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: Моисеенков Илья Павлович

Группа: М80 – 208Б-19

Вариант: 4

Преподаватель: Миронов Евгений Сергеевич

Дата: 11.01.2021

Оценка: хорошо

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.
* kill [id] – удалить узел [id]. Все дочерние узлы будут также удалены.
* exec [id] add [key] [value] – добавить переменную [key] со значением [value] в словарь узла [id]
* exec [id] check [key] – запросить значение переменной [key] в словаре узла [id].
* pingall – проверить доступность узлов. Будет выведен список всех доступных 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 хранения локального словаря используется контейнер std::unordered\_map. Для проверки доступности узлов используется контейнер std::set. Управляющий узел отправляет запрос всем спискам узлов и получает в ответ строку с id всех доступных узлов списка. Все id добавляются в set, а потом выводятся на экран.

1. **Основные файлы программы**

***topology.h***

#include <list>

#include <stdexcept>

class topology {

private:

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

public:

    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;

                }

            }

        }

    }

    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;

    }

    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);

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

            if (\*it2 == id) {

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

                if (it1->empty()) {

                    container.erase(it1);

                }

                return;

            }

        }

    }

    int get\_first\_id(int list\_id) {

        auto it1 = container.begin();

        std::advance(it1, list\_id);

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

            return -1;

        }

        return \*(it1->begin());

    }

};

***zmq\_functions.h***

#include <zmq.hpp>

#include <iostream>

const int MAIN\_PORT = 4040;

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

*//std::cout << "Start sending " << msg << std::endl;*

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

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

    socket.send(message);

*//std::cout << "Finished sending " << msg << std::endl;*

}

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

*//std::cout << "Start receiving" << std::endl;*

    zmq::message\_t message;

    int chars\_read;

    try {

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

    }

    catch (...) {

        chars\_read = 0;

    }

    if (chars\_read == 0) {

        return "Error: node is unavailable [zmq\_func]";

    }

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

*//std::cout << "Received " << received\_msg << std::endl;*

    return received\_msg;

}

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

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

    socket.connect(address);

*//std::cout << "Socket connected to " << address << std::endl;*

}

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

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

    socket.disconnect(address);

*//std::cout << "Socket disconnected from " << address << std::endl;*

}

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

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

    socket.bind(address);

*//std::cout << "Socket binded to " << address << std::endl;*

}

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

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

    socket.unbind(address);

*//std::cout << "Socket unbinded from " << address << std::endl;*

}

***control.cpp***

#include <unistd.h>

#include <sstream>

#include <set>

#include "zmq\_functions.h"

#include "topology.h"

int main() {

    topology network;

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

    zmq::context\_t context;

    std::string cmd;

    while (std::cin >> cmd) {

        if (cmd == "create") {

            int node\_id, parent\_id;

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

            if (network.find(node\_id) != -1) {

                std::cout << "Error: already exists" << std::endl;

            }

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

                pid\_t pid = fork();

                if (pid < 0) {

                    perror("Can't create new process");

                    return -1;

                }

                if (pid == 0) {

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

                    perror("Can't execute new process");

                    return -2;

                }

                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 << std::endl;

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

            }

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

                std::cout << "Error: parent not found" << std::endl;

            }

            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 << std::endl;

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

            }

        }

        else if (cmd == "exec") {

            int dest\_id;

            std::string exec\_cmd, key;

            std::cin >> dest\_id >> exec\_cmd >> key;

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

            if (branch == -1) {

                std::cout << "ERROR: incorrect node id" << std::endl;

            }

            else {

                if (exec\_cmd == "check") {

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

                }

                else if (exec\_cmd == "add") {

                    int value;

                    std::cin >> value;

                    send\_message(branches[branch], std::to\_string(dest\_id) + "add " + key + " " + std::to\_string(value));

                }

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

                std::cout << reply << std::endl;

            }

        }

        else if (cmd == "kill") {

            int id;

            std::cin >> id;

            int branch = network.find(id);

            if (branch == -1) {

                std::cout << "ERROR: incorrect node id" << std::endl;

