRIVATE|MAP\_ANONYMOUS|MAP\_STACK, -1, 0) = 0x7f5139910000**

**mprotect(0x7f5139911000, 8388608, PROT\_READ|PROT\_WRITE) = 0**

**clone(child\_stack=0x7f513a10fd30, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, parent\_tid=[4155], tls=0x7f513a110700, child\_tidptr=0x7f513a1109d0) = 4155**

**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**

**clock\_gettime(CLOCK\_MONOTONIC, {tv\_sec=17988, tv\_nsec=307361000}) = 0**

**epoll\_create1(EPOLL\_CLOEXEC) = 7**

**epoll\_ctl(7, EPOLL\_CTL\_ADD, 6, {0, {u32=3770531120, u64=140736963919152}}) = 0**

**epoll\_ctl(7, EPOLL\_CTL\_MOD, 6, {EPOLLIN, {u32=3770531120, u64=140736963919152}}) = 0**

**mmap(NULL, 8392704, PROT\_NONE, MAP\_PRIVATE|MAP\_ANONYMOUS|MAP\_STACK, -1, 0) = 0x7f5139100000**

**mprotect(0x7f5139101000, 8388608, PROT\_READ|PROT\_WRITE) = 0**

**clone(child\_stack=0x7f51398ffd30, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, parent\_tid=[4156], tls=0x7f5139900700, child\_tidptr=0x7f51399009d0) = 4156**

**clock\_gettime(CLOCK\_MONOTONIC, {tv\_sec=17988, tv\_nsec=308280700}) = 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**

**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=4153, nl\_groups=00000000}, [12]) = 0**

**clock\_gettime(CLOCK\_REALTIME\_COARSE, {tv\_sec=1621064019, tv\_nsec=835082200}) = 0**

**sendto(9, "\24\0\0\0\22\0\1\3Sy\237`\0\0\0\0\0\0\0\0", 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=80, type=0x10 /\* NLMSG\_??? \*/, flags=NLM\_F\_MULTI, seq=1621064019, pid=4153}, "\x00\x00\x01\x00\x04\x00\x00\x00\x43\x10\x00\x00\x00\x00\x00\x00\x0a\x00\x01\x00\xfc\xaa\x14\x20\x7f\x3d\x00\x00\x08\x00\x1b\x00"...}, iov\_len=4096}], msg\_iovlen=1, msg\_controllen=0, msg\_flags=0}, 0) = 80**

**recvmsg(9, {msg\_name={sa\_family=AF\_NETLINK, nl\_pid=0, nl\_groups=00000000}, msg\_namelen=12, msg\_iov=[{iov\_base={{len=80, type=0x10 /\* NLMSG\_??? \*/, flags=NLM\_F\_MULTI, seq=1621064019, pid=4153}, "\x00\x00\x01\x00\x07\x00\x00\x00\x43\x10\x00\x00\x00\x00\x00\x00\x0a\x00\x01\x00\x00\x50\x56\xc0\x00\x01\x00\x00\x08\x00\x1b\x00"...}, iov\_len=4096}], msg\_iovlen=1, msg\_controllen=0, msg\_flags=0}, 0) = 80**

**recvmsg(9, {msg\_name={sa\_family=AF\_NETLINK, nl\_pid=0, nl\_groups=00000000}, msg\_namelen=12, msg\_iov=[{iov\_base={{len=80, type=0x10 /\* NLMSG\_??? \*/, flags=NLM\_F\_MULTI, seq=1621064019, pid=4153}, "\x00\x00\x01\x00\x05\x00\x00\x00\x43\x10\x00\x00\x00\x00\x00\x00\x0a\x00\x01\x00\x00\x50\x56\xc0\x00\x08\x00\x00\x08\x00\x1b\x00"...}, iov\_len=4096}], msg\_iovlen=1, msg\_controllen=0, msg\_flags=0}, 0) = 80**

**recvmsg(9, {msg\_name={sa\_family=AF\_NETLINK, nl\_pid=0, nl\_groups=00000000}, msg\_namelen=12, msg\_iov=[{iov\_base={{len=76, type=0x10 /\* NLMSG\_??? \*/, flags=NLM\_F\_MULTI, seq=1621064019, pid=4153}, "\x00\x00\x04\x03\x01\x00\x00\x00\x49\x00\x00\x00\x00\x00\x00\x00\x0a\x00\x01\x00\x00\x00\x00\x00\x00\x00\x00\x00\x08\x00\x1b\x00"...}, iov\_len=4096}], msg\_iovlen=1, msg\_controllen=0, msg\_flags=0}, 0) = 76**

