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.

2. Общие сведения о программе

Программа написана на языке C++ 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.

3. 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.

4. 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");
        }
    }
}
```

5. 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 +++
```

6. 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.