            }

            else {

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

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

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

                std::cout << reply << std::endl;

                network.erase(id);

                if (is\_first) {

                    unbind(branches[branch], id);

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

                }

            }

        }

        else if (cmd == "pingall") {

            std::set<int> available\_nodes;

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

                int first\_node\_id = network.get\_first\_id(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) {

                    available\_nodes.insert(node);

                }

            }

            std::cout << "OK: ";

            if (available\_nodes.empty()) {

                std::cout << "no available nodes" << std::endl;

            }

            else {

                for (auto v : available\_nodes) {

                    std::cout << v << " ";

                }

                std::cout << std::endl;

            }

        }

        else if (cmd == "exit") {

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

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

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

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

                if (reply != "OK") {

                    std::cout << reply << std::endl;

                }

                else {

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

                }

            }

            exit(0);

        }

        else {

            std::cout << "Incorrect cmd" << std::endl;

        }

    }

}

***counting.cpp***

#include <unordered\_map>

#include <unistd.h>

#include <sstream>

#include <unordered\_map>

#include "zmq\_functions.h"

int main(int argc, char\* argv[]) {

    if (argc != 2 && argc != 3) {

        throw std::runtime\_error("Wrong args for counting node");

    }

    int cur\_id = std::atoi(argv[1]);

    int child\_id = -1;

    if (argc == 3) {

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

    }

    std::unordered\_map<std::string, int> dictionary;

    zmq::context\_t context;

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

    connect(parent\_socket, cur\_id);

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

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

    if (child\_id != -1) {

        bind(child\_socket, child\_id);

    }

    std::string message;

    while (true) {

        message = receive\_message(parent\_socket);

        std::istringstream request(message);

        int dest\_id;

        request >> dest\_id;

        std::string cmd;

        request >> cmd;

        if (dest\_id == cur\_id) {

            if (cmd == "pid") {

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

            }

            else if (cmd == "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");

                    return -1;

                }

                if (pid == 0) {

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

                    perror("Can't execute new process");

                    return -2;

                }

                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 (cmd == "check") {

                std::string key;

                request >> key;

                if (dictionary.find(key) != dictionary.end()) {

                    send\_message(parent\_socket, "OK: " + std::to\_string(cur\_id) + ": " + std::to\_string(dictionary[key]));

                }

                else {

                    send\_message(parent\_socket, "OK: " + std::to\_string(cur\_id) + ": '" + key + "' not found");

                }

            }

            else if (cmd == "add") {

                std::string key;

                int value;

                request >> key >> value;

                dictionary[key] = value;

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

            }

            else if (cmd == "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(cur\_id) + reply);

            }

            else if (cmd == "kill") {

                if (child\_id != -1) {

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

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

                    if (msg == "OK") {

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

                    }

                    unbind(child\_socket, child\_id);

                    disconnect(parent\_socket, cur\_id);

                    break;

                }

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

                disconnect(parent\_socket, cur\_id);

                break;

            }

        }

        else if (child\_id != -1) {

            send\_message(child\_socket, message);

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

            if (child\_id == dest\_id && cmd == "kill") {

                child\_id = -1;