**recvmsg(9, {msg\_name={sa\_family=AF\_NETLINK, nl\_pid=0, nl\_groups=00000000}, msg\_namelen=12, msg\_iov=[{iov\_base={{len=80, type=0x10 /\* NLMSG\_??? \*/, flags=NLM\_F\_MULTI, seq=1621064019, pid=4153}, "\x00\x00\x01\x00\x08\x00\x00\x00\x40\x00\x00\x00\x00\x00\x00\x00\x0a\x00\x01\x00\x00\xff\x78\x0b\xbb\xc3\x00\x00\x08\x00\x1b\x00"...}, iov\_len=4096}], msg\_iovlen=1, msg\_controllen=0, msg\_flags=0}, 0) = 80**

**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=1621064019, pid=4153}, 0}, iov\_len=4096}], msg\_iovlen=1, msg\_controllen=0, msg\_flags=0}, 0) = 20**

**sendto(9, "\24\0\0\0\26\0\1\3Ty\237`\0\0\0\0\0\0\0\0", 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=60, type=0x14 /\* NLMSG\_??? \*/, flags=NLM\_F\_MULTI, seq=1621064020, pid=4153}, "\x02\x18\x00\x00\x04\x00\x00\x00\x08\x00\x01\x00\xc0\xa8\x00\x0c\x08\x00\x04\x00\xc0\xa8\x00\xff\x14\x00\x06\x00\xc3\x49\x01\x00"...}, iov\_len=4096}], msg\_iovlen=1, msg\_controllen=0, msg\_flags=0}, 0) = 60**

**recvmsg(9, {msg\_name={sa\_family=AF\_NETLINK, nl\_pid=0, nl\_groups=00000000}, msg\_namelen=12, msg\_iov=[{iov\_base={{len=64, type=0x14 /\* NLMSG\_??? \*/, flags=NLM\_F\_MULTI, seq=1621064020, pid=4153}, "\x0a\x40\x00\xfd\x04\x00\x00\x00\x14\x00\x01\x00\xfe\x80\x00\x00\x00\x00\x00\x00\xfc\x6d\x29\x99\x2d\xf4\x30\xa3\x14\x00\x06\x00"...}, iov\_len=4096}], msg\_iovlen=1, msg\_controllen=0, msg\_flags=0}, 0) = 64**

**recvmsg(9, {msg\_name={sa\_family=AF\_NETLINK, nl\_pid=0, nl\_groups=00000000}, msg\_namelen=12, msg\_iov=[{iov\_base={{len=60, type=0x14 /\* NLMSG\_??? \*/, flags=NLM\_F\_MULTI, seq=1621064020, pid=4153}, "\x02\x18\x00\x00\x07\x00\x00\x00\x08\x00\x01\x00\xc0\xa8\xe4\x01\x08\x00\x04\x00\xc0\xa8\xe4\xff\x14\x00\x06\x00\x50\x06\x00\x00"...}, iov\_len=4096}], msg\_iovlen=1, msg\_controllen=0, msg\_flags=0}, 0) = 60**

**recvmsg(9, {msg\_name={sa\_family=AF\_NETLINK, nl\_pid=0, nl\_groups=00000000}, msg\_namelen=12, msg\_iov=[{iov\_base={{len=64, type=0x14 /\* NLMSG\_??? \*/, flags=NLM\_F\_MULTI, seq=1621064020, pid=4153}, "\x0a\x40\x00\xfd\x07\x00\x00\x00\x14\x00\x01\x00\xfe\x80\x00\x00\x00\x00\x00\x00\x49\xd9\x30\x1a\xd0\x91\xe0\x04\x14\x00\x06\x00"...}, iov\_len=4096}], msg\_iovlen=1, msg\_controllen=0, msg\_flags=0}, 0) = 64**

**recvmsg(9, {msg\_name={sa\_family=AF\_NETLINK, nl\_pid=0, nl\_groups=00000000}, msg\_namelen=12, msg\_iov=[{iov\_base={{len=60, type=0x14 /\* NLMSG\_??? \*/, flags=NLM\_F\_MULTI, seq=1621064020, pid=4153}, "\x02\x18\x00\x00\x05\x00\x00\x00\x08\x00\x01\x00\xc0\xa8\x91\x01\x08\x00\x04\x00\xc0\xa8\x91\xff\x14\x00\x06\x00\x50\x06\x00\x00"...}, iov\_len=4096}], msg\_iovlen=1, msg\_controllen=0, msg\_flags=0}, 0) = 60**

**recvmsg(9, {msg\_name={sa\_family=AF\_NETLINK, nl\_pid=0, nl\_groups=00000000}, msg\_namelen=12, msg\_iov=[{iov\_base={{len=64, type=0x14 /\* NLMSG\_??? \*/, flags=NLM\_F\_MULTI, seq=1621064020, pid=4153}, "\x0a\x40\x00\xfd\x05\x00\x00\x00\x14\x00\x01\x00\xfe\x80\x00\x00\x00\x00\x00\x00\x1d\xc1\xa7\xe6\xc6\x51\xa7\x79\x14\x00\x06\x00"...}, iov\_len=4096}], msg\_iovlen=1, msg\_controllen=0, msg\_flags=0}, 0) = 64**