            }

        }

        else {

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

        }

    }

}

1. **Демонстрация работы программы**

**mosik@LAPTOP-69S778GL:~/os\_lab6$** ./control

create 1 -1

OK: 17801

create 100 -1

OK: 17818

create 3 1

OK: 17833

create 4 3

OK: 17842

create 5 4

OK: 17857

create 200 100

OK: 17866

pingall

OK: 1 3 4 5 100 200

exec 3 add var1 10

OK: 3

exec 3 check var1

OK: 3: 10

exec 3 check var2

OK: 3: 'var2' not found

exec 4 check var1

OK: 4: 'var1' not found

create 2 1

OK: 17947

pingall

OK: 1 2 3 4 5 100 200

exec 3 check var1

OK: 3: 10

kill 2

OK

pingall

OK: 1 100 200

kill -1

ERROR: incorrect node id

kill 200

OK

pingall

OK: 1 100

exec 3 check var1

ERROR: incorrect node id

exit

**mosik@LAPTOP-69S778GL:~/os\_lab6$** strace -o strace\_log.txt ./control

create 0 -1

OK: 20242

exit

**mosik@LAPTOP-69S778GL:~/os\_lab6$** cat strace\_log.txt

execve("./control", ["./control"], 0x7fffd7c8fb10 /\* 19 vars \*/) = 0

brk(NULL) = 0x7fffdb4a2000

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

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=43552, ...}) = 0

mmap(NULL, 43552, PROT\_READ, MAP\_PRIVATE, 3, 0) = 0x7f71eb6c4000

close(3) = 0

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

openat(AT\_FDCWD, "/usr/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\0P?\1\0\0\0\0\0"..., 832) = 832

fstat(3, {st\_mode=S\_IFREG|0644, st\_size=630464, ...}) = 0

mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7f71eb6c0000

mmap(NULL, 2725560, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f71eb160000

mprotect(0x7f71eb1f3000, 2097152, PROT\_NONE) = 0

mmap(0x7f71eb3f3000, 28672, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x93000) = 0x7f71eb3f3000

close(3) = 0

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

openat(AT\_FDCWD, "/usr/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\220\304\10\0\0\0\0\0"..., 832) = 832

fstat(3, {st\_mode=S\_IFREG|0644, st\_size=1594864, ...}) = 0

mmap(NULL, 3702848, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f71eadd0000

mprotect(0x7f71eaf49000, 2097152, PROT\_NONE) = 0

mmap(0x7f71eb149000, 49152, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x179000) = 0x7f71eb149000

mmap(0x7f71eb155000, 12352, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7f71eb155000

close(3) = 0

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

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\300\*\0\0\0\0\0\0"..., 832) = 832

fstat(3, {st\_mode=S\_IFREG|0644, st\_size=96616, ...}) = 0

mmap(NULL, 2192432, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f71eabb0000

mprotect(0x7f71eabc7000, 2093056, PROT\_NONE) = 0

mmap(0x7f71eadc6000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x16000) = 0x7f71eadc6000

close(3) = 0

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

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\20\35\2\0\0\0\0\0"..., 832) = 832

fstat(3, {st\_mode=S\_IFREG|0755, st\_size=2030928, ...}) = 0

mmap(NULL, 4131552, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f71ea7b0000

mprotect(0x7f71ea997000, 2097152, PROT\_NONE) = 0

mmap(0x7f71eab97000, 24576, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1e7000) = 0x7f71eab97000

mmap(0x7f71eab9d000, 15072, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7f71eab9d000

close(3) = 0

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

openat(AT\_FDCWD, "/usr/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\340\251\0\0\0\0\0\0"..., 832) = 832

fstat(3, {st\_mode=S\_IFREG|0644, st\_size=330440, ...}) = 0

mmap(NULL, 2425864, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f71ea550000

mprotect(0x7f71ea5a0000, 2093056, PROT\_NONE) = 0

mmap(0x7f71ea79f000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x4f000) = 0x7f71ea79f000

close(3) = 0

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

openat(AT\_FDCWD, "/usr/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\0000;\0\0\0\0\0\0"..., 832) = 832

fstat(3, {st\_mode=S\_IFREG|0644, st\_size=293784, ...}) = 0

mmap(NULL, 2406448, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f71ea300000

mprotect(0x7f71ea347000, 2093056, PROT\_NONE) = 0

mmap(0x7f71ea546000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x46000) = 0x7f71ea546000

mmap(0x7f71ea548000, 14384, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7f71ea548000

close(3) = 0

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

openat(AT\_FDCWD, "/usr/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\0000\374\1\0\0\0\0\0"..., 832) = 832

fstat(3, {st\_mode=S\_IFREG|0644, st\_size=522248, ...}) = 0

mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7f71eb6b0000

mmap(NULL, 3340624, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f71e9fd0000

mprotect(0x7f71ea04d000, 2097152, PROT\_NONE) = 0

mmap(0x7f71ea24d000, 12288, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x7d000) = 0x7f71ea24d000

mmap(0x7f71ea250000, 719184, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7f71ea250000

close(3) = 0

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

openat(AT\_FDCWD, "/lib/x86\_64-linux-gnu/librt.so.1", O\_RDONLY|O\_CLOEXEC) = 3

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

fstat(3, {st\_mode=S\_IFREG|0644, st\_size=31680, ...}) = 0

mmap(NULL, 2128864, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f71e9dc0000

mprotect(0x7f71e9dc7000, 2093056, PROT\_NONE) = 0

mmap(0x7f71e9fc6000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x6000) = 0x7f71e9fc6000

close(3) = 0

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

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\0000b\0\0\0\0\0\0"..., 832) = 832

fstat(3, {st\_mode=S\_IFREG|0755, st\_size=144976, ...}) = 0

mmap(NULL, 2221184, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f71e9ba0000

mprotect(0x7f71e9bba000, 2093056, PROT\_NONE) = 0

mmap(0x7f71e9db9000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x19000) = 0x7f71e9db9000

mmap(0x7f71e9dbb000, 13440, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7f71e9dbb000

close(3) = 0

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

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\200\272\0\0\0\0\0\0"..., 832) = 832

fstat(3, {st\_mode=S\_IFREG|0644, st\_size=1700792, ...}) = 0

mmap(NULL, 3789144, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f71e9800000

mprotect(0x7f71e999d000, 2093056, PROT\_NONE) = 0

mmap(0x7f71e9b9c000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x19c000) = 0x7f71e9b9c000

close(3) = 0

mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7f71eb6a0000

mmap(NULL, 12288, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7f71eb690000

arch\_prctl(ARCH\_SET\_FS, 0x7f71eb690b80) = 0

mprotect(0x7f71eab97000, 16384, PROT\_READ) = 0

mprotect(0x7f71e9b9c000, 4096, PROT\_READ) = 0

mprotect(0x7f71e9db9000, 4096, PROT\_READ) = 0

mprotect(0x7f71e9fc6000, 4096, PROT\_READ) = 0

mprotect(0x7f71eadc6000, 4096, PROT\_READ) = 0

mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7f71eb680000

mprotect(0x7f71eb149000, 40960, PROT\_READ) = 0

mprotect(0x7f71ea24d000, 8192, PROT\_READ) = 0

mprotect(0x7f71ea546000, 4096, PROT\_READ) = 0

mprotect(0x7f71ea79f000, 4096, PROT\_READ) = 0

mprotect(0x7f71eb3f3000, 24576, PROT\_READ) = 0

mprotect(0x7f71eba0d000, 4096, PROT\_READ) = 0

mprotect(0x7f71eb629000, 4096, PROT\_READ) = 0

munmap(0x7f71eb6c4000, 43552) = 0

set\_tid\_address(0x7f71eb690e50) = 20241

set\_robust\_list(0x7f71eb690e60, 24) = 0

rt\_sigaction(SIGRTMIN, {sa\_handler=0x7f71e9ba5cb0, sa\_mask=[], sa\_flags=SA\_RESTORER|SA\_SIGINFO, sa\_restorer=0x7f71e9bb2980}, NULL, 8) = 0

rt\_sigaction(SIGRT\_1, {sa\_handler=0x7f71e9ba5d50, sa\_mask=[], sa\_flags=SA\_RESTORER|SA\_RESTART|SA\_SIGINFO, sa\_restorer=0x7f71e9bb2980}, 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) = 0x7fffdb4a2000

brk(0x7fffdb4c3000) = 0x7fffdb4c3000

gettimeofday({tv\_sec=1609673745, tv\_usec=387429}, {tz\_minuteswest=0, tz\_dsttime=0}) = 0

futex(0x7f71eb15609c, FUTEX\_WAKE\_PRIVATE, 2147483647) = 0

futex(0x7f71eb1560a8, FUTEX\_WAKE\_PRIVATE, 2147483647) = 0

time(NULL) = 1609673745 (2021-01-03T14:35:45+0300)