**recvmsg(9, {msg\_name={sa\_family=AF\_NETLINK, nl\_pid=0, nl\_groups=00000000}, msg\_namelen=12, msg\_iov=[{iov\_base={{len=60, type=0x14 /\* NLMSG\_??? \*/, flags=NLM\_F\_MULTI, seq=1621064020, pid=4153}, "\x02\x08\x00\x00\x01\x00\x00\x00\x08\x00\x01\x00\x7f\x00\x00\x01\x08\x00\x04\x00\x7f\xff\xff\xff\x14\x00\x06\x00\xff\xff\xff\xff"...}, iov\_len=4096}], msg\_iovlen=1, msg\_controllen=0, msg\_flags=0}, 0) = 60**

**recvmsg(9, {msg\_name={sa\_family=AF\_NETLINK, nl\_pid=0, nl\_groups=00000000}, msg\_namelen=12, msg\_iov=[{iov\_base={{len=64, type=0x14 /\* NLMSG\_??? \*/, flags=NLM\_F\_MULTI, seq=1621064020, pid=4153}, "\x0a\x80\x00\xfe\x01\x00\x00\x00\x14\x00\x01\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x01\x14\x00\x06\x00"...}, iov\_len=4096}], msg\_iovlen=1, msg\_controllen=0, msg\_flags=0}, 0) = 64**

**recvmsg(9, {msg\_name={sa\_family=AF\_NETLINK, nl\_pid=0, nl\_groups=00000000}, msg\_namelen=12, msg\_iov=[{iov\_base={{len=60, type=0x14 /\* NLMSG\_??? \*/, flags=NLM\_F\_MULTI, seq=1621064020, pid=4153}, "\x02\x10\x00\x00\x08\x00\x00\x00\x08\x00\x01\x00\xa9\xfe\x9b\x77\x08\x00\x04\x00\xa9\xfe\xff\xff\x14\x00\x06\x00\xff\xff\xff\xff"...}, iov\_len=4096}], msg\_iovlen=1, msg\_controllen=0, msg\_flags=0}, 0) = 60**

**recvmsg(9, {msg\_name={sa\_family=AF\_NETLINK, nl\_pid=0, nl\_groups=00000000}, msg\_namelen=12, msg\_iov=[{iov\_base={{len=64, type=0x14 /\* NLMSG\_??? \*/, flags=NLM\_F\_MULTI, seq=1621064020, pid=4153}, "\x0a\x40\x00\xfd\x08\x00\x00\x00\x14\x00\x01\x00\xfe\x80\x00\x00\x00\x00\x00\x00\x05\xcb\x5b\xc9\x1a\x2c\x9b\x77\x14\x00\x06\x00"...}, iov\_len=4096}], msg\_iovlen=1, msg\_controllen=0, msg\_flags=0}, 0) = 64**

**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=1621064020, pid=4153}, 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(4042), sin\_addr=inet\_addr("127.0.0.1")}, 16) = 0**

**listen(9, 100) = 0**

**getsockname(9, {sa\_family=AF\_INET, sin\_port=htons(4042), sin\_addr=inet\_addr("127.0.0.1")}, [128->16]) = 0**

**getsockname(9, {sa\_family=AF\_INET, sin\_port=htons(4042), 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**

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

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

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

**clock\_gettime(CLOCK\_MONOTONIC, {tv\_sec=17988, tv\_nsec=315827700}) = 0**

**poll([{fd=8, events=POLLIN}], 1, 5000) = 1 ([{fd=8, revents=POLLIN}])**

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

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

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

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

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

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

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

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

**write(1, "OK: 4154\n", 9) = 9**

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

**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**

**--- SIGCHLD {si\_signo=SIGCHLD, si\_code=CLD\_EXITED, si\_pid=4154, si\_uid=1000, si\_status=0, si\_utime=0, si\_stime=0} ---**

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

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

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

**lseek(0, -1, SEEK\_CUR) = -1 ESPIPE (Illegal seek)**

**exit\_group(0) = ?**

**+++ exited with 0 +++**

1. **Выводы**

Данная лабораторная работа была направлена на изучении технологии очереди сообщений, на основе которой необходимо было построить сеть с заданной топологией.

Наряду с каналами и отображаемыми файлами, очереди сообщений являются достаточно удобным способом для взаимодействия между процессами. ZeroMQ предоставляет достаточно простой интерфейс для передачи сообщений, а также поддерживает все возможные типы соединений.

Полученные мной навыки работы с очередями сообщений можно использовать при проектировании мессенджеров, многопользовательских игр, и вообще для любых мультипроцессорных программ.