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

read(3, "0-7\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

getdents(3, /\* 13 entries \*/, 32768) = 336

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

close(3) = 0

getpid() = 20241

sched\_getaffinity(20241, 128, [0, 1, 2, 3, 4, 5, 6, 7]) = 64

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

fstat(3, {st\_mode=S\_IFREG|0644, st\_size=513, ...}) = 0

read(3, "# /etc/nsswitch.conf\n#\n# Example"..., 4096) = 513

read(3, "", 4096) = 0

close(3) = 0

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

fstat(3, {st\_mode=S\_IFREG|0644, st\_size=43552, ...}) = 0

mmap(NULL, 43552, PROT\_READ, MAP\_PRIVATE, 3, 0) = 0x7f71eb6c4000

close(3) = 0

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

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", 0x7fffe3679a70) = -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", 0x7fffe3679a70) = -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", 0x7fffe3679a70) = -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", 0x7fffe3679a70) = -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", 0x7fffe3679a70) = -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", 0x7fffe3679a70) = -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", 0x7fffe3679a70) = -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=4096, ...}) = 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", 0x7fffe3679a70) = -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", 0x7fffe3679a70) = -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", 0x7fffe3679a70) = -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", 0x7fffe3679a70) = -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", 0x7fffe3679a70) = -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", 0x7fffe3679a70) = -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", 0x7fffe3679a70) = -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=4096, ...}) = 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", 0x7fffe3679a70) = -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", 0x7fffe3679a70) = -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", 0x7fffe3679a70) = -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", 0x7fffe3679a70) = -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", 0x7fffe3679a70) = -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", 0x7fffe3679a70) = -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", 0x7fffe3679a70) = -1 ENOENT (No such file or directory)

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

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

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

stat("/usr/lib/tls/haswell/x86\_64", 0x7fffe3679a70) = -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", 0x7fffe3679a70) = -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", 0x7fffe3679a70) = -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", 0x7fffe3679a70) = -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", 0x7fffe3679a70) = -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", 0x7fffe3679a70) = -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", 0x7fffe3679a70) = -1 ENOENT (No such file or directory)

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

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

munmap(0x7f71eb6c4000, 43552) = 0

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

fstat(3, {st\_mode=S\_IFREG|0644, st\_size=43552, ...}) = 0

mmap(NULL, 43552, PROT\_READ, MAP\_PRIVATE, 3, 0) = 0x7f71eb6c4000

close(3) = 0

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

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

fstat(3, {st\_mode=S\_IFREG|0644, st\_size=47568, ...}) = 0

mmap(NULL, 2168632, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f71e95e0000

mprotect(0x7f71e95eb000, 2093056, PROT\_NONE) = 0

mmap(0x7f71e97ea000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xa000) = 0x7f71e97ea000

mmap(0x7f71e97ec000, 22328, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7f71e97ec000

close(3) = 0

mprotect(0x7f71e97ea000, 4096, PROT\_READ) = 0

munmap(0x7f71eb6c4000, 43552) = 0

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

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

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

read(3, "", 4096) = 0

close(3) = 0

gettimeofday({tv\_sec=1609673745, tv\_usec=399158}, 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("\x2a\x1c\xa3\x42\xdf\x17\x02\x09\x03\x45\x54\x73\xed\x15\xde\xac", 16, 0) = 16

getrandom("\xf8\x48\x09\xa9\xd0\x1e\x10\xe7\xbb\xac\x48\xcb\xb5\xdb\x18\x37", 16, 0) = 16

fstat(0, {st\_mode=S\_IFCHR|0660, st\_rdev=makedev(4, 1), ...}) = 0

ioctl(0, TCGETS, {B38400 opost isig icanon echo ...}) = 0

read(0, "create 0 -1\n", 4096) = 12

clone(child\_stack=NULL, flags=CLONE\_CHILD\_CLEARTID|CLONE\_CHILD\_SETTID|SIGCHLD, child\_tidptr=0x7f71eb690e50) = 20242

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=10895, tv\_nsec=66665100}) = 0

epoll\_create1(EPOLL\_CLOEXEC) = 5

epoll\_ctl(5, EPOLL\_CTL\_ADD, 4, {0, {u32=3679145984, u64=140736872534016}}) = 0

epoll\_ctl(5, EPOLL\_CTL\_MOD, 4, {EPOLLIN, {u32=3679145984, u64=140736872534016}}) = 0

mmap(NULL, 8392704, PROT\_NONE, MAP\_PRIVATE|MAP\_ANONYMOUS|MAP\_STACK, -1, 0) = 0x7f71e8dd0000

mprotect(0x7f71e8dd1000, 8388608, PROT\_READ|PROT\_WRITE) = 0

clone(child\_stack=0x7f71e95cfb70, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, parent\_tidptr=0x7f71e95d09d0, tls=0x7f71e95d0700, child\_tidptr=0x7f71e95d09d0) = 20243

openat(AT\_FDCWD, "/proc/self/task/20243/comm", O\_RDWR) = 6

write(6, "ZMQbg/0", 7) = 7

close(6) = 0

eventfd2(0, EFD\_CLOEXEC) = 6

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

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

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

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

clock\_gettime(CLOCK\_MONOTONIC, {tv\_sec=10895, tv\_nsec=68580300}) = 0

epoll\_create1(EPOLL\_CLOEXEC) = 7

epoll\_ctl(7, EPOLL\_CTL\_ADD, 6, {0, {u32=3679146240, u64=140736872534272}}) = 0

epoll\_ctl(7, EPOLL\_CTL\_MOD, 6, {EPOLLIN, {u32=3679146240, u64=140736872534272}}) = 0

mmap(NULL, 8392704, PROT\_NONE, MAP\_PRIVATE|MAP\_ANONYMOUS|MAP\_STACK, -1, 0) = 0x7f71e85c0000

mprotect(0x7f71e85c1000, 8388608, PROT\_READ|PROT\_WRITE) = 0

clone(child\_stack=0x7f71e8dbfb70, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, parent\_tidptr=0x7f71e8dc09d0, tls=0x7f71e8dc0700, child\_tidptr=0x7f71e8dc09d0) = 20244

openat(AT\_FDCWD, "/proc/self/task/20244/comm", O\_RDWR) = 8

write(8, "ZMQbg/1", 7) = 7

close(8) = 0

clock\_gettime(CLOCK\_MONOTONIC, {tv\_sec=10895, tv\_nsec=69505700}) = 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=20241, nl\_groups=00000000}, [12]) = 0

time(NULL) = 1609673748 (2021-01-03T14:35:48+0300)

sendto(9, "\24\0\0\0\22\0\1\3\24\254\361\_\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=1609673748, pid=20241}, "\x00\x00\x01\x00\x08\x00\x00\x00\x40\x00\x00\x00\x00\x00\x00\x00\x0a\x00\x01\x00\xd8\xc4\x97\xc2\x6b\x92\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=1609673748, pid=20241}, "\x00\x00\x01\x00\x09\x00\x00\x00\x40\x00\x00\x00\x00\x00\x00\x00\x0a\x00\x01\x00\x1c\x1b\xb5\x88\x10\x48\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=1609673748, pid=20241}, "\x00\x00\x01\x00\x13\x00\x00\x00\x43\x10\x00\x00\x00\x00\x00\x00\x0a\x00\x01\x00\x0a\x00\x27\x00\x00\x13\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=1609673748, pid=20241}, "\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=1609673748, pid=20241}, "\x00\x00\x21\x03\x06\x00\x00\x00\x43\x10\x00\x00\x00\x00\x00\x00\x0a\x00\x01\x00\x1c\x1b\xb5\x88\x10\x44\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=1609673748, pid=20241}, "\x00\x00\x21\x03\x0f\x00\x00\x00\x40\x00\x00\x00\x00\x00\x00\x00\x0a\x00\x01\x00\x1c\x1b\xb5\x88\x10\x45\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=1609673748, pid=20241}, "\x00\x00\x21\x03\x0c\x00\x00\x00\x40\x00\x00\x00\x00\x00\x00\x00\x0a\x00\x01\x00\x1e\x1b\xb5\x88\x10\x44\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=1609673748, pid=20241}, 0}, iov\_len=4096}], msg\_iovlen=1, msg\_controllen=0, msg\_flags=0}, 0) = 20

sendto(9, "\24\0\0\0\26\0\1\3\25\254\361\_\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=1609673749, pid=20241}, "\x02\x10\x00\x00\x08\x00\x00\x00\x08\x00\x01\x00\xa9\xfe\x93\xad\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=1609673749, pid=20241}, "\x0a\x40\x00\xfd\x08\x00\x00\x00\x14\x00\x01\x00\xfe\x80\x00\x00\x00\x00\x00\x00\x3d\x05\xea\xe6\x03\x7d\x93\xad\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=1609673749, pid=20241}, "\x02\x10\x00\x00\x09\x00\x00\x00\x08\x00\x01\x00\xa9\xfe\xd9\x3b\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=1609673749, pid=20241}, "\x0a\x40\x00\xfd\x09\x00\x00\x00\x14\x00\x01\x00\xfe\x80\x00\x00\x00\x00\x00\x00\xe4\x8e\x44\xf8\x07\xe3\xd9\x3b\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=1609673749, pid=20241}, "\x02\x18\x00\x00\x13\x00\x00\x00\x08\x00\x01\x00\xc0\xa8\x38\x01\x08\x00\x04\x00\xc0\xa8\x38\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=1609673749, pid=20241}, "\x0a\x40\x00\xfd\x13\x00\x00\x00\x14\x00\x01\x00\xfe\x80\x00\x00\x00\x00\x00\x00\x54\x5a\x77\xc5\x46\x23\x7e\x74\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=1609673749, pid=20241}, "\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=1609673749, pid=20241}, "\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=1609673749, pid=20241}, "\x02\x18\x00\x00\x06\x00\x00\x00\x08\x00\x01\x00\xc0\xa8\x00\x25\x08\x00\x04\x00\xc0\xa8\x00\xff\x14\x00\x06\x00\x51\x51\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=1609673749, pid=20241}, "\x0a\x40\x00\xfd\x06\x00\x00\x00\x14\x00\x01\x00\xfe\x80\x00\x00\x00\x00\x00\x00\xc5\x1d\xff\x10\x1d\xa5\x8f\x8d\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=1609673749, pid=20241}, "\x02\x10\x00\x00\x0f\x00\x00\x00\x08\x00\x01\x00\xa9\xfe\xfe\x9f\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=1609673749, pid=20241}, "\x0a\x40\x00\xfd\x0f\x00\x00\x00\x14\x00\x01\x00\xfe\x80\x00\x00\x00\x00\x00\x00\x91\xc4\x97\xe8\x18\x2d\xfe\x9f\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=1609673749, pid=20241}, "\x02\x10\x00\x00\x0c\x00\x00\x00\x08\x00\x01\x00\xa9\xfe\x13\x01\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=1609673749, pid=20241}, "\x0a\x40\x00\xfd\x0c\x00\x00\x00\x14\x00\x01\x00\xfe\x80\x00\x00\x00\x00\x00\x00\xb1\x56\x78\x65\x03\x84\x13\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=20, type=NLMSG\_DONE, flags=NLM\_F\_MULTI, seq=1609673749, pid=20241}, 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(4040), sin\_addr=inet\_addr("127.0.0.1")}, 16) = 0

listen(9, 100) = 0

getsockname(9, {sa\_family=AF\_INET, sin\_port=htons(4040), 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=10895, tv\_nsec=74918500}) = 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)

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

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

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

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

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

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

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

fstat(1, {st\_mode=S\_IFCHR|0660, st\_rdev=makedev(4, 1), ...}) = 0

ioctl(1, TCGETS, {B38400 opost isig icanon echo ...}) = 0

write(1, "OK: 20242\n", 10) = 10

read(0, "exit\n", 4096) = 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=20242, 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, да и вообще для любых мультипроцессорных программ.